DMBVP for Tension Splines

Boris I. Kvasov
Institute of Computational Technologies, Russian Academy of Sciences
Lavrentyev Avenue 6, 630090, NOVOSIBIRSK, RUSSIA,
kvasov@lchd.ict.nsc.ru

Abstract. This paper addresses a new approach in solving the problem of shape preserving spline interpolation. Based on the formulation of the latter problem as a differential multipoint boundary value problem for hyperbolic and biharmonic tension splines we consider its finite-difference approximation. The resulting system of linear equations can be efficiently solved either by direct (Gaussian elimination) and iterative methods (successive over-relaxation (SOR) method and finite-difference schemes in fractional steps). We consider the basic computational aspects and illustrate the main advantages of this original approach.

Keywords: Hyperbolic and biharmonic tension splines, differential multipoint boundary value problem, successive over-relaxation method, finite-difference schemes in fractional steps, 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., [19]) and the variational one (where splines are obtained via minimization of quadratic functionals with equality and/or inequality constraints, see, e.g., [13]). Although less common, a third approach [8], where splines are defined as the solutions of differential multipoint boundary value problems (DMBVP for short), has been considered in [3,11,12] and closely relates to the idea of polysplines [10]. Even though some of the important classes of splines can be obtained from all three schemes, specific features sometimes make the last one an important tool in practical settings. We want to illustrate this fact by the examples of interpolating hyperbolic and biharmonic tension splines. Introduced by Schweikert in 1966 [20] hyperbolic tension splines are still very popular [9,15,16,17,18]. Earlier biharmonic (thin plate) tension splines were considered in [2,4,5,7,12], etc.

Applied Mathematics and Scientific Computing: Brijuni 2003
XXX (eds.), pp. 1–26.

ISBN XXX.

For the numerical treatment of a DMBVP we replace the differential operator by its finite-difference approximation. This gives us a linear system of difference equations with a matrix of special structure. The latter system can be efficiently treated by the Gaussian elimination or by iterative methods such as SOR iterative method or finite-difference schemes in fractional steps [21]. We present numerical examples illustrating the main features of this approach.

The content of this paper is as follows. In Section 2 we formulate the 1-D problem. In Section 3 we prove the existence of a mesh solution by constructing its extension as a discrete hyperbolic tension spline. Section 4, with its subsections, is devoted to the discussion of practical aspects and computational advantages of our discrete spline. In Sections 5 and 6 we formulate the 2-D problem and give its finite-difference approximation. The algorithm for the numerical solution of 2-D problem is described in section 7. Section 8 gives the SOR iterative method. In section 9 we consider a finite-difference scheme in fractional steps and treat its approximation and stability properties. Finally, Section 10 provides some graphical examples to illustrate the main properties of discrete hyperbolic and biharmonic tension splines.

§2. 1-D DMBVP. Finite Difference Approximation

Let the data

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

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

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

Definition 1. *An interpolating hyperbolic spline* S *with a set of tension parameters* $\{p_i \geq 0 \mid i = 0, \ldots, N\}$ *is a solution of the DMBVP*

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

$$\mathrm{S} \in C^2[a, b], \tag{3}$$

with the interpolation conditions

$$\mathrm{S}(x_i) = f_i, \quad i = 0, \ldots, N+1, \tag{4}$$

and the end conditions

$$\mathrm{S}''(a) = f_0'' \quad \text{and} \quad \mathrm{S}''(b) = f_{N+1}''. \tag{5}$$

The classical end constraints (5) we consider only for the sake of simplicity. One can also use other types of the end conditions [11].

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:

$$\left[\Lambda_i^2 - \left(\frac{p_i}{h_i}\right)^2 \Lambda_i\right] u_{ij} = 0, \qquad j = 1, \dots, n_i - 1 \ , \quad i = 0, \dots, N, \tag{6}$$

where

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

The smoothness condition (3) is changed into

$$\begin{aligned} u_{i-1,n_{i-1}} &= u_{i0} \ , \\ \frac{u_{i-1,n_{i-1}+1} - u_{i-1,n_{i-1}-1}}{2\tau_{i-1}} &= \frac{u_{i,1} - u_{i,-1}}{2\tau_i} \ , \quad i = 1, \dots, N, \\ \Lambda_{i-1} u_{i-1,n_{i-1}} &= \Lambda_i u_{i,0} \end{aligned} \tag{7}$$

while conditions (4)–(5) take the form

$$\begin{aligned} & u_{i,0} = f_i \ , \ i = 0, \dots, N, \quad u_{N,n_N} = f_{N+1} \ , \\ & \Lambda_0 u_{0,0} = f_0'', \qquad \Lambda_N u_{N,n_N} = f_{N+1}'' \ . \end{aligned} \tag{8}$$

Our discrete *mesh solution* will be then defined as

$$\{u_{ij}, \quad j = 0, \dots, n_i, \quad i = 0, 1, \dots, N\}. \tag{9}$$

In the next section we prove the existence of the solution of the previous linear system while we postpone to Section 4 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 (6)–(8) we introduce the notation

$$m_{ij} = \Lambda_i u_{ij}, \quad j = 0, \dots, n_i, \quad i = 0, \dots, N. \tag{10}$$

Then, on the interval $[x_i, x_{i+1}]$, (6) takes the form

$$\begin{aligned} & m_{i0} = m_i, \\ & \frac{m_{i,j-1} - 2m_{ij} + m_{i,j+1}}{\tau_i^2} - \left(\frac{p_i}{h_i}\right)^2 m_{ij} = 0, \ j = 1, \dots, n_i - 1, \\ & m_{i,n_i} = m_{i+1}, \end{aligned} \tag{11}$$

where m_i and m_{i+1} are prescribed numbers. The system (11) has a unique solution, which can be represented as follows

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

with

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

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 (10) and from the interpolation conditions (8) we have

$$\begin{aligned} u_{i0} &= f_i, \\ \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\tau_i^2} &= m_{ij}, \ j = 0, \dots, n_i, \\ u_{i,n_i} &= f_{i+1}. \end{aligned} \tag{12}$$

For each sequence m_{ij}, $j = 0, \dots, n_i$, system (12) has a unique solution which can be represented as follows

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

where

$$\mathrm{U}_i(x) = f_i(1-t) + f_{i+1}t + \varphi_i(1-t)h_i^2 m_i + \varphi_i(t)h_i^2 m_{i+1}, \tag{13}$$

with

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

In order to solve system (6)–(8), we only need to determine the values m_i, $i = 0, \dots, N+1$, so that the smoothness conditions (7) and the end conditions in (8) are verified. From (12)–(13), conditions (7) can be rewritten as

$$\begin{gathered} \mathrm{U}_{i-1}(x_i) = \mathrm{U}_i(x_i), \\ \frac{\mathrm{U}_{i-1}(x_i + \tau_{i-1}) - \mathrm{U}_{i-1}(x_i - \tau_{i-1})}{2\tau_{i-1}} = \frac{\mathrm{U}_i(x_i + \tau_i) - \mathrm{U}_i(x_i - \tau_i)}{2\tau_i}, \\ \Lambda_{i-1}\mathrm{U}_{i-1}(x_i) = \Lambda_i \mathrm{U}_i(x_i), \end{gathered} \tag{14}$$

where

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

Then, from (10)–(11) and (12), the first and the third equalities in (14) are immediately satisfied, while, using (13) and the end conditions in (8), the second equality provides the following linear system with a 3-diagonal matrix for the unknown values m_i:

$$\begin{aligned} m_0 &= f_0'', \\ \alpha_{i-1}h_{i-1}m_{i-1} + (\beta_{i-1}h_{i-1} + \beta_i h_i)m_i + \alpha_i h_i m_{i+1} &= d_i, \ \ i = 1, \dots, N, \\ m_{N+1} &= f_{N+1}'', \end{aligned} \tag{15}$$

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

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

Therefore, the system (15) is diagonally dominant and has a unique solution. We can now conclude that system (6)–(8) has a unique solution which can be represented as $\mathrm{U}_i(x_{ij})$, $j = -1, \dots, n_i + 1$, $i = 0, \dots, N$, whenever the constants m_i are solution of (15).

Let us put

$$\mathrm{U}(x) := \mathrm{U}_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, \dots, N. \tag{16}$$

Due to the previous construction we will refer to U as *discrete hyperbolic tension spline* interpolating the data (1). We observe that we recover the result of [14] for discrete cubics since

$$\lim_{p_i \to 0} \alpha_i = \frac{1}{6}\Big(1 - \frac{1}{n_i^2}\Big), \quad \lim_{p_i \to 0} \beta_i = \frac{1}{6}\Big(2 + \frac{1}{n_i^2}\Big), \quad \lim_{p_i \to 0} \varphi_i(t) = \frac{t(t^2-1)}{6}. \tag{17}$$

§4. Computational Aspects

The aim of this section is to investigate the practical aspects related to the numerical evaluation of the mesh solution defined in (9).

A standard approach, [17], consists of solving the tridiagonal system (15) and then evaluating (13) 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 (15). 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 (15) and to tabulate functions (13). This is a very difficult task, both for cancellation errors (when $k_i \to 0$) and for overflow problems (when $k_i \to \infty$). A stable computation of the hyperbolic functions was proposed in [17], 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 (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 (6)–(8).

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

Step 1. S*olve the* 3*-diagonal system* (15) *for* m_i, $i = 1, \ldots, N$.

Step 2. S*olve* $N + 1$ 3*-diagonal systems* (11) *for* m_{ij}, $j = 1, \ldots, n_i - 1$, $i = 0, \ldots, N$,

Step 3. S*olve* $N + 1$ 3*-diagonal systems* (12) *for* u_{ij}, $j = 1, \ldots, n_i - 1$, $i = 0, \ldots, N$.

In this algorithm, hyperbolic functions need only be computed in step 1. Furthermore, the solution of any system (11) or (12) requires $8q$ arithmetic operations, namely, $3q$ additions, $3q$ multiplications, and $2q$ divisions [22], where q is the number of unknowns, and is thus substantially cheaper than direct computation by formula (13).

Steps 2 and 3 can be replaced by a direct splitting of the system

(6)–(8) into $N+1$ systems with 5-diagonal matrices

$$\begin{aligned} u_{i,0} &= f_i, \quad \Lambda_i u_{i,0} = M_i, \\ \Lambda_i^2 u_{i,j} - \left(\frac{p_i}{h_i}\right)^2 \Lambda_i u_{i,j} &= 0, \quad j = 1, \ldots, n_i - 1, \quad i = 0, \ldots, N. \\ u_{i,n_i} &= f_{i+1}, \quad \Lambda_i u_{i,n_i} = M_{i+1}, \end{aligned} \tag{18}$$

Also, in this case the calculations for steps 2 and 3 or for system (18) can be tailored for a multiprocessor computer system.

Let us discuss now the direct solution of system (6)–(8) which, of course, only involves rational computations on the given data. In order to do this in the next subsections we investigate in some details the structure of the mentioned system.

4.1 The Pentadiagonal System

Eliminating the unknowns $\{u_{i,-1},\ i = 1, \ldots, N,\}$ and $\{u_{i,n_i+1},\ i = 0, \ldots, N-1\}$, from (7) determining the values of the mesh solution at the data sites x_i by the interpolation conditions and eliminating $u_{0,-1},\ u_{N,n_N+1}$ from the end conditions (8) we can collect (6)–(8) into the system

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{19}$$

where

$$\mathbf{u} = (u_{01}, \ldots, u_{0,n_0-1}, u_{11}, \ldots, u_{21}, \ldots, u_{N1}, \ldots, u_{N,n_N-1})^T,$$

$\mathbf{A}$ is the following pentadiagonal matrix:

$$\left[\begin{array}{cccccccccccc}
b_0-1 & a_0 & 1 & & & & & & & & & \\
a_0 & b_0 & a_0 & 1 & & & & & & & & \\
1 & a_0 & b_0 & a_0 & 1 & & & & & & & \\
 & & \ldots & & & & & & & & & \\
 & & 1 & a_0 & b_0 & a_0 & & & & & & \\
 & & & 1 & a_0 & \eta_{0,n_0-1} & \delta_{0,n_0-1} & & & & & \\
 & & & & & \delta_{1,1} & \eta_{1,1} & a_1 & 1 & & & \\
 & & & & & & a_1 & b_1 & a_1 & 1 & & \\
 & & & & & & & & \ldots & & & \\
 & & & & & & & 1 & a_N & b_N & a_N & 1 \\
 & & & & & & & & 1 & a_N & b_N & a_N \\
 & & & & & & & & & 1 & a_N & b_N-1
\end{array}\right]$$

with

$$
\begin{aligned}
&a_i = -(4+\omega_i)\ ,\ b_i = 6+2\omega_i\ ,\ \omega_i = \left(\frac{p_i}{n_i}\right)^2;\quad i=0,1,\ldots,N,\\
&\eta_{i-1,n_{i-1}-1} = 6+2\omega_{i-1}+\frac{1-\rho_i}{1+\rho_i}\ ,\ \eta_{i,1} = 6+2\omega_i+\frac{\rho_i-1}{\rho_i+1}\\
&\delta_{i-1,n_{i-1}-1} = \frac{2}{\rho_i(\rho_i+1)}\ ,\ \delta_{i,1} = 2\frac{\rho_i^2}{\rho_i+1}\ ,\\
&\rho_i = \frac{\tau_i}{\tau_{i-1}},\qquad i=1,2,\ldots,N;
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbf{b} = (&-(a_0+2)f_0-\tau_0^2 f_0'', -f_0, 0,\ldots,0,-f_1,-\gamma_{0,n_0-1}f_1,-\gamma_{1,1}f_1,-f_1,0,\\
&\ldots,0,-f_{N+1},-(a_N+2)f_{N+1}-\tau_N^2 f_{N+1}'')^T,
\end{aligned}
$$

with

$$
\left.\begin{aligned}
\gamma_{i-1,n_{i-1}-1} &= -(4+\omega_{i-1}+2\frac{1-\rho_i}{\rho_i}),\\
\gamma_{i,1} &= -(4+\omega_i+2(\rho_i-1)),
\end{aligned}\right.\quad i=1,2,\ldots,N.
$$

4.2 The Uniform Case

From the practical point of view it is interesting to examine the structure of $\mathbf{A}$ when we are dealing with a uniform mesh, that is $\tau_i = \tau$. In such a case it is immediately seen that $\mathbf{A}$ is symmetric. In addition, following [14] we observe that $\mathbf{A} = \mathbf{C} + \mathbf{D}$, where both $\mathbf{C}$ and $\mathbf{D}$ are symmetric block diagonal matrices. To be more specific,

$$
\mathbf{C} = \begin{bmatrix} \mathbf{C}_0 & & & \\ & \mathbf{C}_1 & & \\ & & \ddots & \\ & & & \mathbf{C}_N \end{bmatrix},\quad \mathbf{C}_i = \mathbf{B}_i^2 - \omega_i \mathbf{B}_i,
$$

where $\mathbf{B}_i$ is the $(n_i-1)\times(n_i-1)$ tridiagonal matrix

$$
\mathbf{B}_i = \begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & & \cdots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix};
$$

and

$$
\mathbf{D} = \begin{bmatrix}
0 & 0 & & & & & & & & & & & & & \\
0 & 0 & & & & & & & & & & & & & \\
 & & \ddots & & & & & & & & & & & & \\
 & & & 0 & & & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & \\
 & & & & & & 0 & & & & & & & & \\
 & & & & & & & \ddots & & & & & & & \\
 & & & & & & & & 0 & & & & & & \\
 & & & & & & & & & 1 & 1 & & & & \\
 & & & & & & & & & 1 & 1 & & & & \\
 & & & & & & & & & & & 0 & & & \\
 & & & & & & & & & & & & \ddots & & \\
 & & & & & & & & & & & & & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0
\end{bmatrix}.
$$

The eigenvalues of $\mathbf{C}$, $\lambda_k(\mathbf{C})$, are the collection of the eigenvalues of $\mathbf{C}_i$. Since, (see [14]),

$$
\lambda_j(\mathbf{B}_i) = -2\Big(1 - \cos\frac{j\pi}{n_i}\Big), \ \ j = 1, \ldots, n_i - 1,
$$

we have

$$
\lambda_j(\mathbf{C}_i) = 4\Big(1 - \cos\frac{j\pi}{n_i}\Big)^2 + 2\omega_i\Big(1 - \cos\frac{j\pi}{n_i}\Big) \quad j = 1, \ldots, n_i - 1.
$$

In addition, the eigenvalues of $\mathbf{D}$ are 0 and 2, thus we deduce from a corollary of the Courant-Fisher theorem [6] that the eigenvalues of $\mathbf{A}$ satisfy the following inequalities

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

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 [14] for the limit case $p_i = 0, \ \ i = 0, \ldots, N$,

$$
\begin{aligned}
\|\mathbf{A}\|_\infty \|\mathbf{A}^{-1}\|_\infty &\leq \frac{\max_i \big[16 + 4(\frac{p_i}{n_i})^2\big]}{\min_i \big[4(1 - \cos\frac{\pi}{n_i})^2 + 2(\frac{p_i}{n_i})^2(1 - \cos\frac{\pi}{n_i})\big]} \\
&\simeq \frac{\max_i \big[16 + 4(\frac{p_i}{n_i})^2\big]}{\min_i (\frac{1}{n_i})^4 [\pi^4 + (\pi p_i)^2]}.
\end{aligned} \tag{20}
$$

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 [22], where q is the number of unknowns.

Moreover, since the upper bound (20) 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 section 10.

§5. 2-D DMBVP. Problem Formulation

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

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

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

$$\begin{aligned} \Delta_x &: a = x_0 < x_1 < \cdots < x_{N+1} = b, \\ \Delta_y &: c = y_0 < y_1 < \cdots < y_{M+1} = d, \end{aligned}$$

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

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

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

Let us associate to the mesh Δ the data

$$\begin{array}{lll} (x_i, y_j, f_{ij}), & i = 0, \ldots, N+1, & j = 0, \ldots, M+1, \\ f_{ij}^{(2,0)}, & i = 0, N+1, & j = 0, \ldots, M+1, \\ f_{ij}^{(0,2)}, & i = 0, \ldots, N+1, & j = 0, M+1, \\ f_{ij}^{(2,2)}, & i = 0, N+1, & j = 0, M+1, \end{array}$$

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 $S \in C^{2,2}[\overline{\Omega}]$ such that $S(x_i, y_j) = f_{ij}$, $i = 0, \ldots, N+1$, $j = 0, \ldots, M+1$, and S preserves the shape of the initial data the shape preserving interpolation problem. This means that wherever the data increases (decreases) monotonically, S has the same behaviour, and S is convex (concave) over intervals where the data is 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 biharmonic tension spline.

Definition 2. *An interpolating biharmonic spline S with two sets of tension parameters $\{\, 0 \le p_{ij} < \infty \mid i = 0, \ldots, N,\ j = 0, \ldots, M+1 \,\}$ and $\{\, 0 \le q_{ij} < \infty \mid i = 0, \ldots, N+1,\ j = 0, \ldots, M \,\}$ is a solution of the DMBVP*

$$\frac{\partial^4 S}{\partial x^4} + 2\frac{\partial^4 S}{\partial x^2 \partial y^2} + \frac{\partial^4 S}{\partial y^4} - \left(\frac{\overline{p}_{ij}}{h_i}\right)^2 \frac{\partial^2 S}{\partial x^2} - \left(\frac{\overline{q}_{ij}}{l_j}\right)^2 \frac{\partial^2 S}{\partial y^2} = 0 \tag{21}$$

$$\begin{aligned} \text{in each } \ \Omega_{ij},\ \ & h_i = x_{i+1} - x_i,\ l_j = y_{j+1} - y_j, \\ & \overline{p}_{ij} = \max(p_{ij}, p_{i,j+1}), \quad \overline{q}_{ij} = \max(q_{ij}, q_{i+1,j}), \\ & i = 0, \ldots, N,\ j = 0, \ldots, M, \end{aligned}$$

$$\frac{\partial^4 S}{\partial x^4} - \left(\frac{p_{ij}}{h_i}\right)^2 \frac{\partial^2 S}{\partial x^2} = 0, \quad \begin{aligned} & x \in (x_i, x_{i+1}), \quad i = 0, \ldots, N, \\ & y = y_j, \quad j = 0, \ldots, M+1, \end{aligned} \tag{22}$$

$$\frac{\partial^4 S}{\partial y^4} - \left(\frac{q_{ij}}{l_j}\right)^2 \frac{\partial^2 S}{\partial y^2} = 0, \quad \begin{aligned} & y \in (y_j, y_{j+1}), \quad j = 0, \ldots, M, \\ & x = x_i, \quad i = 0, \ldots, N+1, \end{aligned} \tag{23}$$

$$S \in C^{2,2}[\overline{\Omega}], \tag{24}$$

with the interpolation conditions

$$S(x_i, y_j) = f_{ij}, \quad i = 0, \ldots, N+1, \quad j = 0, \ldots, M+1, \tag{25}$$

and the boundary conditions

$$\begin{aligned} S^{(2,0)}(x_i, y_j) &= f_{ij}^{(2,0)}, & i &= 0, N+1, & j &= 0, \ldots, M+1, \\ S^{(0,2)}(x_i, y_j) &= f_{ij}^{(0,2)}, & i &= 0, \ldots, N+1, & j &= 0, M+1, \\ S^{(2,2)}(x_i, y_j) &= f_{ij}^{(2,2)}, & i &= 0, N+1, & j &= 0, M+1. \end{aligned} \tag{26}$$

By this definition an interpolating biharmonic tension spline S is a set of the interpolating biharmonic tension functions which satisfy (21), match up smoothly and form a twice continuously differentiable function both in x and y variables

$$\begin{aligned} S^{(r,0)}(x_i-0,y) =& S^{(r,0)}(x_i+0,y), \quad r=0,1,2, \quad i=1,\dots,N, \\ S^{(0,s)}(x,y_j-0) =& S^{(0,s)}(x,y_j+0), \quad s=0,1,2, \quad j=1,\dots,M. \end{aligned} \tag{27}$$

C^2 smoothness of the interpolating hyperbolic tension splines in (22) and (23) was proven in [3,11]. The computation of the interpolating biharmonic tension spline reduces to a computation of infinitely many proper one-dimensional hyperbolic tension splines.

For all $p_{ij}, q_{ij} \to 0$ the solution of (21)–(26) becomes a biharmonic spline [4] while in the limiting case as $p_{ij}, q_{ij} \to \infty$ 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 towards an inherent shape while at the same time keeping its smoothness. Thus, the DMBVP gives an approach to the solution of the shape preserving interpolation problem.

§6. 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 $h_i/n_i = l_j/m_j = h$. 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

$$\left[\Lambda_1^2 + 2\Lambda_1\Lambda_2 + \Lambda_2^2 - \left(\frac{\overline{p}_{ij}}{h_i}\right)^2 \Lambda_1 - \left(\frac{\overline{q}_{ij}}{l_j}\right)^2 \Lambda_2\right] u_{ik;jl} = 0, \tag{28}$$

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

$$\left[\Lambda_1^2 - \left(\frac{p_{ij}}{h_i}\right)^2 \Lambda_1\right] u_{ik;jl} = 0, \tag{29}$$

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

$$\left[\Lambda_2^2 - \left(\frac{q_{ij}}{l_j}\right)^2 \Lambda_2\right] u_{ik;jl} = 0, \tag{30}$$

$$k = \begin{cases} 0, & \text{if } i = 0, \dots, N-1, \\ 0, n_N & \text{if } i = N, \end{cases}; \quad l = 1, \dots, m_j - 1, \; j = 0, \dots, M,$$

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 (27) are changed to

$$\begin{aligned} u_{i-1,n_{i-1};jl} =& u_{i0;jl}, \\ \frac{u_{i-1,n_{i-1}+1;jl} - u_{i-1,n_{i-1}-1;jl}}{2h} =& \frac{u_{i1;jl} - u_{i,-1;jl}}{2h}, \\ \Lambda_1 u_{i-1,n_{i-1};jl} =& \Lambda_1 u_{i0;jl}, \end{aligned} \tag{31}$$

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

$$\begin{aligned} u_{ik;j-1,m_{j-1}} =& u_{ik;j0}, \\ \frac{u_{ik;j-1,m_{j-1}+1} - u_{ik;j-1,m_{j-1}-1}}{2h} =& \frac{u_{ik;j1} - u_{ik;j,-1}}{2h}, \\ \Lambda_2 u_{ik;j-1,m_{j-1}} =& \Lambda_2 u_{ik;j0}, \end{aligned} \tag{32}$$

$$k = 0, \dots, n_i, \; i = 0, \dots, N, \; j = 1, \dots, M.$$

Conditions (25) and (26) take the form

$$\begin{aligned} u_{i0;j0} =& f_{ij}, & u_{N,n_N;j0} =& f_{N+1,j}, \\ u_{i0;M,m_M} =& f_{i,M+1}, & u_{N,n_N;M,m_M} =& f_{N+1,M+1}, \end{aligned} \tag{33}$$

$$i = 0, \dots, N, \; j = 0, \dots, M,$$

and

$$\begin{aligned} \Lambda_1 u_{00;j0} =& f_{0j}^{(2,0)}, \quad j = 0, \dots, M; & \Lambda_1 u_{00;M,m_M} =& f_{0,M+1}^{(2,0)}, \\ \Lambda_1 u_{N,n_N;j0} =& f_{N+1,j}^{(2,0)}, \quad j = 0, \dots, M; & \Lambda_1 u_{N,n_N;M,m_M} =& f_{N+1,M+1}^{(2,0)}, \\ \Lambda_2 u_{i0;00} =& f_{i0}^{(0,2)}, \quad i = 0, \dots, N; & \Lambda_2 u_{N,n_N;00} =& f_{N+1,0}^{(0,2)}, \\ \Lambda_2 u_{i0;M,m_M} =& f_{i,M+1}^{(0,2)}, \quad i = 0, \dots, N; & \Lambda_2 u_{N,n_N;M,m_M} =& f_{N+1,M+1}^{(0,2)}, \\ \Lambda_1 \Lambda_2 u_{00;00} =& f_{00}^{(2,2)}, & \Lambda_1 \Lambda_2 u_{N,n_N;00} =& f_{N+1,0}^{(2,2)}, \\ \Lambda_1 \Lambda_2 u_{00;M,m_M} =& f_{0,M+1}^{(2,2)}, & \Lambda_1 \Lambda_2 u_{N,n_N;M,m_M} =& f_{N+1,M+1}^{(2,2)}. \end{aligned} \tag{34}$$

§7. Algorithm

To solve finite-difference system (28)–(34) we propose first to find its solution on the refinement of the main mesh Δ. The latter can be achieved in the four steps.

First step. Evaluate all tension parameters p_{ij} on the lines $y = y_j$, $j = 0, \ldots, M+1$ and q_{ij} on the lines $x = x_i$, $i = 0, \ldots, N+1$ by one of 1-D algorithms for automatic selection of shape control parameters, see, e.g., [11,16,17], etc.

Second step. Construct discrete hyperbolic tension splines [3] in the x direction by solving the $M+2$ linear systems (29). As a result, one finds the values of the mesh solution on the lines $y = y_j$, $j = 0, \ldots, M+1$ of the mesh Δ in x direction.

Third step. Construct discrete hyperbolic tension splines in the y direction by solving the $N+2$ linear systems (30). This gives us the values of the mesh solution on the lines $x = x_i$, $i = 0, \ldots, N+1$ of the mesh Δ in y direction.

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

$$\begin{gathered} \Lambda_1 u_{00;jl}, \quad \Lambda_1 u_{N,n_N;jl}, \quad l = 0, \ldots, m_j, \quad j = 0, \ldots, M, \\ \Lambda_2 u_{ik;00}, \quad \Lambda_2 u_{ik;M,m_M}, \quad k = 0, \ldots, n_i, \quad i = 0, \ldots, N. \end{gathered} \tag{35}$$

Now the system of difference equations (28)–(34) can be substantially simplified by eliminating the unknowns

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

using relations (31), (32), and the boundary values (35).

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 Ω_{ij}, $i = 0, \ldots, N$, $j = 0, \ldots, M$. This linear system can be efficiently solved by the SOR algorithm or applying finite-difference schemes in fractional steps on single- or multi-processor computers.

§8. 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

$$\{u_{ik;jl}^{(0)} \,|\, k = 0, \ldots, n_i, \ i = 0, \ldots, N, \ l = 0, \ldots, m_j, \ j = 0, \ldots, M\}. \tag{36}$$

In each rectangle Ω_{ij}, $i = 0,\dots,N$, $j = 0,\dots,M$, the difference equation (28) can be rewritten in componentwise form

$$\begin{aligned} u_{ik;jl} =& \frac{1}{\alpha_{ij}}\Big\{\beta_{ij}\left[u_{i,k-1;jl} + u_{i,k+1;jl}\right] + \gamma_{ij}\left[u_{ik;j,l-1} + u_{ik;j,l+1}\right] \\ & - 2\big[u_{i,k-1;j,l-1} + u_{i,k-1;j,l+1} + u_{i,k+1;j,l-1} + u_{i,k+1;j,l+1}\big] \\ & - u_{ik;j,l-2} - u_{ik;j,l+2} - u_{i,k-2;jl} - u_{i,k+2;jl}\Big\}, \end{aligned} \tag{37}$$

where

$$\alpha_{ij} = 20 + 2\left(\frac{\overline{p}_{ij}}{n_i}\right)^2 + 2\left(\frac{\overline{q}_{ij}}{m_j}\right)^2, \quad \beta_{ij} = 8 + \left(\frac{\overline{p}_{ij}}{n_i}\right)^2, \quad \gamma_{ij} = 8 + \left(\frac{\overline{q}_{ij}}{m_j}\right)^2.$$

Now using (37) we can write down SOR iterations to obtain a numerical solution on the refinement

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

Note that outside the domain $\overline{\Omega}$ 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 (35) and are not part of the iterations.

§9. Method of Fractional Steps

The system of difference equations obtained in section 4 can be efficiently solved by the method of fractional steps [21]. Using the initial approximation (36) let us consider in each rectangle Ω_{ij}, $i = 0,\dots,N$, $j = 0,\dots,M$, the following splitting scheme

$$\begin{aligned} \frac{u^{n+1/2} - u^n}{\tau} + \Lambda_{11}u^{n+1/2} + \Lambda_{12}u^n = 0, \\ \frac{u^{n+1} - u^{n+1/2}}{\tau} + \Lambda_{22}u^{n+1} + \Lambda_{12}u^{n+1/2} = 0, \end{aligned} \tag{38}$$

where

$$\Lambda_{11} = \Lambda_1^2 - p\Lambda_1,\ \Lambda_{22} = \Lambda_2^2 - q\Lambda_2,\ \Lambda_{12} = \Lambda_1\Lambda_2,\ p = \left(\frac{\overline{p}_{ij}}{h_i}\right)^2,\ q = \left(\frac{\overline{q}_{ij}}{l_j}\right)^2,$$

$$u = \{u_{ik;jl} \mid k = 1,\dots,n_i-1,\ i = 0,\dots,N;\quad l = 1,\dots,m_j-1,\ j = 0,\dots,M\}.$$

Eliminating from here the fractional step $u^{n+1/2}$ yields the following scheme in whole steps, equivalent to the scheme (38),

$$\frac{u^{n+1}-u^n}{\tau} + (\Lambda_{11}+\Lambda_{22})u^{n+1} + 2\Lambda_{12}u^n + \tau(\Lambda_{11}\Lambda_{22}u^{n+1} - \Lambda_{12}^2u^n) = 0. \quad (39)$$

It follows from here that the scheme (39) and the equivalent scheme (38) possess the property of complete approximation [21] only in the case if

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

Let us prove the unconditional stability of the scheme (38) or, which is equivalent, the scheme (39). Using usual harmonic analysis [21] assume that

$$u^n = \eta_n e^{i\pi z}, \quad u^{n+1/2} = \eta_{n+1/2} e^{i\pi z}, \quad z = k_1\frac{x-x_i}{h_i} + k_2\frac{y-y_j}{l_j}. \quad (40)$$

Substituting equations (40) into equations (38) we obtain the amplification factors

$$\rho_1 = \frac{\eta_{n+1/2}}{\eta_n} = \frac{1-a_1a_2}{1-p\sqrt{\tau}a_1+a_1^2}, \quad \rho_2 = \frac{\eta_{n+1}}{\eta_{n+1/2}} = \frac{1-a_1a_2}{1-q\sqrt{\tau}a_2+a_2^2},$$

$$\rho = \rho_1\rho_2 = \frac{(1-a_1a_2)^2}{(1-p\sqrt{\tau}a_1+a_1^2)(1-q\sqrt{\tau}a_2+a_2^2)},$$

where

$$a_1 = -\frac{4\sqrt{\tau}}{h^2}\sin^2\left(\frac{k_1h}{2}\frac{\pi}{h_i}\right), \quad k_1 = 1,\dots,n_i-1, \quad n_ih = h_i,$$

$$a_2 = -\frac{4\sqrt{\tau}}{h^2}\sin^2\left(\frac{k_2h}{2}\frac{\pi}{l_j}\right), \quad k_2 = 1,\dots,m_j-1, \quad m_jh = l_j.$$

It follows from here that

$$0 \le \rho \le \frac{(1-a_1a_2)^2}{(1+a_1^2)(1+a_2^2)} \le \left(\frac{1-a_1a_2}{1+a_1a_2}\right)^2 < 1$$

for any τ. This proves the unconditional stability of the scheme (38).

At each fractional step in (38) one has to solve a linear system with a symmetric positive definite pentadiagonal matrix. This is much cheaper than directly solving the linear system (28). However, in general the scheme (38) has the property of incomplete approximation [21]. For this reason, in iterations we have to use small values of the iteration parameter τ, e.g., $\sqrt{\tau}/h^2 = const$.

§10. Graphical Examples

The aim of this final section is to illustrate the tension features of discrete hyperbolic and biharmonic tension splines with some (famous) examples. Before, we want to notice that the continuous form U_i of our solution given in (13) has the good shape-preserving properties of cubics (see e.g. [17]) 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. [16, 17].

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. 1–2. 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 (15), $m_0 = m_{N+1} = 0$.

Table 1. *Radio chemical* data:

x_i	7.99	8.09	8.19	8.7	9.2	10	12	15	20
f_i	0	2.76429E-5	4.37498E-2	0.169183	0.469428	0.943740	0.998636	0.999916	0.999994

Fig. 1 is obtained setting $p_i = 0$, that is considering the discrete cubic spline interpolating the data. In Fig 2 a new discrete interpolant with $p_0 = p_1 = 300$, $p_i = 15$, $i = 2, \ldots, 7$, 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 end conditions $m_0 = m_{N+1} = 0$. Fig. 3 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

In 2-D case the approach developed in this paper was tested on a number of numerical examples. Because of space limitations we consider here only some of them. The initial data $(x_i, y_j, \tilde{f}_{ij})$ in Fig. 4 was obtained

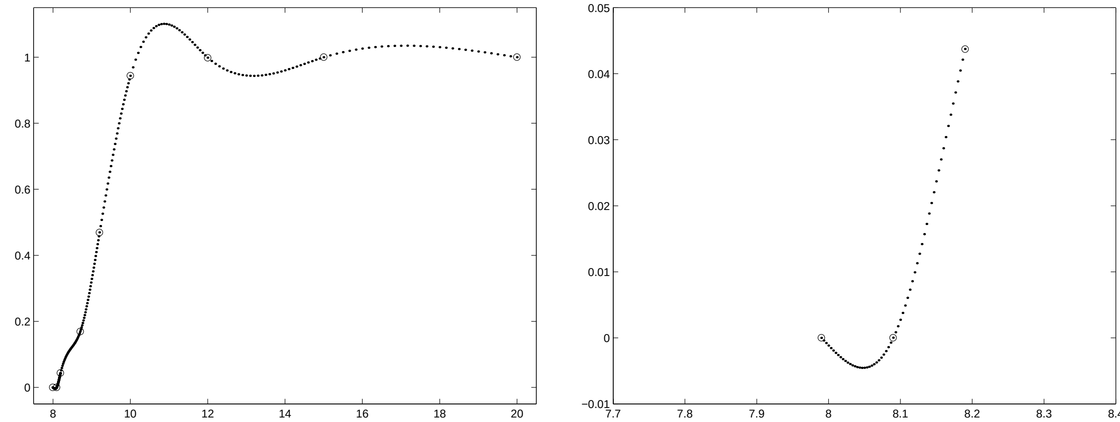

Fig. 1. The radio chemical data with natural end conditions $m_0 = m_{N+1} = 0$.
Interpolation by discrete cubic spline ($p_i = 0$).
Right: a magnification of the lower left corner.

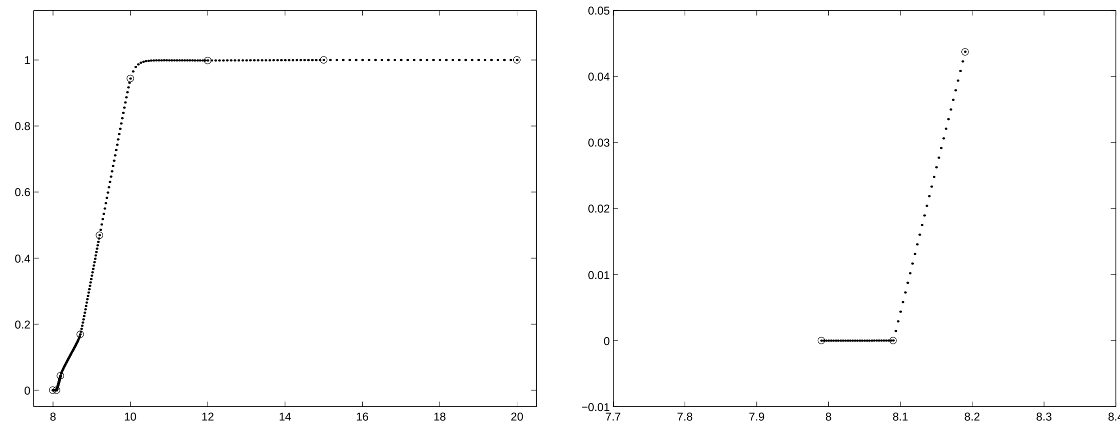

Fig. 2. The same as Fig. 1 with $p_0 = p_1 = 300,\ p_i = 15, i = 2, \ldots, 7$.

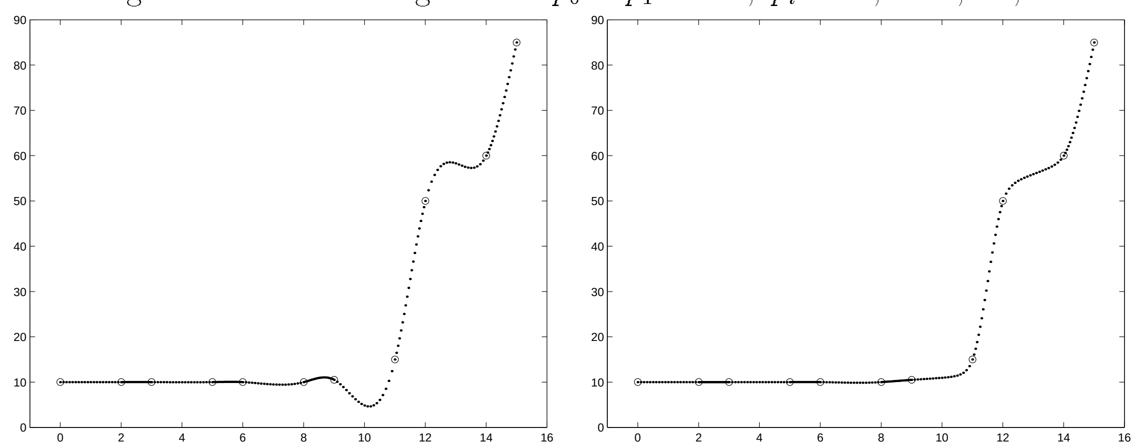

Fig. 3. Akima's data with natural end conditions.
Left: Discrete interpolating cubic spline ($p_i = 0$).
Right: discrete hyperbolic spline with $p_5 = p_6 = p_8 = 10$.

by taking Akima's data in Table 2 both in x and y directions and using the formula $\tilde{f}_{ij} = f_i + f_j$. As shown in Fig. 5 the usual discrete biharmonic spline does not preserve the monotonicity and convexity properties of the initial data. On the other hand the discrete biharmonic tension spline in Fig. 6 preserves the data shape and gives a visually smooth surface.

The exponential function

$$
\begin{aligned}
f(x,y) = & \frac{3}{4} e^{-\frac{1}{4}[(9x-2)^2+(9y-2)^2]} + \frac{3}{4} e^{-[\frac{1}{49}(9x+1)^2+\frac{1}{10}(9y+1)]} \\
& - \frac{1}{5} e^{-[(9x-4)^2+(9y-7)^2]} + \frac{1}{2} e^{-\frac{1}{4}[(9x-7)^2+(9y-3)^2]}
\end{aligned}
\tag{41}
$$

has been used in [5,7] to obtain a scattered data. A graph of the function (41) with the data points marked by circles is shown in Fig. 7. A projection of the data points on xy plane and a surface obtained by joining the data points by pieces of straight lines are given in Fig. 8. Fig. 9 presents the resulting biharmonic surface under tension.

The initial topographical data in the next test is shown in Fig. 10. Fig. 11 is obtained by setting all tension parameters to zero, that is, considering usual discrete biharmonic spline interpolating the data. It gives oscillations which are unnatural for the data. The situation can be substantially improved by using biharmonic tension spline with automatic selection of the shape control parameters. The resulting discrete tension spline in Fig. 12 has no oscillations and simultaneously keeps a visually smooth surface.

A reconstruction of the jet's surface is shown in Figs. 13–15. The initial data was defined as a set of 16 pointwise-assigned non-intersecting and in general curvilinear sections of a 3-D body. The number of points varied from section to section with a total of 212 points. Fig. 13 gives the initial data. Figs. 14 and 15 show the biharmonic surfaces without tension and with "optimal" tension parameters, respectively.

As a last numerical test, we tried to reconstruct the surface of a "Viking ship". The initial data, which the author obtained from Professor T. Lyche of the Oslo University, was defined pointwise in the form of the envelopes of the sides and the keel of the boat, as well as six ribs. 3-D view of the data is given in Fig. 16. In Figs. 17 and 18 the resulting biharmonic tension surface is given for very large and "optimal" tension parameters with a mesh of lines 100×100.

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.

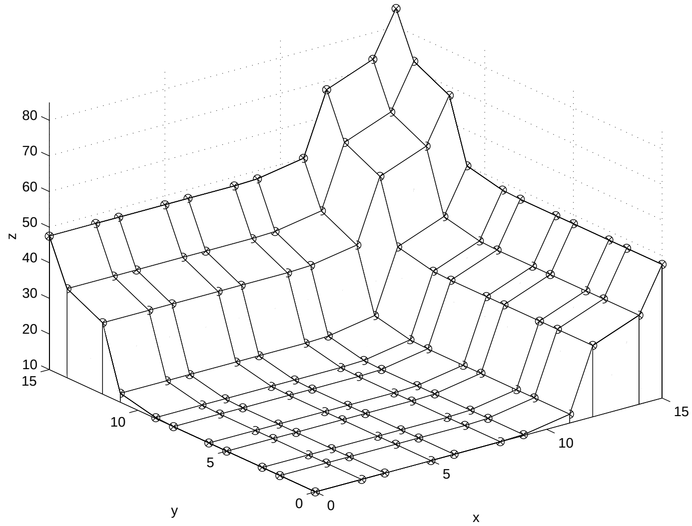

Fig. 4. The initial data.

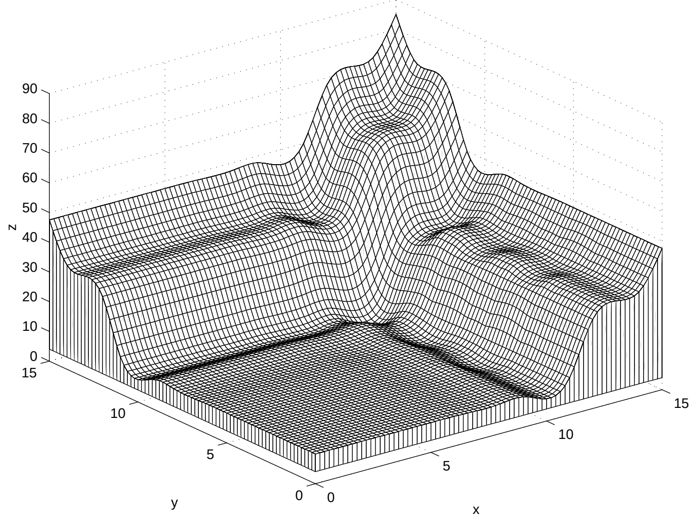

Fig. 5. The biharmonic spline interpolation.

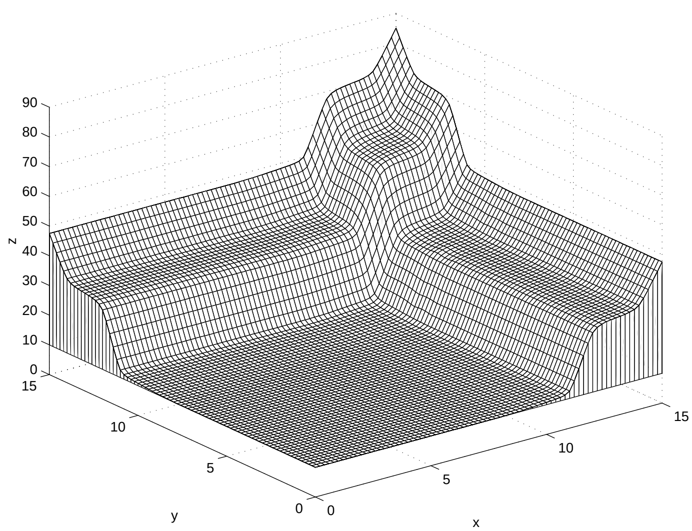

Fig. 6. The biharmonic tension spline interpolation.

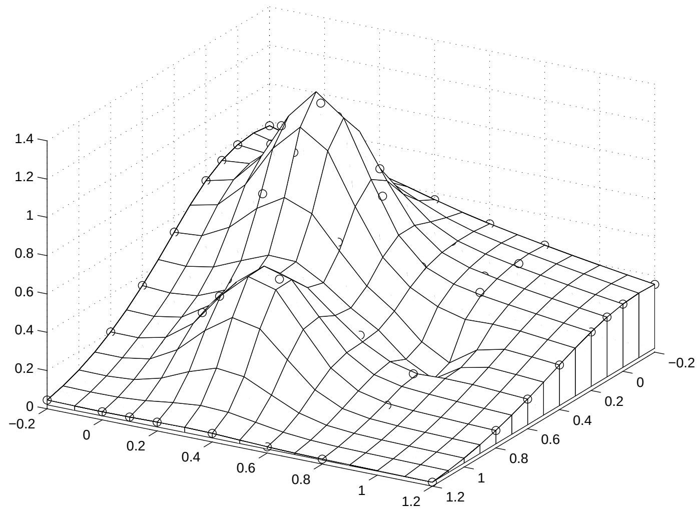

Fig. 7. A graph of the function (41) with the data points marked by circles.

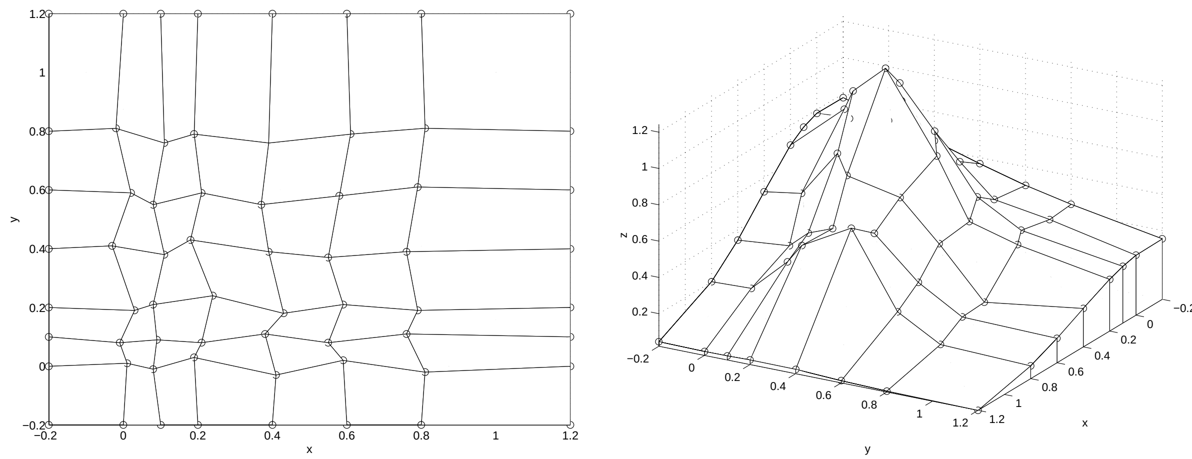

Fig. 8. The initial data. Left: a projection of the data points on the xy plane. Right: a surface obtained by joining the data points.

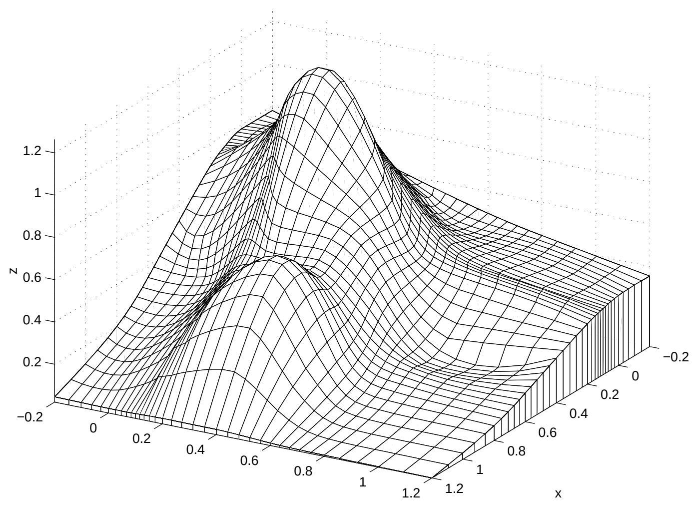

Fig. 9. The resulting biharmonic surface under tension.

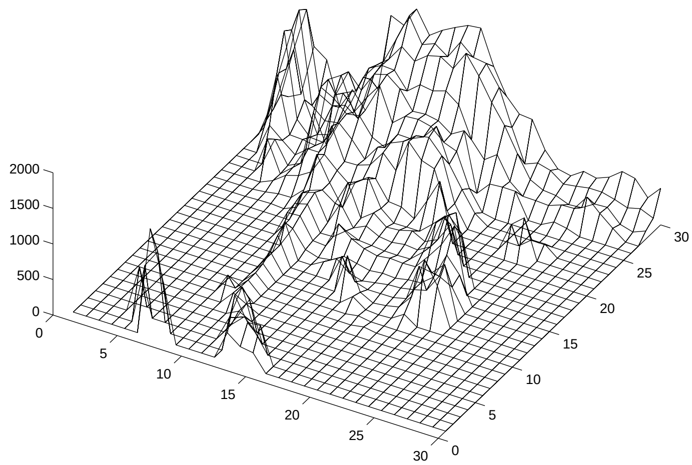

Fig. 10. A view of the initial topographical data.

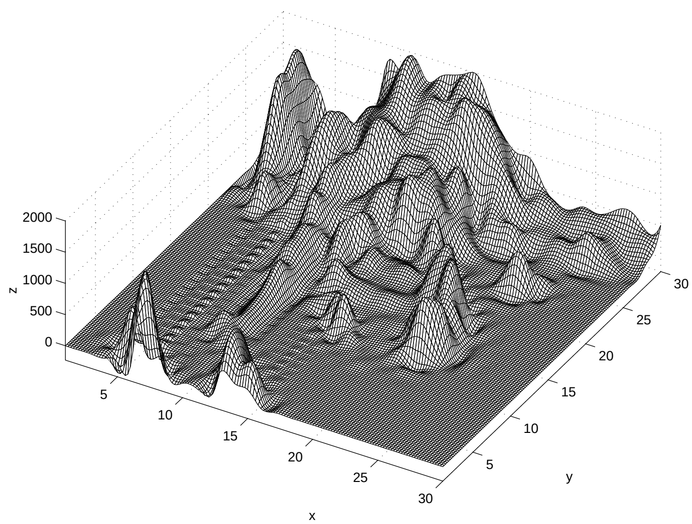

Fig. 11. A surface "without tension".

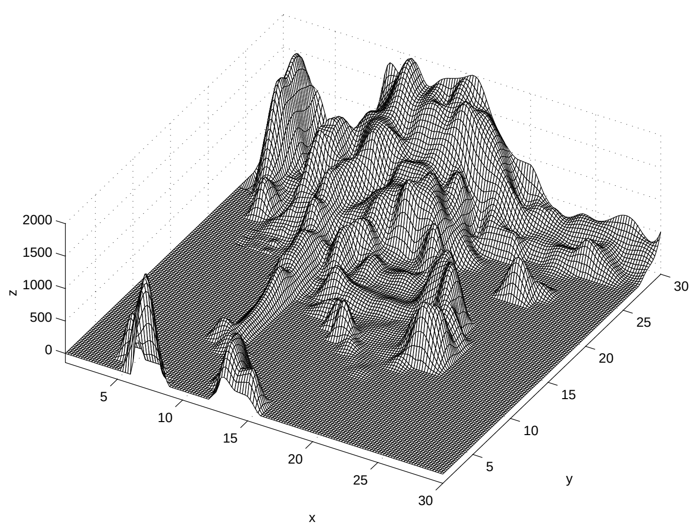

Fig. 12. The resulting surface under tension.

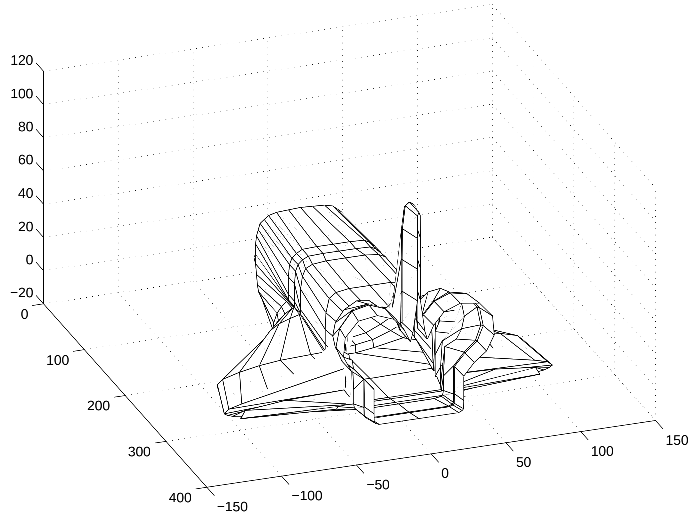

Fig. 13. The initial jet's data.

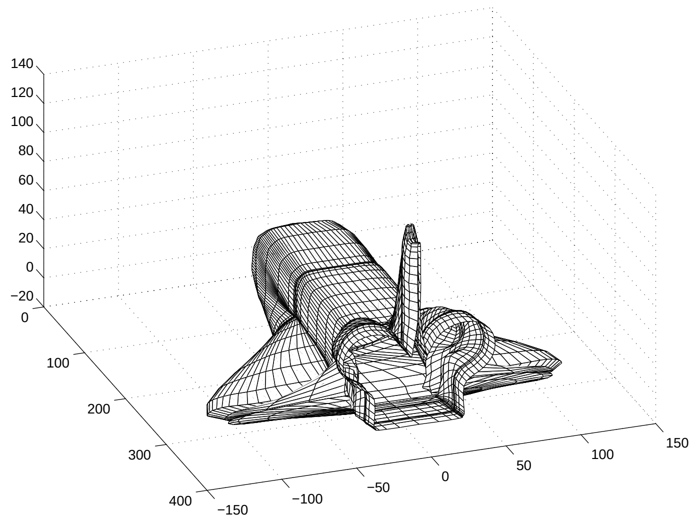

Fig. 14. The biharmonic surface without tension.

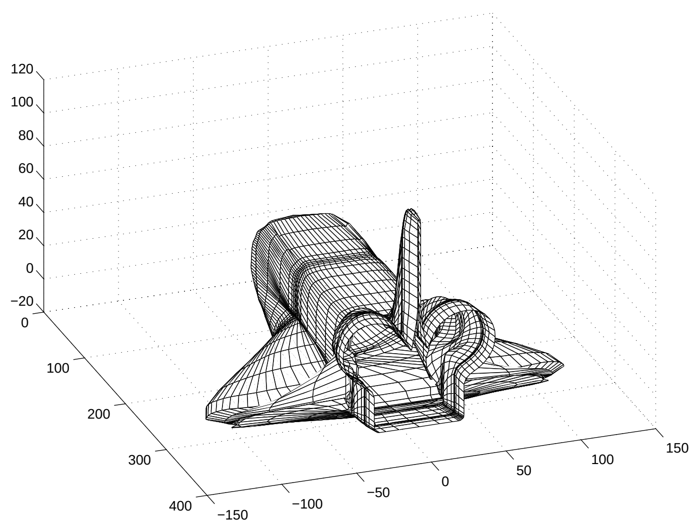

Fig. 15. The resulting surface under tension.

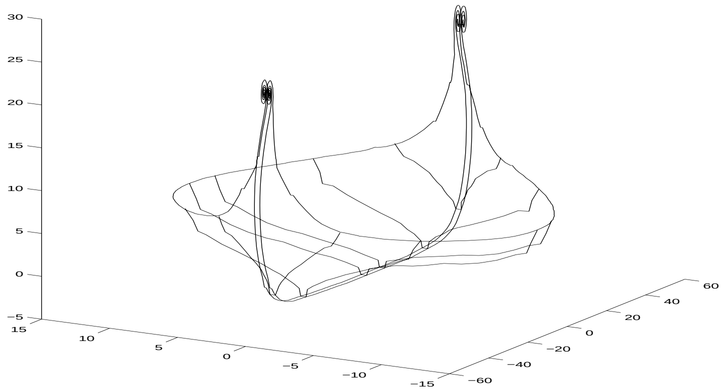

Fig. 16. 3-D view of the data.

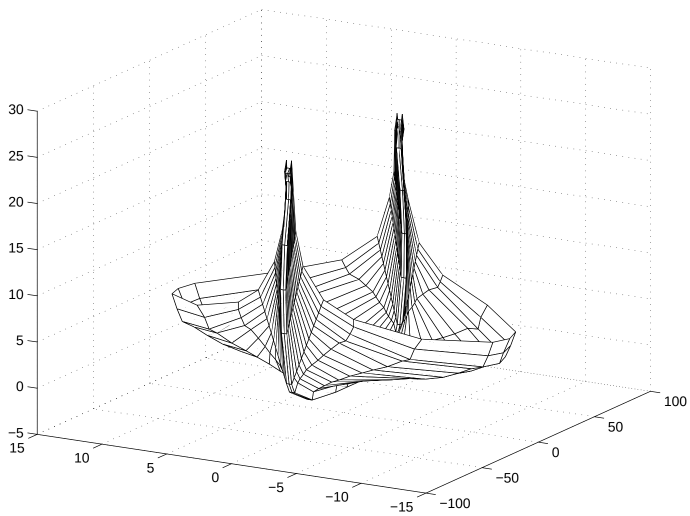

Fig. 17. The biharmonic surface for very large tension parameters.

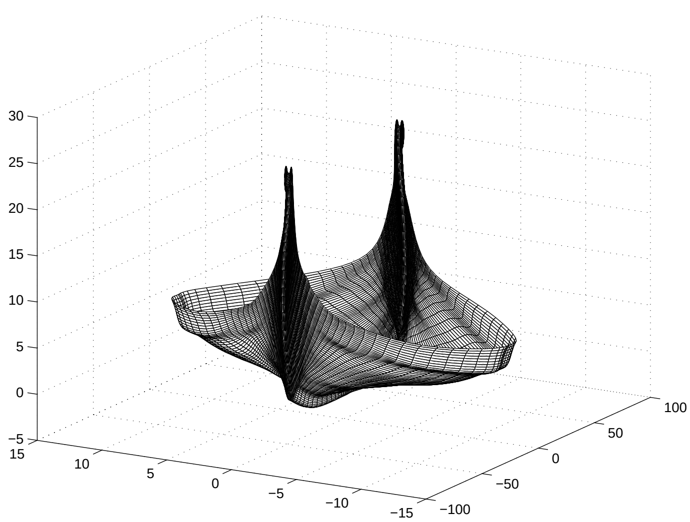

Fig. 18. The resulting biharmonic surface with "optimal" tension parameters.

Acknowledgments. Work partially supported by the Russian Academy of Sciences. Appreciation is also rendered to the Russian Foundation for Basic Research for the financial support which has made this research possible.

References

1. Akima, H., A new method of interpolation and smooth curve fitting based on local procedures, J. Assoc. Comput. Mech. **17** (1970), 589–602.
2. Bouhamedi, A. and Le Méhauté, A., Spline curves and surfaces with tension, *Wavelets, Images, and Surface Fitting*, Laurent, P. J., Le Méhauté, A., and Schumaker, L. L. (eds.), A K Peters, Wellesley, MA, pp. 51–58, 1994.
3. Costantini, P., Kvasov, B. I., and Manni, C., On discrete hyperbolic tension splines, Adv. Comput. **11** (1999), 331–354.
4. Duchon, J., Splines minimizing rotation invariant semi-norms in Sobolev spaces, *Constructive Theory of Functions of Several Variables*, Schempp, W. and Zeller, K. (eds.), Lecture Notes in Mathematics, Vol. 571, Springer, pp. 85–100, 1977.
5. Franke, R., Thin plate splines with tension, *Surfaces in CAGD'84*, Barnhill, R. E. and Böhm, W. (eds.), North-Holland, pp. 87–95, 1985.
6. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, John Hopkins University Press, Baltimore, 1991.
7. Hoschek, J. and Lasser, D., *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley, MA, 1993.
8. Janenko, N. N. and Kvasov, B. I., An iterative method for construction of polycubic spline functions, Soviet Math. Dokl. **11** (1970), 1643–1645.
9. Koch, P. E. and Lyche, T., Interpolation with Exponential B-splines in Tension, in: *Geometric Modelling*, Computing/Supplementum 8. Farin G. et al. (eds.), Springer-Verlag, Wien, pp. 173–190, 1993.
10. Kounchev, O., *Multivariate Polysplines: Applications to Numerical and Wavelet Analysis*, Academic Press, San Diego, 2001.
11. Kvasov, B. I., *Methods of Shape-Preserving Spline Approximation*, World Scientific Publ. Co. Pte. Ltd., Singapore, 2000.
12. Kvasov, B. I., On interpolating thin plate tension splines, *Curve and Surface Fitting: Saint-Malo 2002*, Cohen, A., Merrien, J.-L., and Schumaker, L. L., Nashboro Press, Brentwood, 2003, pp. 239–248.
13. Laurent, P. J., *Approximation et Optimization*, Hermann, Paris, 1972.

14. Malcolm, M. A., On the computation of nonlinear spline functions, SIAM J. Numer. Anal. **14** (1977), 254–282.

15. Marušić, M. and Rogina, M., Sharp error bounds for interpolating splines in tension, J. of Comp. Appl. Math. **61** (1995), 205–223.

16. Renka, R. J., Interpolation tension splines with automatic selection of tension factors, SIAM J. Sci. Stat. Comp. **8** (1987), 393–415.

17. Rentrop, P., An algorithm for the computation of exponential splines, Numer. Math. **35** (1980), 81–93.

18. Sapidis, N. S. and Kaklis, P. D., An algorithm for constructing convexity and monotonicity-preserving splines in tension, Computer Aided Geometric Design **5** (1988), 127–137.

19. Schumaker, L. L., *Spline Functions: Basic Theory*, John Wiley & Sons, New York, 1981.

20. Schweikert, D. G., An interpolating curve using a spline in tension, J. Math. Phys. **45** (1966), 312–317.

21. Yanenko, N. N., *The Method of Fractional Steps*, Springer Verlag, New York, 1971.

22. Zav'yalov, Yu. S., Kvasov, B. I., and Miroshnichenko, V. L., *Methods of Spline Functions*, Nauka, Moscow, 1980 (in Russian).