On Shape Preserving Thin Plate Splines

Boris I. Kvasov

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 thin plate tension spline we consider its finite-difference approximation. The resulting system of linear equations can be efficiently solved by successive over-relaxation (SOR) iterative method or using finite-difference schemes in fractional steps. We consider the basic computational aspects and illustrate the main advantages of this original approach.

§1. Introduction

Spline theory is mainly grounded on two approaches: the algebraic one (where splines are understood as smooth piecewise functions, see, e.g., [8]) and the variational one (where splines are obtained via minimization of quadratic functionals with equality and/or inequality constraints, see, e.g., [5]). Although less common, a third approach [3], where splines are defined as the solutions of differential multipoint boundary value problems (DMBVP for short), has been considered in one-dimesional case in [1,4]. 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 example of shape preserving thin plate tension splines.

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 SOR iterative method or by applying a finite-difference scheme in fractional steps [9]. We present a numerical example illustrating the main features of this approach.

Curve and Surface Fitting: Saint-Malo 2002
XXX (eds.), pp. 1–10.

ISBN XXX.

§2. 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

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

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

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

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

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

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. 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 thin plate tension spline.

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

$$\frac{\partial^4 S}{\partial x^4} + 2\frac{\partial^4 S}{\partial x^2 \partial y^2} + \frac{\partial^4 S}{\partial y^4} - \left(\frac{p_{ij}}{h_i}\right)^2 \frac{\partial^2 S}{\partial x^2} - \left(\frac{q_{ij}}{l_j}\right)^2 \frac{\partial^2 S}{\partial y^2} = 0 \qquad (1)$$

$$\text{in each } \Omega_{ij}, \quad h_i = x_{i+1} - x_i, \; l_j = y_{j+1} - y_j, \quad i = 0, \dots, N, \; j = 0, \dots, M,$$

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

$$\frac{\partial^4 S}{\partial y^4} - \left(\frac{q_{ij}}{l_j}\right)^2 \frac{\partial^2 S}{\partial y^2} = 0, \quad x = x_i, \quad y \in (y_j, y_{j+1}), \quad i = 0, \dots, N+1, \; j = 0, \dots, M, \tag{3}$$

$$S \in C^{2,2}[\overline{\Omega}], \tag{4}$$

with the interpolation conditions

$$S(x_i, y_j) = f_{ij}, \quad i = 0, \dots, N+1, \quad j = 0, \dots, M+1, \tag{5}$$

and the boundary conditions

$$\begin{aligned} S^{(2,0)}(x_i, y_j) &= f_{ij}^{(2,0)}, \quad i = 0, N+1, \quad j = 0, \dots, M+1, \\ S^{(0,2)}(x_i, y_j) &= f_{ij}^{(0,2)}, \quad i = 0, \dots, N+1, \quad j = 0, M+1, \\ S^{(2,2)}(x_i, y_j) &= f_{ij}^{(2,2)}, \quad i = 0, N+1, \quad j = 0, M+1. \end{aligned} \tag{6}$$

If all tension parameters of the thin plate tension spline S in (1)–(6) are zero then one obtains a smooth thin plate spline [2], interpolating the data (x_i, y_j, f_{ij}), $i = 0, \dots, N$, $j = 0, \dots, M$. If the tension parameters p_{ij} and q_{ij} approach infinity then in the rectangles $\overline{\Omega}_{ij}$, $i = 0, \dots, N$, $j = 0, \dots, M$, the thin plate tension spline S turns into a linear function separately by x and y, and obviously preserves the shape properties of the data on $\overline{\Omega}_{ij}$. So, by changing values of the shape control parameters p_{ij} and q_{ij} one can preserve various characteristics of the data including positivity, monotonicity, convexity, as well as linear and planar sections. By increasing one or more of these parameters the surface is pulled towards an inherent shape while at the same time keeping its smoothness. Thus, the DMBVP gives a reasonable mathematical formulation of the shape preserving interpolation problem.

§3. 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 it can be extended on $\overline{\Omega}$ to a smooth function which has shape properties very similar to those of S.

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

$$\left[\Lambda_1^2 + 2\Lambda_1\Lambda_2 + \Lambda_2^2 - \left(\frac{p_{ij}}{h_i}\right)^2 \Lambda_1 - \left(\frac{q_{ij}}{l_j}\right)^2 \Lambda_2\right] u_{ik;jl} = 0, \tag{7}$$
$$k = 1,\dots,n_i-1,\ i = 0,\dots,N;\ l = 1,\dots,m_j-1,\ j = 0,\dots,M,$$

$$\left[\Lambda_1^2 - \left(\frac{p_{ij}}{h_i}\right)^2 \Lambda_1\right] u_{ik;jl} = 0, \tag{8}$$

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

$$\left[\Lambda_2^2 - \left(\frac{q_{ij}}{l_j}\right)^2 \Lambda_2\right] u_{ik;jl} = 0, \tag{9}$$

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

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 condition (4) is changed to

$$\begin{aligned} u_{i-1,n_{i-1};jl} &= u_{i0;jl}, \\ \frac{u_{i-1,n_{i-1}+1;jl} - u_{i-1,n_{i-1}-1;jl}}{2h} &= \frac{u_{i1;jl} - u_{i,-1;jl}}{2h}, \\ \Lambda_1 u_{i-1,n_{i-1};jl} &= \Lambda_1 u_{i0;jl}, \end{aligned} \tag{10}$$

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

$$\begin{aligned} u_{ik;j-1,m_{j-1}} &= u_{ik;j0}, \\ \frac{u_{ik;j-1,m_{j-1}+1} - u_{ik;j-1,m_{j-1}-1}}{2h} &= \frac{u_{ik;j1} - u_{ik;j,-1}}{2h}, \\ \Lambda_2 u_{ik;j-1,m_{j-1}} &= \Lambda_2 u_{ik;j0}, \end{aligned} \tag{11}$$

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

Conditions (5) and (6) take the form

$$\begin{aligned} u_{i0;j0} &= f_{ij}, & u_{N,n_N;j0} &= f_{N+1,j}, \\ u_{i0;M,m_M} &= f_{i,M+1}, & u_{N,n_N;M,m_M} &= f_{N+1,M+1}, \end{aligned} \tag{12}$$

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

and

$$\begin{aligned} \Lambda_1 u_{00;j0} &= f^{(2,0)}_{0j}, \quad j = 0, \dots, M; & \Lambda_1 u_{00;M,m_M} &= f^{(2,0)}_{0,M+1}, \\ \Lambda_1 u_{N,n_N;j0} &= f^{(2,0)}_{N+1,j}, \quad j = 0, \dots, M; & \Lambda_1 u_{N,n_N;M,m_M} &= f^{(2,0)}_{N+1,M+1}, \\ \Lambda_2 u_{i0;00} &= f^{(0,2)}_{i0}, \quad i = 0, \dots, N; & \Lambda_2 u_{N,n_N;00} &= f^{(0,2)}_{N+1,0}, \\ \Lambda_2 u_{i0;M,m_M} &= f^{(0,2)}_{i,M+1}, \quad i = 0, \dots, N; & \Lambda_2 u_{N,n_N;M,m_M} &= f^{(0,2)}_{N+1,M+1}, \\ \Lambda_1 \Lambda_2 u_{00;00} &= f^{(2,2)}_{00}, & \Lambda_1 \Lambda_2 u_{N,n_N;00} &= f^{(2,2)}_{N+1,0}, \\ \Lambda_1 \Lambda_2 u_{00;M,m_M} &= f^{(2,2)}_{0,M+1}, & \Lambda_1 \Lambda_2 u_{N,n_N;M,m_M} &= f^{(2,2)}_{N+1,M+1}. \end{aligned} \tag{13}$$

We can evaluate all tension parameters p_{ij}, q_{ij}, $i = 0, \dots, N$, $j = 0, \dots, M$, by one of the algorithms for their automatic selection, see, e.g., [6,7], etc., and first find the mesh solution on the main mesh Δ. The latter can be achieved in the three steps.

F i r s t s t e p. Construct discrete hyperbolic tension splines [1] in the x direction by solving the $M + 2$ systems (8). As a result, one finds the values of the mesh solution on the lines of the mesh Δ in x direction.

S e c o n d s t e p. Construct discrete hyperbolic tension splines in the y direction by solving the $N + 2$ systems (9). This gives us the values of the mesh solution on the lines of the mesh Δ in y direction.

T h i r d s t e p. Construct discrete hyperbolic tension splines in the x and y directions interpolating the data $f^{(2,0)}_{ij}$, $i = 0, N+1$, $j = 0, \dots, M+1$, and $f^{(0,2)}_{ij}$, $i = 0, \dots, N+1$, $j = 0, M+1$, on the boundary Γ. This gives us the values

$$\begin{aligned} &\Lambda_1 u_{00;jl}, \quad \Lambda_1 u_{N,n_N;jl}, \quad l = 0, \dots, m_j, \quad j = 0, \dots, M, \\ &\Lambda_2 u_{ik;00}, \quad \Lambda_2 u_{ik;M,m_M}, \quad k = 0, \dots, n_i, \quad i = 0, \dots, N. \end{aligned} \tag{14}$$

Now the system of difference equations (7)–(13) can be substantially simplified by eliminating the unknowns

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

using relations (10), (11), and the boundary values (14), and eliminating the known values of the mesh solution on the mesh Δ.

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

§4. SOR Algorithm

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

$$\{u^{(0)}_{ik;jl}\,|\,k=0,\dots,n_i,\ i=0,\dots,N,\ l=0,\dots,m_j,\ j=0,\dots,M\}. \quad (15)$$

In each rectangle Ω_{ij}, $i=0,\dots,N$, $j=0,\dots,M$, the difference equation (8) can be rewritten in componentwise form

$$\begin{aligned} u_{ik;jl} =& \frac{1}{\alpha_{ij}}\Big\{\beta_{ij}\left[u_{i,k-1;jl}+u_{i,k+1;jl}\right]+\gamma_{ij}\left[u_{ik;j,l-1}+u_{ik;j,l+1}\right] \\ &-2\left[u_{i,k-1;j,l-1}+u_{i,k-1;j,l+1}+u_{i,k+1;j,l-1}+u_{i,k+1;j,l+1}\right] \\ &-u_{ik;j,l-2}-u_{ik;j,l+2}-u_{i,k-2;jl}-u_{i,k+2;jl}\Big\}, \end{aligned} \quad (16)$$

where

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

Now using (16) we can write down SOR iterations to obtain a numerical solution on the refinement

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

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 (14) and are not part of the iterations.

§5. Method of Fractional Steps

The system of difference equations obtained in section 3 can be efficiently solved by the method of fractional steps [9]. Using the initial approximation (15) let us consider in each rectangle Ω_{ij}, $i=0,\ldots,N$, $j=0,\ldots,M$, the following splitting scheme

$$\begin{aligned} \frac{u^{n+1/2}-u^n}{\tau} + \Lambda_{11}u^{n+1/2} + \Lambda_{12}u^n = 0, \\ \frac{u^{n+1}-u^{n+1/2}}{\tau} + \Lambda_{22}u^{n+1} + \Lambda_{12}u^{n+1/2} = 0, \end{aligned} \tag{17}$$

where

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

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

Equations (17) can be rewritten in the form

$$\begin{aligned} (I+\tau\Lambda_{11})u^{n+1/2} =&(I-\tau\Lambda_{12})u^n, \\ (I+\tau\Lambda_{22})u^{n+1} =&(I-\tau\Lambda_{12})u^{n+1/2}, \end{aligned}$$

where I is an identity operator. Eliminating from here the fractional step $u^{n+1/2}$ yields

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

After some simple transformations we obtain the following scheme in whole steps, equivalent to the scheme (17),

$$\frac{u^{n+1}-u^n}{\tau} + (\Lambda_{11}+\Lambda_{22})u^{n+1} + 2\Lambda_{12}u^n + \tau(\Lambda_{11}\Lambda_{22}u^{n+1} - \Lambda_{12}^2u^n) = 0. \tag{18}$$

It follows from here that the scheme (18) and the equivalent scheme (17) for the tension thin plate equation (1) possess the property of complete approximation [9] 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 (17) or, which is equivalent, the scheme (18). Using usual harmonic analysis [9] assume that

$$u^n = \eta_n e^{i\pi z}, \quad u^{n+1/2} = \eta_{n+1/2}e^{i\pi z}, \quad z = k_1\frac{x-x_i}{h_i} + k_2\frac{y-y_j}{l_j}. \tag{19}$$

Substituting equations (19) into equations (17) we obtain the amplification factors

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

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

where

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

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

It follows from here that

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

for any τ. This proves the strong stability of the scheme (17).

At each fractional step in (17) one has to solve a linear system with a symmetric positive definite pentadiagonal matrix. This is much cheaper than directly solving the linear system (7). However, in general the scheme (17) has the property of incomplete approximation [9]. For this reason, in iterations we have to use small values of the iteration parameter τ, e.g., $\sqrt{\tau}/h^2 = const$.

§6. Numerical Example

The approach developed in this paper was tested on several practical examples. Because of space limitations we consider here only one such example. The initial topographical data is shown in Figure 1. Figure 2 is obtained by setting all tension parameters to zero, that is, considering usual thin plate spline interpolating the data. It gives oscillations which are unnatural for the data. The situation can be substantially improved by using thin plate tension spline with automatic selection of the shape control parameters. The resulting shape preserving spline in Figure 3 perfectly reproduces the data shape and simultaneously keeps a 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.

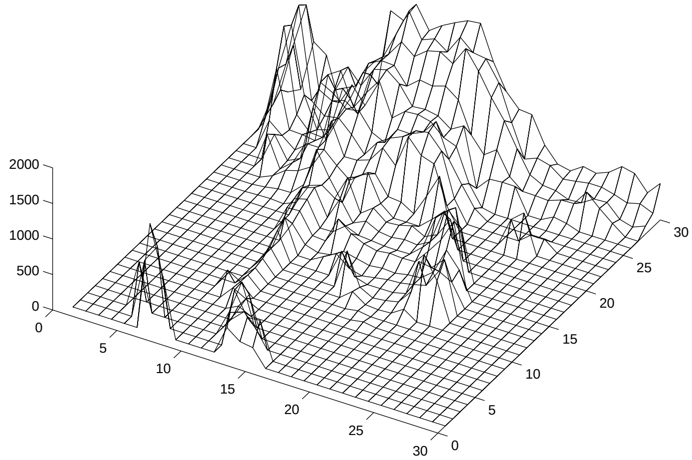

Fig. 1. A view of the initial topographical data.

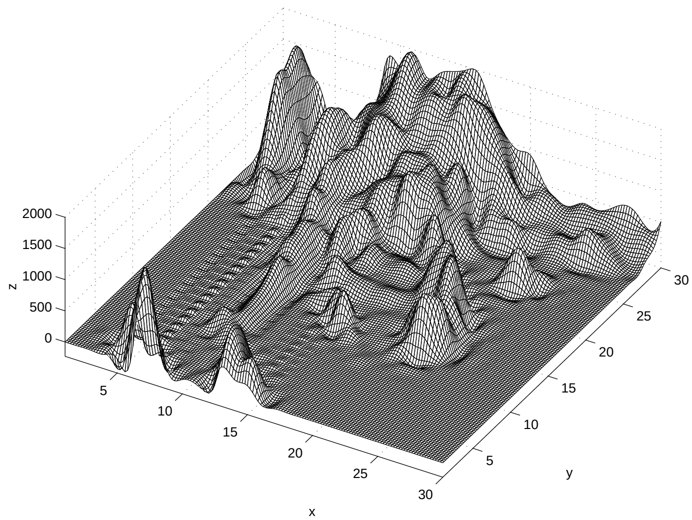

Fig. 2. A surface "without tension".

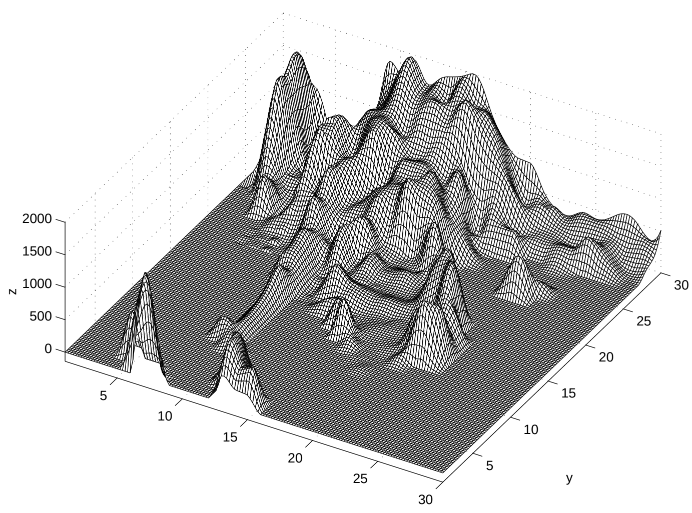

Fig. 3. The resulting shape preserving surface.

References

1. Costantini, P., B. I. Kvasov, and C. Manni, On discrete hyperbolic tension splines, Adv. Comput. **11** (1999), 331–354.
2. Hoschek, J. and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley, Massachusetts, 1993.
3. Janenko, N. N. and B. I. Kvasov, An iterative method for construction of polycubic spline functions, Soviet Math. Dokl. **11** (1970), 1643–1645.
4. Kvasov, B. I., *Methods of Shape-Preserving Spline Approximation*, World Scientific Publ. Co. Pte. Ltd., Singapore, 2000.
5. Laurent, P. J., *Approximation et Optimization*, Hermann, Paris, 1972.
6. Renka, R. J., Interpolation tension splines with automatic selection of tension factors, SIAM J. Sci. Statist. Comput. **8** (1987), 393–415.
7. Rentrop, P., An algorithm for the computation of exponential splines, Numer. Math. **35** (1980), 81–93.
8. Schumaker, L. L., *Spline Functions: Basic Theory*, John Wiley & Sons, New York, 1981.
9. Yanenko, N. N., *The Method of Fractional Steps*, Springer Verlag, New York, 1971.

Boris I. Kvasov
School of Mathematics
Suranaree University of Technology
Nakhon Ratchasima 30000
Thailand
boris@math.sut.ac.th