Far East J. Appl. Math. **26**(3) (2007), 361-373

SHAPE-PRESERVING SPLINE APPROXIMATION BY SIMPLE ITERATIVE METHOD

A. LUADSONG and O. SUTTISRI

Department of Mathematics
King Mongkut's University of Technology Thonburi (KMUTT)
126 Prachauthit Road, Bangmod
Thung-Kharu District, Bangkok 10140, Thailand
e-mail: anirut.lua@kmutt.ac.th
iornisri@yahoo.com

Abstract

In this paper, 2-D problem of shape-preserving splines is formulated as the Differential Multipoint Boundary Value Problem (DMBVP) for thin plate tension splines. For a numerical treatment of this problem, we replace the differential operator by its difference approximation. This gives us a system of linear equations with the matrix of a special structure. We found that this matrix is positive definite. Therefore, we can solve efficiently this system of linear equations by direct or iterative methods. For the required memory of this algorithm is about n and the computational time especially the operation count is about $O(n)$ for each iteration, where n is the number of unknowns in the interpolation problem.

1. Introduction

Spline functions constitute the main tool in computer aided geometric design (CAGD for short) which is concerned with the approximation and representation of curves and surfaces that arises when these objects have

2000 Mathematics Subject Classification: 41A51.

Keywords and phrases: shape-preserving spline, DMBVP, CAGD, finite-difference, SOR.

This research was supported by grants from KMUTT's Research Fund.

Received October 31, 2006

to be processed by a computer. Applications of CAGD include not only geometric design of different products like car bodies, ship hulls, airplane fuselages, etc., but also computer vision and inspection of manufactured parts, medical research (software for digital diagnostic equipment), image analysis, high resolution TV systems, cartography, etc. In the majority of these applications, it is important to construct curves and surfaces which preserve certain properties of the data. For example, we may want the surface to be positive, monotone, or convex in some sense. Standard methods of spline functions do not retain these properties of the data. This problem is known as *the problem of shape-preserving interpolation*. The purpose of this research is to develop new efficient methods for solving this problem.

In this paper, splines are defined as solutions of differential multipoint boundary value problems (DMBVP) [3]. This method has substantial computational advantages. We develop this approach on the examples of hyperbolic and thin plate tension splines. It can be generalized to smoothing splines and even to scattered data in a straightforward manner.

2. Problem Formulation

We introduce necessary notations and define 2-D problem of shape-preserving interpolation as DMBVP which consists of differential equation, smoothness, interpolation, and boundary conditions.

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

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

Γ is the boundary of Ω, and a rectangular mesh $\Delta = \Delta_x \times \Delta_y$ 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

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

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

We denote by $C^{2,2}[\overline{\Omega}]$ the set of all continuous on $\overline{\Omega}$ functions f having continuous partial and mixed derivatives up to the order 2. We say that the problem of searching for a function $S \in C^{2,2}[\overline{\Omega}]$ such that

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

and that S preserves the form of the initial data is the *shape-preserving interpolation problem*.

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 2.1. An interpolating thin plate tension spline S with two sets of tension parameters $\{p_{ij} \geq 0 \mid i = 0, \ldots, N;\ j = 0, \ldots, M\}$ and $\{q_{ij} \geq 0 \mid i = 0, \ldots, N;\ j = 0, \ldots, M\}$ is a solution of the DMBVP

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

$$\text{in each } \Omega_{ij},\ i = 0, \ldots, N;\ j = 0, \ldots, M, \tag{2}$$

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

$$i = 0, \ldots, N;\ j = 0, \ldots, M+1, \tag{3}$$

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

$$i = 0, \ldots, N+1; \; j = 0, \ldots, M, \tag{4}$$

$$S \in C^{2,2}[\Omega], \tag{5}$$

with the interpolation conditions

$$S(x_i, y_j) = f_{ij}, \quad i = 0, \ldots, N+1, \quad j = 0, \ldots, M+1, \tag{6}$$

and the boundary conditions

$$D^{(2,0)}S(x_i, y_j) = D^{(2,0)}f(x_i, y_j), \quad i = 0, N+1, \quad j = 0, \ldots, M+1,$$

$$D^{(0,2)}S(x_i, y_j) = D^{(0,2)}f(x_i, y_j), \quad i = 0, \ldots, N+1, \quad j = 0, M+1,$$

$$D^{(2,2)}S(x_i, y_j) = D^{(2,2)}f(x_i, y_j), \quad i = 0, N+1, \quad j = 0, M+1, \tag{7}$$

where $D^{(r,s)}f(x_i, y_j) = \dfrac{\partial^{r+s} f(x_i, y_j)}{\partial x^r \partial y^s}$; $r, s = 0, 2$.

If all tension parameters of the thin plate tension spline S are zero, then one obtains a smooth thin plate spline [2], interpolating the data (x_i, y_j, f_{ij}); $i = 0, \ldots, N$; $j = 0, \ldots, M$. If tension parameters p_{ij} and q_{ij} approach to the infinity, then in the rectangle $\overline{\Omega}_{ij}$; $i = 0, \ldots, N$; $j = 0, \ldots, M$; thin plate spline S turns into a linear function separately by x and y, and obviously preserves on $\overline{\Omega}_{ij}$ shape properties of the data. 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 at the same time keeping its smoothness. Thus, DMBVP gives a reasonable mathematical formulation of this problem.

3. Successive Over-Relaxation (SOR) Algorithm

For a numerical treatment of this problem, we replace the differential operator by its difference approximation. This gives us a system of linear equations with the matrix of a special structure. Since the coefficient matrix A of our linear system is symmetric positive definite matrix [5], iterative methods such as Jacobi, Gauss-Seidel, etc. should ever converge to the solution of this system. In general, the convergence of Gauss-Seidel method can be accelerated by using a relaxation parameter ω. This gives us successive over-relaxation (SOR for short) method which is chosen for solving the system of difference equations.

On the refinement, we define a mesh function

$$\{u_{ik;\,jl}^{(0)} \mid k = 1, \dots, n_i,\ i = 0, \dots, N,\ l = 1, \dots, m_j,\ j = 0, \dots, M\}$$

by a piecewise linear interpolation of the initial data either in x or y direction.

In each subdomain Ω_{ij}, $i = 0, \dots, N$, $j = 0, \dots, M$, difference equations from (2) can be rewritten in the componentwise form

$$u_{ik;\,j,l-2} + 2u_{i,k-1;\,j,l-1} - \gamma_{ij}u_{ik;\,j,l-1} + 2u_{i,k+1;\,j,l-1} + u_{i,k-2;\,jl}$$
$$- \beta_{ij}u_{i,k-1;\,jl} + \alpha_{ij}u_{ik;\,jl} - \beta_{ij}u_{i,k+1;\,jl} + u_{i,k+2;\,jl} + 2u_{i,k-1;\,j,l+1}$$
$$- \gamma_{ij}u_{ik;\,j,l+1} + 2u_{i,k+1;\,j,l+1} + u_{ik;\,j,l+2} = 0,$$

or

$$u_{ik;\,jl} = \frac{1}{\alpha_{ij}}\{\beta_{ij}[u_{i,k-1;\,jl} + u_{i,k+1;\,jl}] + \gamma_{ij}[u_{ik;\,j,l-1} + u_{ik;\,j,l+1}]$$
$$- 2[u_{i,k-1;\,j,l-1} + u_{i,k-1;\,j,l+1} + u_{i,k+1;\,j,l-1} + u_{i,k+1;\,j,l+1}]$$
$$- u_{ik;\,j,l-2} - u_{ik;\,j,l+2} - u_{i,k-2;\,jl} - u_{i,k+2;\,jl}\}, \tag{8}$$

where

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

Now to obtain a numerical solution on the refinement we apply in each subdomain Ω_{ij}, $i = 0, \dots, N$, $j = 0, \dots, M$, by using SOR method

$$\overline{u}_{ik;\,jl} = \frac{1}{\alpha_{ij}}\{\beta_{ij}[u^{(\nu+1)}_{i,\,k-1;\,jl} + u^{(\nu)}_{i,\,k+1;\,jl}] + \gamma_{ij}[u^{(\nu+1)}_{ik;\,j,\,l-1} + u^{(\nu)}_{ik;\,j,\,l+1}]$$

$$- 2[u^{(\nu+1)}_{i,\,k-1;\,j,\,l-1} + u^{(\nu)}_{i,\,k-1;\,j,\,l+1} + u^{(\nu+1)}_{i,\,k+1;\,j,\,l-1} + u^{(\nu)}_{i,\,k+1;\,j,\,l+1}]$$

$$- u^{(\nu+1)}_{ik;\,j,\,l-2} - u^{(\nu)}_{ik;\,j,\,l+2} - u^{(\nu+1)}_{i,\,k-2;\,jl} - u^{(\nu)}_{i,\,k+2;\,jl}\}, \tag{9}$$

$$u^{(\nu+1)}_{ik;\,jl} = u^{(\nu)}_{ik;\,jl} + \omega(\overline{u}_{ik;\,jl} - u^{(\nu)}_{ik;\,jl}), \tag{10}$$

where

$$1 < \omega < 2, \quad k = 1, \ldots, n_i - 1, \quad i = 0, \ldots, N,$$

$$l = 1, \ldots, m_j - 1, \quad j = 0, \ldots, M.$$

The formula (9) gives an approximation by Gauss-Seidel method while the next step (10) is used for the acceleration of the convergence.

Note that near the border of the domain Ω the extra unknowns $u_{0,\,-1;\,jl}$, $u_{N+1,\,1;\,jl}$, $j = 0, \ldots, M$, $l = 1, \ldots, m_j$, and $u_{ik;\,0,\,-1}$, $u_{ik;\,M+1,\,1}$, $i = 0, \ldots, N$, $k = 1, \ldots, n_i$ are eliminated using the boundary conditions of this problem and do not participate in the iterations. The above described approach can be formalized as the following algorithm.

Algorithm 3.1. SOR method for solving the system of difference equations.

Let u be the approximate solution of the original problem, ε be an error bound of exact solution of this system and ν be an iteration number.

1. Input the data

   $N, \; M, \; h, \; \omega, \; \varepsilon,$

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

   $f^{(2,\,0)}_{ij}, \quad i = 0, \ldots, N+1, \quad j = 0, \; M+1,$

   $f^{(0,\,2)}_{ij}, \quad i = 0, \; N+1, \quad j = 0, \ldots, M+1,$

   $f^{(2,\,2)}_{ij}, \quad i = 0, \; N+1, \quad j = 0, \; M+1.$

2. Calculate the following quantities:

$h_i,\ n_i,\quad i = 0, \ldots, N,$

$l_j,\ m_j,\quad j = 0, \ldots, M,$

α_{ij}, β_{ij}, and γ_{ij} for each $i = 0, \ldots, N,\ j = 0, \ldots, M.$

3. Define tension parameters $p_{ij},\ q_{ij},\ i = 0, \ldots, N,\ j = 0, \ldots, M$ by one of 1-D algorithms of shape-preserving interpolation.

4. Solve the difference equations from (3) and (4) as 1-D problems to find:

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

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

$$u_{ik;\, jl}^{(2,0)},\ k = \begin{cases} 0 & \text{if } i = 0, \\ n_N & \text{if } i = N, \end{cases} \quad l = 1, \ldots, m_j - 1,\ j = 0, \ldots, M,$$

$$u_{ik;\, jl}^{(0,2)},\ k = 1, \ldots, n_i - 1,\ i = 0, \ldots, N,\ l = \begin{cases} 0 & \text{if } j = 0, \\ m_M & \text{if } j = M. \end{cases}$$

5. Apply the boundary conditions to find

$u_{0,-1;\, jl},\quad u_{N,n_N+1;\, jl},\quad l = 1, \ldots, m_j,\ j = 0, \ldots, M;$

$u_{ik;\, 0,-1},\quad u_{ik;\, M, m_M+1},\quad k = 1, \ldots, n_i,\ i = 0, \ldots, N.$

6. Find the initial values

$$u_{ik,\, jl}^{(0)},\ k = 1, \ldots, n_i - 1,\ i = 0, \ldots, N,\ l = 1, \ldots, m_j - 1,\ j = 0, \ldots, M$$

by using a piecewise linear interpolation.

7. Make the iterations:

```
ν = 0
DO WHILE max_{i,k;j,l} | u^{(ν+1)}_{ik;jl} − u^{(ν)}_{ik;jl} | ≥ ε
   DO j = 0, ..., M
      DO i = 0, ..., N
         DO l = 1, ..., m_j − 1
            DO k = 1, ..., n_i − 1
               Find ū by the equation (9).
               Find u^{(ν+1)} by the equation (10).
            END DO
         END DO
      END DO
   END DO
   ν = ν + 1
END DO
```

8. Print the output

$$u^{(\nu)}_{00;00},\ u^{(\nu)}_{00;jl},\ u^{(\nu)}_{ik;00}, \text{ and } u^{(\nu)}_{ik;jl}$$

$$k = 1, \ldots, n_i, \quad i = 0, \ldots, N, \quad l = 1, \ldots, m_j, \quad j = 0, \ldots, M.$$

4. Numerical Experiments and Examples

The algorithms introduced in the previous sections work well on more general data than used in examples given below. If the algorithm fails in the data monotonicity and/or convexity on some intervals, then we have to increase the values of the corresponding tension parameters using algorithm of automatic selection of shape control parameters. This provides the properties of monotonicity and convexity for any data.

Example 4.1. We tried to reconstruct the surface by the data of Thailand's topography. The initial data was obtained from Mr. A. Boonleart of King Mongkut's University of Technology Thonburi, Bangkok. Three-dimensional view of the initial data is shown in Figure 1. Figure 2 was obtained by setting all tension parameters to zero. The surface in Figure 2 does not preserve the shape of the data. The new surface which preserves the shape of data was obtained by adjusting the tension parameters and shown in Figure 3.

Example 4.2. We tried to reconstruct the surface of test function with the scattered data,

$$f(x,\ y) = \frac{3}{4}e^{-\frac{1}{4}[(9x-2)^2+(9y-2)^2]} + \frac{3}{4}e^{-\left[\frac{1}{49}(9x+1)^2+\frac{1}{10}(9y+1)\right]}$$

$$-\frac{1}{5}e^{-[(9x-4)^2+(9y-7)^2]} + \frac{1}{2}e^{-\frac{1}{4}[(9x-7)^2+(9y-3)^2]}.$$

This function is well-known and usually used for testing each algorithm. Three-dimensional view of the test function was shown in Figure 4. Figures 5 and 6 were the initial grid and data points. Figure 7 was obtained by setting all tension parameters to zero. That is considering an approximation of the usual thin plate splines interpolating the data. Figure 8 was obtained by setting optimal values of all tension parameters. These two figures are quite similar.

5. Conclusion

This work aims to develop the new efficient algorithm for solving the problem of shape-preserving spline interpolation. SOR iterative method for the numerical solution of this problem is considered. In this algorithm, the computer time especially the operation count is around $18n$, i.e., $O(n)$, for each iteration, where n is the number of unknowns in the interpolation problem. The required memory is about n. The algorithm can be easily converted into a programming code.

The results of this work can be used in many applied problems and first of all in CAGD (design of curved and surfaces in the construction of different products such as car bodies, ship hulls, airplane fuselages, etc.). Other applications include the description of geological, physical and medical phenomena, image analysis, high resolution TV system, cartography, etc.

Acknowledgements

The author is very thankful to his mentor, B. I. Kvasov, who directed author's attention to spline functions, and whose influence contributed substantially towards the research interest and activities of the author in this direction of work. I would also like to thank my family, whose continuous encouragement and support sustained me through the preparation of this work.

References

[1] G. H. Golub and C. F. Van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, London, 1996.

[2] J. Hoschek and D. Lasser, Fundamentals of Computer Aided Geometric Design, A. K. Peters Ltd., Wellesley, Massachusetts, 1993.

[3] N. N. Janenko and B. I. Kvasov, An iterative method for construction of polycubic spline functions, Soviet Math. Dok. 11 (1970), 1643-1645.

[4] B. I. Kvasov, Methods of Shape-Preserving Spline Approximation, World Scientific Publ. Co. Pte., Singapore, 2000.

[5] A. Luadsong, The method for 2-D shape preserving interpolation problem, Proceedings of the 7th Annual National Symposium on Computational Science and Engineering (ANSCSE7), Chulalongkorn University, Bangkok, Thailand, March 24-26, 2003, pp. 351-360.

[6] N. N. Yanenko, The Method of Fractional Steps, Springer-Verlag, New York, Heidelberg, Berlin, 1971.

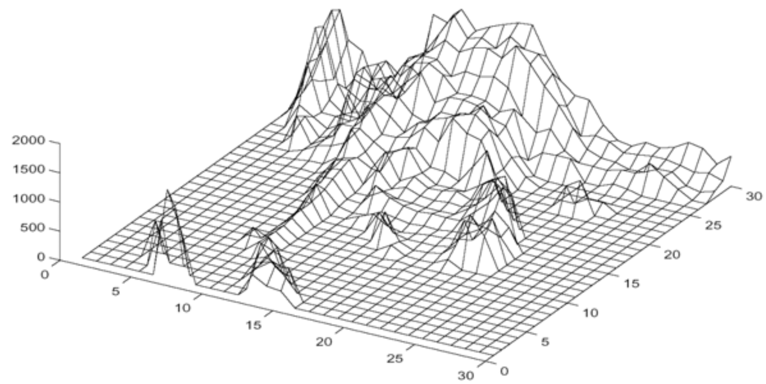

Figure 1. The initial data.

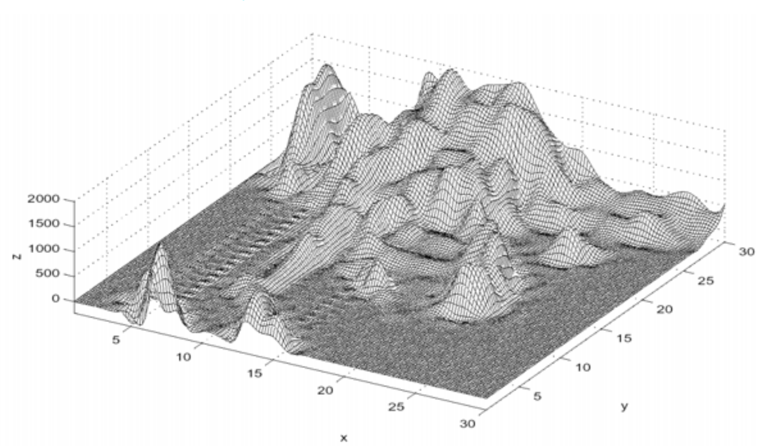

Figure 2. The surface with zero tension parameters.

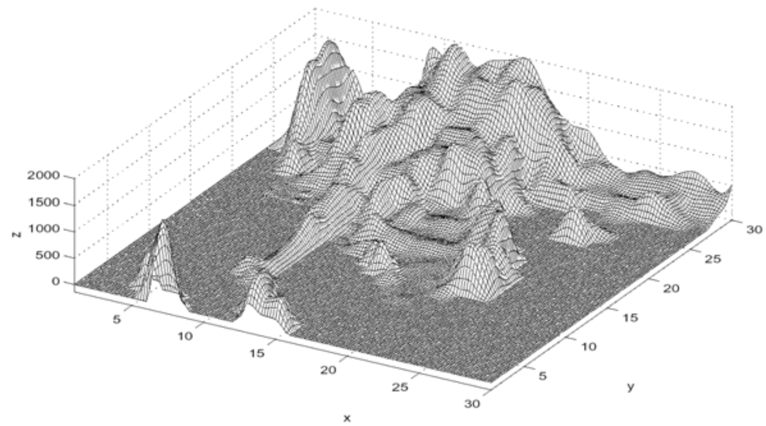

Figure 3. The surface with optimal tension parameters.

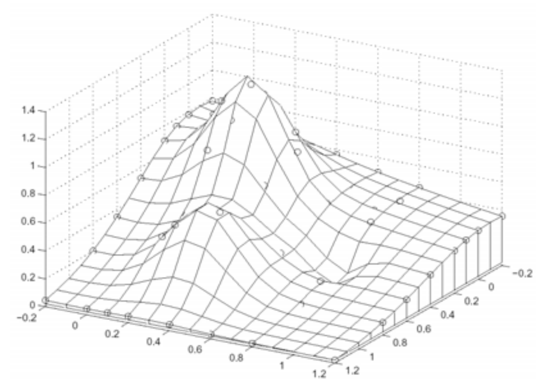

Figure 4. 3-D view of the test function.

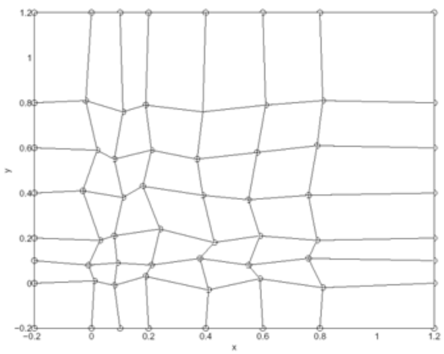

Figure 5. The data points.

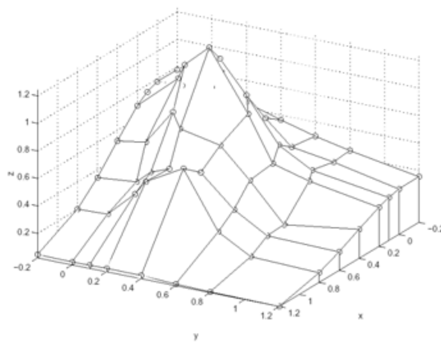

Figure 6. The initial data.

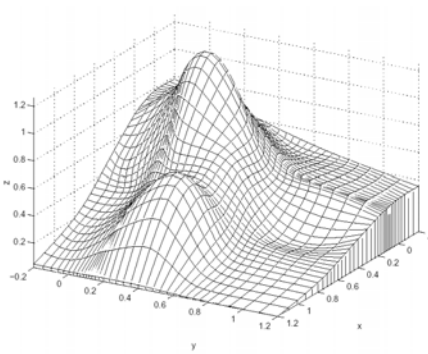

Figure 7. The surface with zero tension parameters.

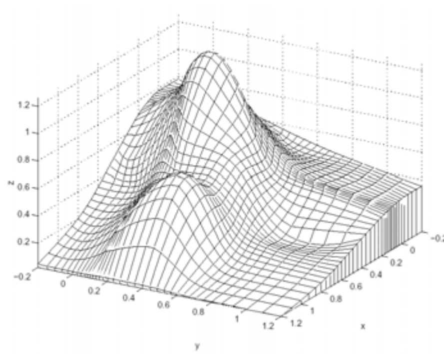

Figure 8. The surface with optimal tension parameters.

■