Interpolation of data in $\mathbb{R}^3$ using quartic triangular Bézier surfaces

Cite as: AIP Conference Proceedings **2325**, 020061 (2021); https://doi.org/10.1063/5.0040457
Published Online: 09 February 2021

Krassimira Vlachkova, and Krum Radev

Interpolation of Data in $\mathbb{R}^3$ using Quartic Triangular Bézier Surfaces

Krassimira Vlachkova[a)] and Krum Radev[b)]

Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski", 1164 Sofia, Bulgaria

[a)]*Corresponding author: krassivl@fmi.uni-sofia.bg*
[b)]*Electronic mail: kvradev@uni-sofia.bg*

Abstract. We consider the problem of interpolation of scattered data in $\mathbb{R}^3$ and propose a solution based on Nielson's minimum norm network and triangular Bézier patches. Our algorithm applies splitting to all triangles of an associated triangulation and constructs G^1-continuous bivariate interpolant consisting of quartic triangular Bézier patches. The algorithm is computationally simple and produces visually pleasant smooth surfaces. We have created a software package for implementation, 3D visualization and comparison of our algorithm and the known Shirman and Séquin's method which is also based on splitting and quartic triangular Bézier patches. The results of our numerical experiments are presented and analysed.

INTRODUCTION

Interpolation of data points in $\mathbb{R}^3$ by smooth surface is an important problem in applied mathematics which finds applications in various areas such as medicine, architecture, archeology, computer graphics, bioinformatics, scientific visualization, etc. In general the problem can be formulated as follows: Given a set of points $\mathbf{d}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, $i = 1, \ldots, n$, find a bivariate function $F(x, y)$ defined in a certain domain D containing points $\mathbf{v}_i = (x_i, y_i)$, such that F possesses continuous partial derivatives up to a given order and $F(x_i, y_i) = z_i$.

Various methods for solving this problem were proposed and applied, see [1, 2, 3, 4]. A standard approach to solve the problem consists of two steps, see [1]:

1. Construct a triangulation $T = T(\mathbf{v}_1, \ldots, \mathbf{v}_n)$;
2. For every triangle in T construct a surface (patch) which interpolates the data at the three vertices of T.

The interpolation surface constructed in Step 2 is usually polynomial or piecewise polynomial. Typically, the patches are computed with a priori prescribed normal vectors at the data points. G^1 or G^2 smoothness of the resulting surface is achieved either by increasing the degree of the patches, or by the so called *splitting*. Splitting was originally proposed by Clough and Tocher [5] and further developed by Percell [6] and Farin [7] for solving different problems. In practice using patches of least degree and splitting is preferable since it is computationally simple and efficient.

Shirman and Séquin [8, 9] construct a G^1 smooth surface consisting of quartic triangular Bézier surfaces (TBS). Their method assumes that the normal vectors at points $\mathbf{d}_i$, $i = 1, \ldots, n$, are given as part of the input. Shirman and Séquin construct a smooth cubic curve network defined on the edges of T, first, and then degree elevate it to quartic. This increases the degrees of freedom and allows them to connect smoothly the adjacent Bézier patches. Next, they apply splitting where for each triangle in T a macro-patch consisting of three quartic Bézier sub-patches is constructed. To compute the inner Bézier control points closest to the boundary of the macro-patch, Shirman and Séquin use a method proposed by Chiyokura and Kimura [10, 11]. The interpolation surfaces constructed by Shirman and Séquin's algorithm often suffer from unwanted bulges, tilts, and shears as pointed out by the authors in [12] and more recently by Hettinga and Kosinka in [13].

Nielson [14] proposes a method which computes a smooth interpolation curve network defined on the edges of T so as to satisfy an extremal property and then extends it to a smooth interpolation surface using an appropriate *blending* method based on convex combination schemes. The interpolation curve network is called *minimum norm network* (MNN) and is cubic. Nielson's interpolant is a rational function on every triangle in T. A significant advantage of Nielson's method is that the normal vectors at the data points are obtained through the computation of the MNN.

In this paper we propose an algorithm for interpolation of data in $\mathbb{R}^3$ which improves on Shirman and Séquin's approach in two ways. First, we use Nielson's MNN and second, we apply different strategy for computation of the control points. During the computation of control points closest to the boundary of a macro-patch we adopt additional criteria so that to avoid unwanted distortions and twists which appear in surfaces constructed by Shirman and Séquin's method. As a result, the quality of the resulting surfaces is improved. Furthermore, Shirman and Séquin impose an additional condition that the three quartic curves defined on the common edges of the three sub-patches are degree

International Conference on Analysis and Applied Mathematics (ICAAM 2020)
AIP Conf. Proc. 2325, 020061-1–020061-6; https://doi.org/10.1063/5.0040457
Published by AIP Publishing. 978-0-7354-4069-2/$30.00

elevated cubic curves. This condition is not necessary to obtain G^1-continuity across the common edges of the sub-patches. We apply different criterion for choosing the inner points of the sub-patches and believe that our choice would facilitate the construction of macro-patches that are convex in addition.

We implemented a software package for construction and 3D visualization of interpolation surfaces obtained by both Shirman and Séquin's algorithm and our algorithm. We tested the two algorithms extensively using data of increasing complexity and analysed the results with respect to different criteria.

RELATED WORK

Let $n \geq 3$ be an integer and $\mathbf{d}_i := (x_i, y_i, z_i),\ i = 1, \ldots, n$ be different points in $\mathbb{R}^3$. We call this set of points *data* and assume that the projections $\mathbf{v}_i := (x_i, y_i)$ onto the plane Oxy are different and non-collinear.

A *triangulation* T of points $\mathbf{v}_i$ is a collection of non-overlapping, non-degenerate closed triangles in Oxy such that the set of the vertices of the triangles coincides with the set of points $\mathbf{v}_i$. Hereafter we assume that a triangulation T of the points $\mathbf{v}_i,\ i = 1, \ldots, n$, is given and fixed.

Nielson's MNN

Furthermore, for the sake of simplicity, we assume that the domain D formed by the union of the triangles in T is connected. In general D is a collection of polygons with holes.

The set of the edges in T is denoted by E. If there is an edge between $\mathbf{v}_i$ and $\mathbf{v}_j$ in E, it will be referred to by e_{ij} or simply by e if no ambiguity arises.

A *curve network* is a collection of real-valued univariate functions $\{f_e\}_{e \in E}$ defined on the edges in E. With any real-valued bivariate function F defined on D we naturally associate the curve network defined as the restriction of F on the edges in E, i.e. for $e = e_{ij} \in E$,

$$f_e(t) := F\left(\left(1 - \frac{t}{\|e\|}\right)x_i + \frac{t}{\|e\|}x_j,\ \left(1 - \frac{t}{\|e\|}\right)y_i + \frac{t}{\|e\|}y_j\right), \text{ where } 0 \leq t \leq \|e\| \text{ and } \|e\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (1)$$

Furthermore, according to the context F will denote either a real-valued bivariate function or a curve network defined by (1). We introduce the following class of *smooth interpolants* defined on D

$$\mathscr{F} := \left\{F(x,y) \in C(D) \,|\, F(x_i, y_i) = z_i,\ i = 1, \ldots, n,\ \partial F/\partial x, \partial F/\partial y \in C(D),\ f'_e \in AC,\ f''_e \in L^2,\ e \in E\right\}$$

and the corresponding class of so-called *smooth interpolation curve networks*

$$\mathscr{C}(E) := \left\{F_{|E} = \{f_e\}_{e \in E} \mid F(x,y) \in \mathscr{F}\right\},$$

where $C(D)$ is the class of bivariate continuous functions defined in D, AC is the class of univariate absolutely continuous functions defined in $[0, \|e\|]$, and L^2 is the class of univariate functions defined in $[0, \|e\|]$ whose second power is Lebesgue integrable.

The smoothness of the interpolation curve network $F \in \mathscr{C}(E)$ geometrically means that at each point $\mathbf{d}_i$ there is a *tangent plane* to F, where a plane is *tangent* to the curve network at a point $\mathbf{d}_i$ if it contains the tangent vectors at $\mathbf{d}_i$ of the curves incident to $\mathbf{d}_i$.

The L^2-norm is defined in $\mathscr{C}(E)$ by $\|F\|_{L^2(T)} := \|F\| = \left(\sum_{e \in E} \int_0^{\|e\|} |f_e(t)|^2 dt\right)^{1/2}$. We denote the networks of the second derivative of F by $F'' := \{f''_e\}_{e \in E}$ and consider the following extremal problem:

$$(\mathbf{P}) \qquad \textit{Find } F^* \in \mathscr{C}(E) \ \textit{ such that } \|F^{*\prime\prime}\| = \inf_{F \in \mathscr{C}(E)} \|F''\|.$$

Nielson [14] showed that $(\mathbf{P})$ possesses a unique solution (MNN). The MNN is obtained by solving a linear system of equations.

Shirman and Séquin's method

This method assumes that the normal vectors at the data points $\mathbf{d}_i$, $i = 1,\dots,n$, are given a priori. First, Shirman and Séquin construct smooth interpolation cubic curve network defined on the edges of T which is compatible with the normal vectors given and degree elevate it to quartic. Then, for each triangle in T they apply splitting procedure which constructs three triangular Bézier patches (sub-patches) that possess a common vertex and form a G^1-continuous polynomial surface (macro-patch) defined in the triangle. The Bézier patches are computed through computation of their control points as shown in Fig. 1**a.** We use the following notation.

- ■ vertices of the sub-patches;
- ● inner control points on the boundary of the macro-patch;
- □ inner control points of the three inner boundary curves of the sub-patches;
- ○ inner control points of the sub-patches.

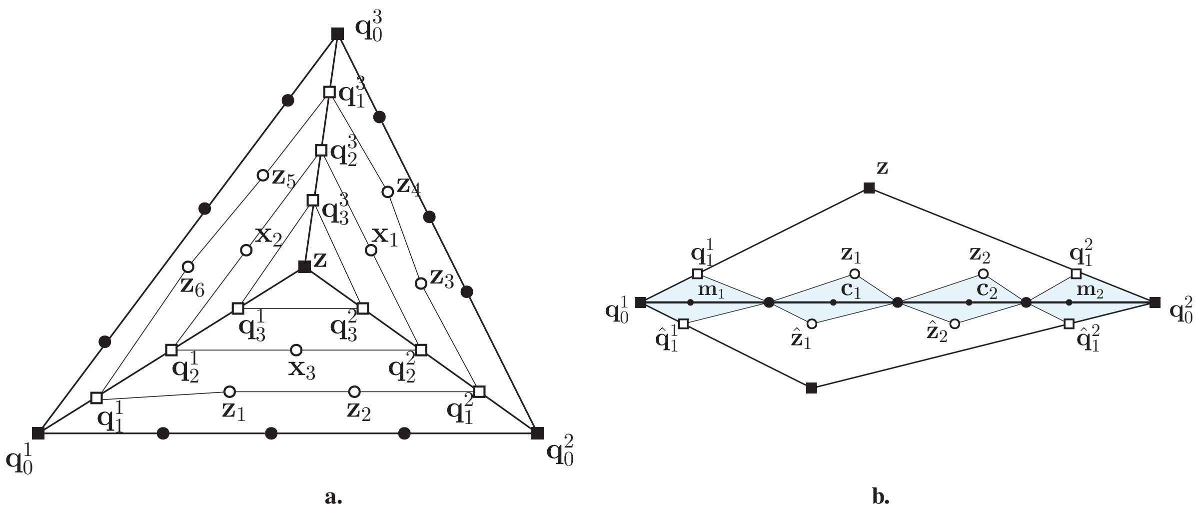


FIGURE 1. a. Construction of a G^1-continuous Bézier macro-patch by splitting to three sub-patches; **b.** A sufficient condition for G^1 continuity between $\mathscr{T}$ and its neighbouring patch is that the four shaded quadrilaterals are planar

Shirman and Séquin's Algorithm 1 below takes a triangle in T and the degree-elevated quartic boundary control points of the corresponding macro-patch and computes 19 control points of the three G^1-continuous quartic Bézier sub-patches.

Algorithm 1

Step 1. Compute the control points closest to the boundary of the macro-patch :
1.1 Points $\mathbf{q}_1^i,\ i = 1,2,3,$ are centers of the three small triangles with vertices ●■●.
1.2 Then points $\mathbf{z}_i,\ i = 1,\dots,6,$ are computed using Chiyokura and Kimura's method [10, 11].

Step 2. Compute points $\mathbf{q}_2^1 = \frac{1}{3}(\mathbf{q}_1^1 + \mathbf{z}_1 + \mathbf{z}_6),\ \mathbf{q}_2^2 = \frac{1}{3}(\mathbf{q}_1^2 + \mathbf{z}_2 + \mathbf{z}_3),\ \mathbf{q}_2^3 = \frac{1}{3}(\mathbf{q}_1^3 + \mathbf{z}_4 + \mathbf{z}_5).$

Step 3. Compute points $\mathbf{p}_i = 2\mathbf{q}_2^i - \frac{4}{3}\mathbf{q}_1^i + \frac{1}{3}\mathbf{q}_0^i,\ i = 1,2,3.$

Step 4. Compute the splitting point $\mathbf{z} = \frac{1}{3}(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3).$

Step 5. Compute points $\mathbf{q}_3^i = \frac{3}{4}\mathbf{p}_i + \frac{1}{4}\mathbf{z},\ i = 1,2,3.$

Step 6. Compute points $\mathbf{x}_1 = \frac{3}{2}(-\mathbf{q}_3^1 + \mathbf{q}_3^2 + \mathbf{q}_3^3) - \frac{1}{2}(-\mathbf{q}_2^1 + \mathbf{q}_2^2 + \mathbf{q}_2^3),\ \mathbf{x}_2 = \frac{3}{2}(\mathbf{q}_3^1 - \mathbf{q}_3^2 + \mathbf{q}_3^3) - \frac{1}{2}(\mathbf{q}_2^1 - \mathbf{q}_2^2 + \mathbf{q}_2^3),$
$\mathbf{x}_3 = \frac{3}{2}(\mathbf{q}_3^1 + \mathbf{q}_3^2 - \mathbf{q}_3^3) - \frac{1}{2}(\mathbf{q}_2^1 + \mathbf{q}_2^2 - \mathbf{q}_2^3).$

OUR ALGORITHM

In this section we propose our algorithm to construct G^1-continuous interpolation surface consisting of quartic TBP. First, we compute the MNN and then degree elevate it to quartic. Then for each triangle in T we apply splitting and construct three triangular Bézier sub-patches with a common vertex which form a G^1-continuous macro-patch defined in the triangle.

Let τ be a triangle in T and $\mathscr{T}$ be the corresponding macro-patch. We compute the control points of the three Bézier sub-patches of $\mathscr{T}$ consecutively in four layers as shown in Fig. 1**a.** The first layer consists of inner control points closest to the boundary of $\mathscr{T}$. The last fourth layer contains of single point $\mathbf{z}$. This point $\mathbf{z}$ is the splitting point and will be computed as a center of the triangle with vertices in the previous third layer.

Computing the control points in the first layer. There are three points of type □ and six points of type ○ in this layer, see Fig. 1**a.** First, we compute points of type □ as centers of the three small triangles with vertices ●■● on the boundary of the macro-patch. Then we compute the points $\mathbf{z}_i$, $i=1,2$, of type ○ as follows.

Let the corresponding edge of τ be inner for T and $\hat{\mathbf{q}}_1^i$, $\hat{\mathbf{z}}_i$, $i=1,2$, be the corresponding control points of the neighbouring patch, see Fig. 1**b.** A sufficient condition for G^1 continuity between the two patches is that the four shaded quadrilaterals in Fig. 1**b.** are planar. We compute the following points.

$$\begin{aligned}
&\mathbf{z}_i' = \mathbf{q}_1^1+\mathbf{c}_i-\mathbf{m}_1,\ \hat{\mathbf{z}}_i' = \hat{\mathbf{q}}_1^1+\mathbf{c}_i-\mathbf{m}_1,\ \mathbf{z}_i'' = \mathbf{q}_1^2+\mathbf{c}_i-\mathbf{m}_2,\ \hat{\mathbf{z}}_i'' = \hat{\mathbf{q}}_1^2+\mathbf{c}_i-\mathbf{m}_2,\\
&\mathbf{a}_i = (1-\frac{i}{3})\mathbf{z}_i'+\frac{i}{3}\mathbf{z}_i'',\ \hat{\mathbf{a}}_i = (1-\frac{i}{3})\hat{\mathbf{z}}_i'+\frac{i}{3}\hat{\mathbf{z}}_i'',\ i=1,2,
\end{aligned} \tag{2}$$

where $\mathbf{c}_i$, $i=1,2$, are control points of the cubic curve defined on the common edge of τ and its neighbouring triangle, and $\mathbf{m}_i$, $i=1,2$, are the intersection points of the diagonals of the quadrilaterals $\mathbf{q}_0^1\mathbf{q}_1^1\bullet\hat{\mathbf{q}}_1^1$ and $\bullet\mathbf{q}_1^2\mathbf{q}_0^2\hat{\mathbf{q}}_1^2$, respectively, as shown in Fig. 1**b.**

Let $\mathbf{z}_i=(\xi_i,\eta_i,\zeta_i)$ and $\hat{\mathbf{z}}_i=(\hat{\xi}_i,\hat{\eta}_i,\hat{\zeta}_i)$. We choose ξ_i,η_i and $\hat{\xi}_i,\hat{\eta}_i$ to be equal to the corresponding coordinates of $\mathbf{a}_i$ and $\hat{\mathbf{a}}_i$, respectively, $i=1,2$. In this way the projections of $\mathbf{z}_i$, $i=1,2$, onto Oxy lie inside τ. Hence, we avoid unwanted twisting and tilting of the patch. We compute the third coordinates ζ_i, $\hat{\zeta}_i$ so that $\zeta_i=\hat{\zeta}_i$ and $\mathbf{z}_i$, $\hat{\mathbf{z}}_i$, $\mathbf{c}_i$ to be collinear, $i=1,2$. In this way we avoid unwanted oscillations between the adjacent patches.

In the case where the corresponding edge of τ is boundary for T, i.e. there is no neighbouring patch of $\mathscr{T}$, we compute $\mathbf{z}_i$, $i=1,2$ as follows.

$$\mathbf{r}_i' = \mathbf{q}_1^1-\mathbf{q}_0^1+\mathbf{c}_i,\ \mathbf{r}_i'' = \mathbf{q}_1^2-\mathbf{q}_0^2+\mathbf{c}_i,\ \mathbf{z}_i = (1-\frac{i}{3})\mathbf{r}_i'+\frac{i}{3}\mathbf{r}_i'',\ i=1,2. \tag{3}$$

The rest of the points $\mathbf{z}_i$, $i=3,\ldots,6$, are computed analogously.

Computing the control points in the second and third layers. Now we already know the control points of type ■ and type ● on the boundary of the macro-patch and control points □,○ in the first layer. Let us consider an inner boundary curve, e.g. the curve with control points $\mathbf{q}_i^1,\mathbf{z}$, $i=0,\ldots,3$, see Fig. 1**a.** We have to compute the remaining control points so as to satisfy the G^1-continuity conditions across that curve. Hence, $\mathbf{q}_2^1$ is a center of the triangle $\mathbf{z}_1\mathbf{q}_1^1\mathbf{z}_6$, and $\mathbf{q}_3^1$ is a center of the triangle $\mathbf{x}_3\mathbf{q}_2^1\mathbf{x}_2$. Now suppose that we know points $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$ in the second layer. Then points $\mathbf{q}_3^1$,$\mathbf{q}_3^2$,$\mathbf{q}_3^3$ in the third layer are

$$\mathbf{q}_3^1 = (\mathbf{q}_2^1+\mathbf{x}_2+\mathbf{x}_3)/3,\ \mathbf{q}_3^2 = (\mathbf{q}_2^2+\mathbf{x}_1+\mathbf{x}_3)/3,\ \mathbf{q}_3^3 = (\mathbf{q}_2^3+\mathbf{x}_1+\mathbf{x}_2)/3.$$

Finally, the splitting point is $\mathbf{z} := \frac{1}{3}(\mathbf{q}_3^1+\mathbf{q}_3^2+\mathbf{q}_3^3)$.

It remains to compute the three points $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$. They provide three degrees of freedom which can be used for example to control the shape of the surface. In [8, 9] Shirman and Séquin use the condition that the three quartic curves defined on the inner edges are degree elevated cubic curves as it is for the boundary curves of the macro-patch. This condition is not necessary for G^1-continuity between two sub-patches and hence we do not need to impose it. We compute points $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$ as mid-points of the edges of the triangle $\mathbf{q}_2^1\mathbf{q}_2^2\mathbf{q}_2^3$ as follows.

$$\mathbf{x}_1 = (\mathbf{q}_2^2+\mathbf{q}_2^3)/2,\ \mathbf{x}_2 = (\mathbf{q}_2^1+\mathbf{q}_2^3)/2,\ \mathbf{x}_3 = (\mathbf{q}_2^1+\mathbf{q}_2^2)/2.$$

Then the control points in second, third, and fourth layers become co-planar. We believe that this choice of $\mathbf{x}_1$,$\mathbf{x}_2$,$\mathbf{x}_3$ will be useful when we try to construct a simple macro-patch which is convex in addition.

Algorithm 2 below takes a triangle τ in T and the degree-elevated quartic boundary control points of the corresponding patch $\mathscr{T}$ and computes 19 control points of the three G^1-continuous quartic Bézier sub-patches.

Algorithm 2

Step 1.	Compute the control points in the first layer: 1.1 Points of type □ are centers of the three small triangles with vertices ●■●. 1.2 Then points of type ○ are computed as described in (2) and (3).
Step 2.	Compute the control points in the second layer: 2.1 Points of type □ are centers of the three small triangles with vertices ○□○ in the first layer. 2.2 Then points of type ○ are mid-points of the segments with vertices of type □ in the second layer.
Step 3.	Compute the control points in the third layer: The three points of type □ are centers of the small triangles with vertices ○□○ in the second layer.
Step 4.	Compute the splitting point of type ■ as a center of the triangle with vertices □ in the third layer.

EXAMPLE

We implemented our Algorithm 2 as a web application using HTML, JavaScript, and the open source library `Plotly` [15]. The advantages of using `Plotly` are the options to display the coordinate system and the coordinates of the points when the cursor is placed on them, and to control the surfaces displayed. To demonstrate the results of our work we present here a simple example that clearly shows the typical differences between the surfaces produced by Shirman and Séquin's and our algorithms. As input for both algorithms we use the MNN for the data given.

Example 1 We consider data obtained from a regular triangular pyramid. We have $N = 4$, $\mathbf{v}_1 = (-1/2, -\sqrt{3}/6)$, $\mathbf{v}_2 = (1/2, -\sqrt{3}/6)$, $\mathbf{v}_3 = (0, \sqrt{3}/3)$, $\mathbf{v}_4 = (0,0)$, and $z_i = 0$, $i = 1,2,3$, $z_4 = -1$. The corresponding MNN is shown in Fig. 3. The triangulation and the corresponding Shirman and Séquin's surface are shown in Fig. 3**a.** The surface generated by our Algorithm 2 and the triangulation are shown in Fig. 3**b.**

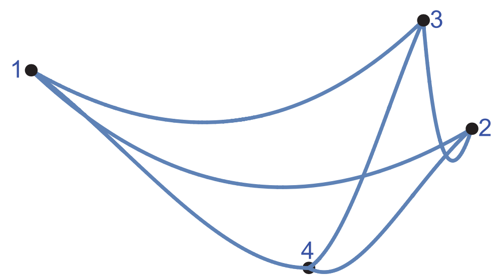


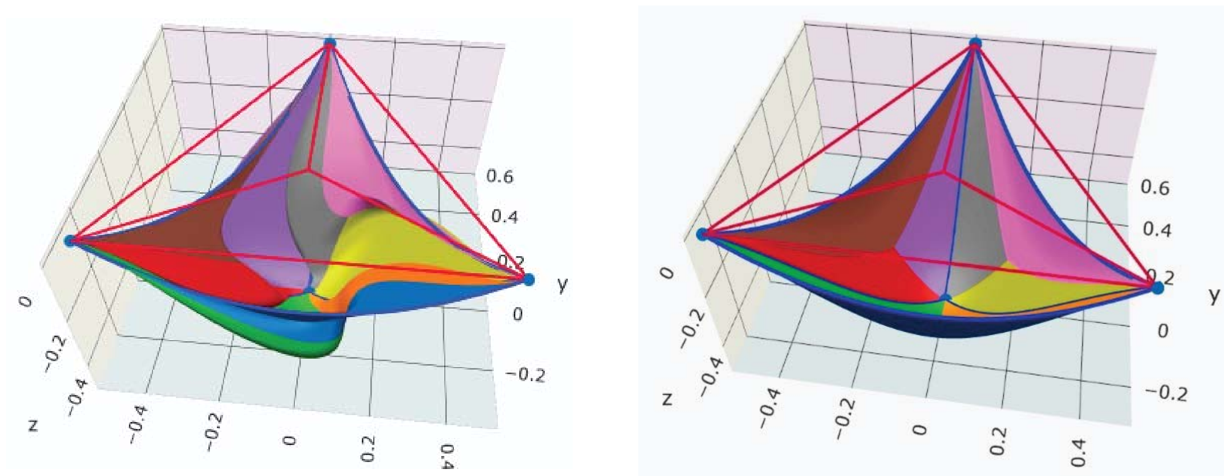
FIGURE 2. The MNN for the data in Example 1

CONCLUSION AND FUTURE WORK

In this paper we have proposed a method for constructing a G^1-continuous interpolation surface that consists of triangular quartic Bézier patches based on the MNN construction on the edges of the underlying triangulation. We have tested our method extensively and have witnessed various advantages over the existing methods. A promising approach for further improvement is to identify a smaller, possibly minimal, subset of T where splitting is to be done so that a smooth interpolant can be constructed. It is known that splitting decreases the smoothness of the resulting surface and hence, it would be advantageous to minimize the number of triangles where splitting is done.

ACKNOWLEDGMENTS

This research was supported by Sofia University Science Fund Grant No. 80-10-171/2020, and by European Regional Development Fund and the Operational Program "Science and Education for Smart Growth" under Contract BG05M2OP001-1.001-0004 (2018-2023).



a. b.

FIGURE 3. Comparison of the two surfaces for the data in Example 1: **a.** The surface generated using Algorithm 1 (Shirman and Séquin); **b.** The surface generated using our Algorithm 2

REFERENCES

1. S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan, "A survey of parametric scattered data fitting using triangular interpolants," in *Curve and Surface Design*, edited by H. Hagen (SIAM, Philadelphia, 1992) pp. 145–172.
2. T. K. Dey, *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*, Cambridge Monographs on Applied and Computational Mathematics (Cambridge University Press, 2006).
3. K. Anjyo, J. Lewis, and F. Pighin, "Scattered data interpolation for computer graphics, SIGGRAPH 2014 course notes," `http://olm.co.jp/rd/research_event/scattered-data-interpolation-for-computer-graphics` (Last accessed June 20, 2020).
4. M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, G. Guennebaud, J. Levine, A. Sharf, and C. Silva, Comput. Graph. Forum **36**, 301–329 (2017).
5. R. Clough and J. Tocher, "Finite elements stiffness matrices for analysis of plate bending," in *Proceedings of the 1st Conference on Matrix Methods in Structural Mechanics*, Vol. 66–80 (Wright-Patterson A. F. B., Ohio, 1965) pp. 515–545.
6. P. Percell, SIAM J. Numer. Anal. **13**, 100–103 (1976).
7. G. Farin, Comput. Aided Geom. Des. **2**, 19–27 (1985).
8. L. Shirman and C. Séquin, Comput. Aided Geom. Des. **4**, 279–295 (1987).
9. L. Shirman and C. Séquin, Comput. Aided Geom. Des. **8**, 217–221 (1991).
10. H. Chiyokura and F. Kimura, "Design of solids with free-form surfaces," in *SIGGRAPH '83 Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, Vol. 17, edited by P. P. Tanner (ACM, New York, 1983) pp. 289–298.
11. H. Chiyokura, "Localized surface interpolation method for irregular meshes," in *Advanced Computer Graphics, Proceedings of Computer Graphics Tokyo'86*, Vol. 66–80, edited by T. Kunii (Springer, Tokyo, 1986) pp. 3–19.
12. L. Shirman and C. Séquin, Comput. Aided Geom. Des. **7**, 375–388 (1990).
13. G. Hettinga and J. Kosinka, Comput. Aided Geom. Des. **62**, 166–180 (2018).
14. G. Nielson, Math. Comput. **40**, 253–271 (1983).
15. "Plotly.js," `https://plot.ly/javascript` (Last accessed June 20, 2020).