Cubic Bézier-Like Triangular Patches for Rainfall Scattered Data Interpolation and Visualization

Samsul Ariffin Abdul Karim[1,2,3*], *Mohammad Fadhli* Asli[4], *Chin Kim* On[5], *Ghulam* Mustafa[6], *Faheem* Khan[7], *Mohammad Khatim* Hasan[8], *Jumat* Sulaiman[9] and *Ahmed* Kherd[10]

[1]Software Engineering Programme, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia
[2]Data Technologies and Applications (DaTA) Research Lab, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia
[3]Creative Advanced Machine Intelligence (CAMI) Research Centre, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia
[4]Optimization and Visual Analytics (OVA) Research Lab, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Labuan International Campus, Jalan Sungai Pagar, 87000 Labuan Federal Territory, Malaysia
[5]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia
[6,7]Department of Mathematics, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[8]Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi, Selangor 43600, Malaysia
[9]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia
[10]AL-Ahgaff University, Faculty of Computer Science & Engineering, Department of Mathematics, Mukalla, Yemen

Abstract. Scattered data interpolation plays an important role in computer graphics and scientific visualization. This method can be used to interpolate any regular or irregular data sets. For instance, rainfall distribution is an example of an irregular data that usually appear in statistics. In this study, cubic Bézier-Like triangular patches defined on triangular domain is used to interpolate rainfall data sets. From graphical results, we obtain a very smooth surface. Therefore, the proposed scheme can be used to interpolate and provide a good visualization to the given irregular data sets.

1 Introduction

One of the active research topics in computer-aided design (CAD) and computer graphics (CG) is scattered data interpolation and approximation. These methods can be used to interpolate or approximate any regular or irregular data sets. There are many methods

* Corresponding author: samsulariffin.karim@ums.edu.my, drsamsul.karim@gmail.com

proposed in the literature. Either methods-based triangulations [1, 3, 4-6, 8, 12] or not involving any triangulations like meshfree methods such as radial basis functions (RBFs) including the famous Shepard's functions [2]. The scattered data methods must be able to produce smooth curves and surfaces everywhere. Basically, the main problem in $\boldsymbol{C^1}$ scattered data interpolation is described below [4, 6, 10, 11]:

Given the 3D functional data.

$$(x_i, y_i, z_i), \qquad i = 1,2,\dots,N$$

we wish to construct the smooth surface $\boldsymbol{z = F(x, y)}$ that satisfies the following condition:

$$z_i = F(x_i, y_i), \;\; i = 1,2,\dots,N$$

In the next Sections, we will discuss on how to solve the problem stated above.

2 Cubic Bézier-Like Triangular Patches

In this Section, we will provide brief explanation of the cubic Bézier-Like triangular patches. More details can be found in [4].

Definition 1 [4]. Let $a, b, c \in (0, \infty)$ with $u \geq 0, v \geq 0, w \geq 0$. The cubic Bézier-Like triangle basis on a triangular domain $D = \{(u, v, w) | u + v + w = 1\}$ is defined by :

$$\begin{cases} B^3_{3,0,0}(u,v,w) = u^2\big(1 + a(u-1)\big) \\ B^3_{0,3,0}(u,v,w) = v^2\big(1 + b(v-1)\big) \\ B^3_{0,0,3}(u,v,w) = w^2\big(1 + c(w-1)\big) \\ B^3_{2,1,0}(u,v,w) = (a+2)u^2v \\ B^3_{2,0,1}(u,v,w) = (a+2)u^2w \\ B^3_{1,2,0}(u,v,w) = (b+2)uv^2 \\ B^3_{0,2,1}(u,v,w) = (b+2)v^2w \\ B^3_{1,0,2}(u,v,w) = (c+2)uw^2 \\ B^3_{0,1,2}(u,v,w) = (c+2)vw^2 \\ B^3_{1,1,1}(u,v,w) = 6uvw \end{cases} \tag{1}$$

Definition 2 [4]. Given control points $\boldsymbol{b_{ijk}, i + j + k = 3}$ (See Figure 1) the cubic Bézier-Like triangular patches is defined as

$$P(u,v,w) = \textstyle\sum_{|i+j+k|=3} b_{ijk}\, B^3_{i,j,k}(u,v,w), \;\; u + v + w = 1 \tag{2}$$

which can be simplified to

$$P(u,v,w) = u^2\big(1 + a(u-1)\big)b_{300} + v^2\big(1 + b(v-1)\big)b_{030} + w^2\big(1 + c(w-1)\big)b_{003} + (a+2)u^2vb_{210} + (a+2)u^2wb_{201} + (b+2)v^2ub_{120} + (b+2)v^2wb_{021} + (c+2)w^2ub_{102} + (c+2)w^2vb_{012} + 6uvwb_{111} \tag{3}$$

where $u + v + w = 1$.

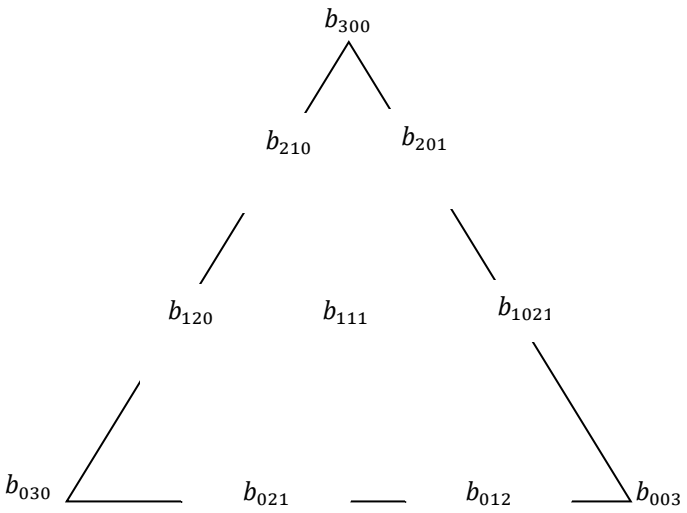

Fig. 1. The control points or ordinates for the Bézier-Like triangular

3 Scattered Data Interpolation Using Cubic Bézier-Like Triangular

In this Section we construct scattered data interpolation scheme by using the cubic Bézier-Like triangular patches developed in the previous Section. The scheme is comprising a convex combination blended between three local schemes P_1, P_2 and P_3 defined below:

$$P(u,v,w) = \frac{vwP_1+uwP_2+uvP_3}{vw+uw+uv},\ u+v+w=1 \tag{4}$$

The final C^1 scattered data interpolation scheme is defined as

$$P(u,v,w) = \sum_{\substack{i+j+k=3 \\ i,j,k\neq 1}} b_{ijk}\, B^3_{i,j,k}(u,v,w) + 6uvw\ \left(s_1 b^1_{111} + s_2 b^2_{111} + s_3 b^3_{111}\right) \tag{5}$$

with

$$s_1 = \frac{vw}{vw+uw+uv}, s_2 = \frac{uw}{vw+uw+uv}, cs_1 = \frac{uv}{vw+uw+uv} \tag{6}$$

where $b^d_{111}, d = 1,2,3$ are inner ordinates on each triangle; calculated by using the cubic precision method [4]. This method will ensure that the final interpolating surface has at least a cubic precision. This means that, the proposed scheme can reproduce exactly at most cubic polynomial [8]. Meanwhile, the remaining cubic Bézier-Like triangular ordinates are calculated by the method discussed in [4]. The following is an algorithm that can be used for the computer implementation.

Algorithm 1

Input : scattered data points and free parameters $a, b, c \in (0, \infty)$.
Output : smooth surface.
Step 1 : Triangulate the domain by using Delaunay triangulation [4].
Step 2 : Estimate the partial derivative at the data points by using [9].
Step 3 : Calculate the boundary control points using method developed in [4].

Step 4 : Calculate inner control points for the local scheme, $b_{111}^{d}, d = 1,2,3$ by using the cubic precision method [4].
Step 5 : Construct the interpolating surface using (5).
Step 6 : Calculate CPU time (in seconds).
Step 7 : Repeat Steps 1-6 with different free parameter values.

Now, we establish the main result of the present study. Let Ω represents the convex hull of points $\{(x_i, y_i)\}, i = 1,2,\ldots,N$ and let Δ isa triangulation of Ω. In order to obtain the following error bound, we have adopted the error bound established in [8].

Theorem 1. For every $f \epsilon C^4(\Omega)$, the C^1 Bézier-Like triangular interpolant P defined by (5) satisfies the following error bound

$$\|f - P\| \leq K|\Delta|^4 |f|_4$$

where the constant K depends on the smallest angle in Δ. Note that, $|\Delta|$ is the size of the triangulation, i.e. the Δ largest length of edges of Δ.

Based on Theorem 1 above, we found that our proposed scattered data interpolation scheme by using Bézier-Like triangular is the same as error bound by using rational quartic interpolant in [8]. The main advantage of the present scheme is that the interpolant is cubic polynomial meanwhile in [8], the interpolant is a rational quartic which will require tremendous computational times to execute and produce interpolating surface especially for large scattered data sets.

4 Results and Discussion

In this section, the scattered data interpolation scheme presented in the Section 3 is tested to interpolate the irregular rainfall data sets obtained from 25 major stations throughout Peninsular Malaysia [1] as shown in Figure 2. Each stations have different longitude and latitude. Figure 3 shows the Delaunay triangulation for the data. We implemented the proposed scheme using MATLAB 2022 version on AMD Ryzen 5 5600H with Graphics NIVIDIA GeForce RTX 3050 4.2 GHz (Laptop GPU). MATLAB coding was developed based on Algorithm 1 presented in Section 3.

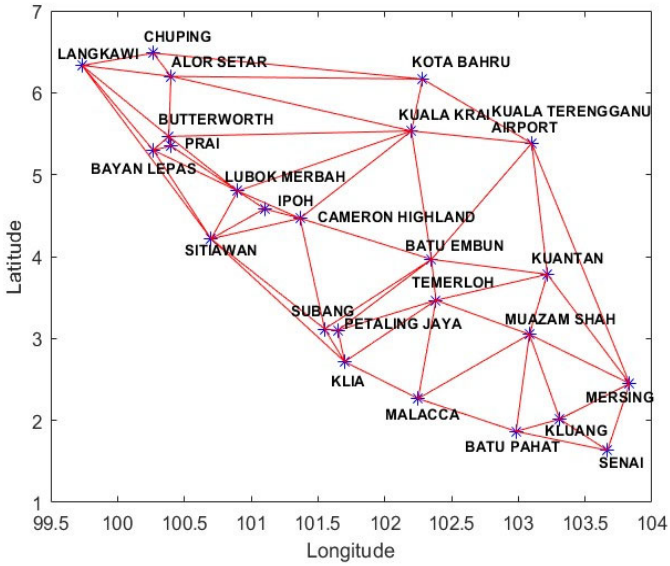

Fig. 2. Location of all 25 rainfall stations.

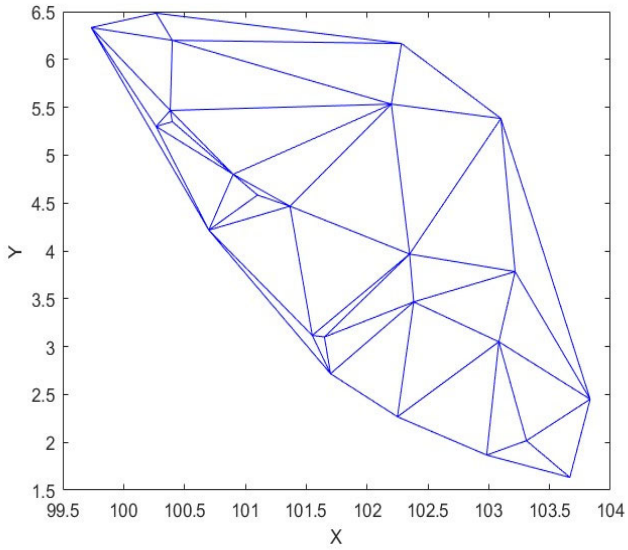

Fig. 3. Delaunay Triangulation for the irregular rainfall data sets.

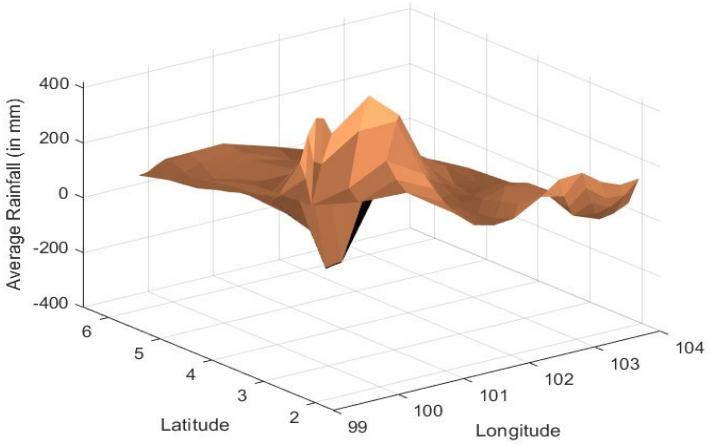

Fig. 4. Surface produced with Ball ($a = b = c = 0$).

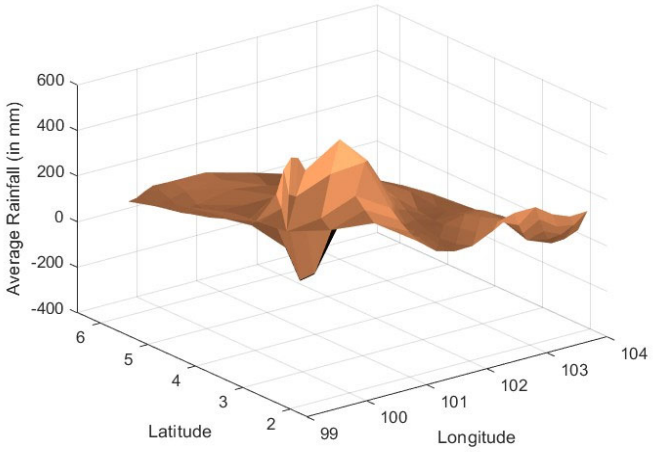

Fig. 5. Surface produced with Bézier ($a = b = c = 1$).

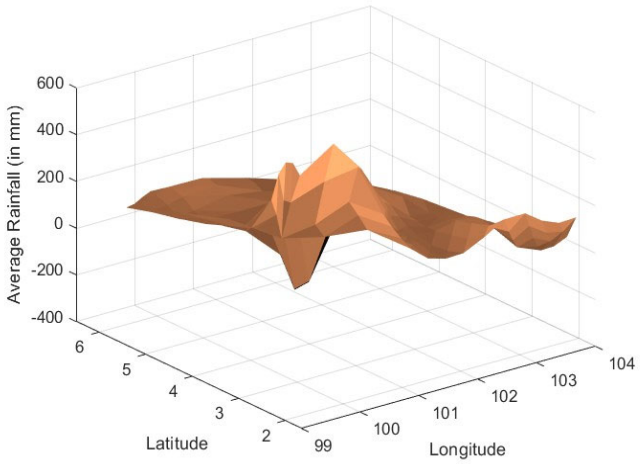

Fig. 6. Surface produced with Timmer ($a = b = c = 2$).

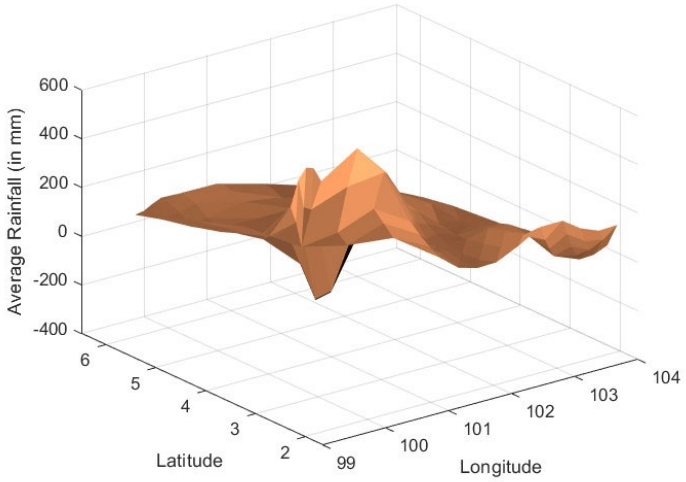

Fig.7. Surface produced with $a = b = c = 1.5$.

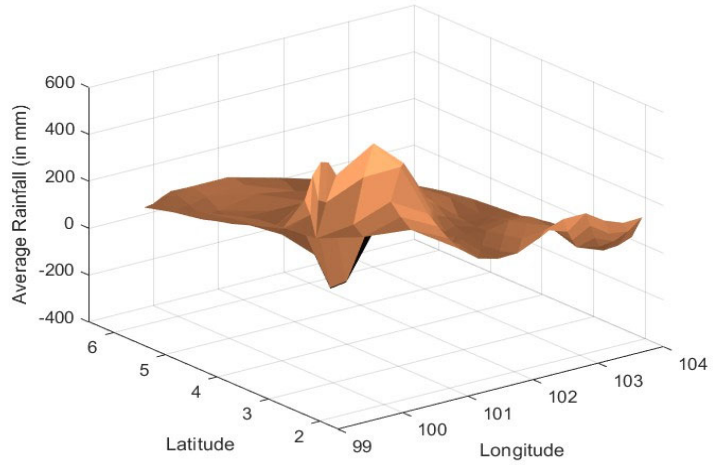

Fig. 8. Surface produced with $a = b = c = 0.5$.

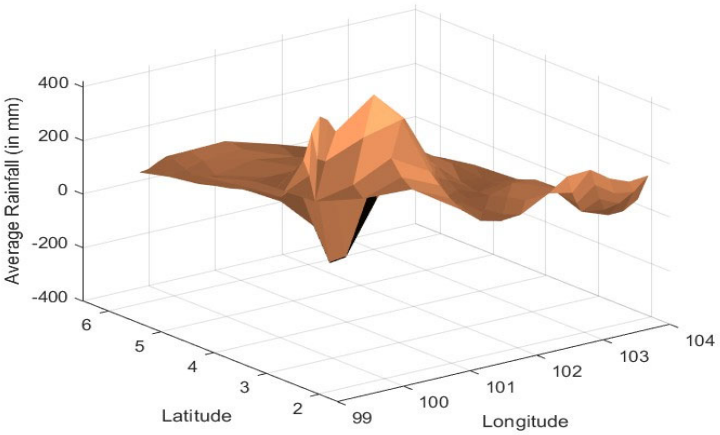

Fig. 9. Surface produced with $a = 1.5, b = c = 0$.

Figure 4 until Figure 9 shows various interpolating surfaces by using different parameter values. Clearly the scattered data interpolation by using cubic Bézier-Like triangular patches can produce a smooth surface. We compared CPU times (in second) required to generate all the interpolating surfaces shown in Figures 3-8. Table 1 summarize the results.

Table 1. CPU times (in second).

Figure	Parameter Values	CPU times (in second)
3	$a = b = c = 0$	1.118073
4	$a = b = c = 1$	1.112325
5	$a = b = c = 2$	1.152847
6	$a = b = c = 1.5$	1.179505
7	$a = b = c = 0.5$	1.223217
8	$a = 1.5, b = c = 0.$	1.147903

From Table 1, we found that, the fastest CPU time i.e., 1.112325 seconds is when $a = b = c = 1$. Meanwhile, the slowest CPU time is when $a = b = c = 0$ i.e., 1.223217 seconds. Final comparison is made against scattered data interpolation using rational quartic spline interpolant constructed in [8]. Figure 10 shows the resulting rainfall surface interpolation.

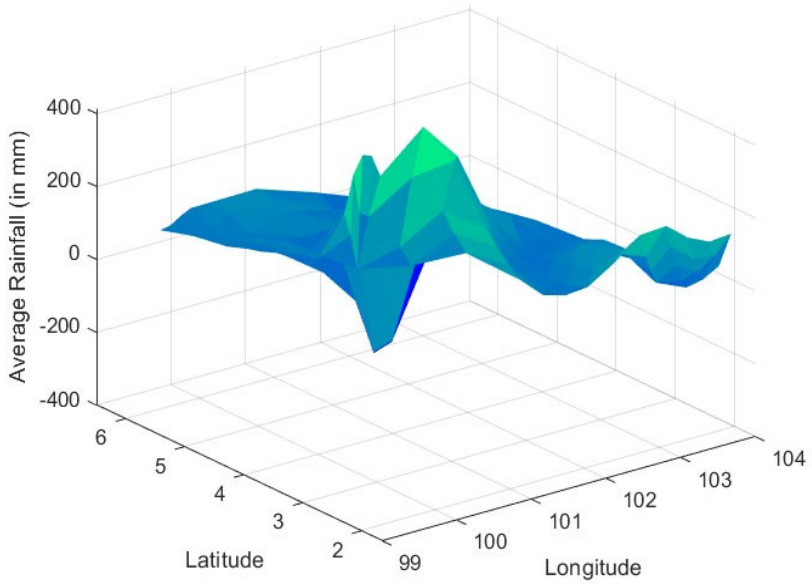

Fig. 10. Surfaced produced by [8].

In term of CPU times, scattered data interpolation scheme presented in [8] requires about 1.308483 seconds to produce interpolating surfaces shown in Figure 10. Overall, the [8] scheme require 17.63% more CPU times compared with the proposed cubic Bézier triangular patches. This is significant especially when we are dealing with big-data sets.

5 Conclusion

In this study, a scattered data interpolation scheme initiated in [4] have been used to interpolate irregular rainfall data sets. From the simulations, we found that the method is fastest when the parameter values is set to $a = b = c = 1$. These values is corresponding to the standard polynomial cubic Bézier triangular patches. One possible extension to the present study is to impose positivity-preserving condition on the ordinates. By having this positivity-preserving, we can produce a meaningful scattered data interpolation to the rainfall data sets because rainfall cannot be negative [3, 6, 12]. Another potential extension is to generalise the cubic Bézier-Like triangular bases to the quintic Bézier-Like triangular bases. By doing this, we can increase the smoothness from C^1 to C^2 parametric continuity [7]. Work on this is presently underway by the authors.

This research was fully supported by Ministry of Higher Education (MOHE) of Malaysia through Fundamental Research Grant Scheme [**FRGS/1/2023/ICT06/ UMS/02/1**] (New Scattered Data Interpolation Scheme Using Quasi Cubic Triangular Patches for RGB Image Interpolation) and Universiti Malaysia Sabah. Special thanks to the Faculty of Computing and Informatics, Universiti Malaysia Sabah for the tremendous computing facilities support.

References

1. Ali, F.A.M.; Karim, S.A.A.; Bin Saaban, A.; Hasan, M.K.; Ghaffar, A.; Nisar, K.S.; Baleanu, D. Construction of Cubic Timmer Triangular Patches and its Application in Scattered Data Interpolation. *Mathematics* ***8***, 159, doi:10.3390/math8020159, (2020).
2. Cavoretto, R.; De Rossi, A.; Dell'Accio, F.; Di Tommaso, F. Fast computation of triangular Shepard interpolants. *J. Comput. Appl. Math.*, ***354***, 457–470, doi:10.1016/j.cam.2018.03.012, (2019).
3. Karim, S.A.A.; Saaban, A.; Hasan, M.K.; Sulaiman, J.; Hashim, I. Interpolation using cubic Bézier triangular patches. *Int. J. Adv. Sci. Eng. Inf. Technol*. ***8***, 1746–1752 (2018).
4. Karim, S.A.A., Saaban, A., Skala, V. *et al.* Construction of new cubic Bézier-like triangular patches with application in scattered data interpolation. *Adv Differ Equ*, 151 (2020). https://doi.org/10.1186/s13662-020-02598-w (2020).
5. Karim, S.A.B.A.; Saaban, A. Visualization Terrain Data Using Cubic Ball Triangular Patches. *MATEC Web Conf.* ***225***, 06023, doi:10.1051/matecconf/201822506023 (2018).
6. Abdul Karim, S.A.; Saaban, A.; Nguyen, V.T. Scattered Data Interpolation Using Quartic Triangular Patch for Shape-Preserving Interpolation and Comparison with Mesh-Free Methods. *Symmetry* **2020**, *12*, 1071. https://doi.org/10.3390/sym12071071 (2020).
7. Chang, L.; Said, H. A C^2 triangular patch for the interpolation of functional scattered data. *Comput. Des. 29*, 407–412, doi:10.1016/s0010-4485(96)00068-1 (1995).
8. Draman, N.N.C.; Karim, S.A.A.; Hashim, I. Scattered Data Interpolation Using Rational Quartic Triangular Patches with Three Parameters. *IEEE Access*, ***8***, 44239–44262, doi:10.1109/access.2020.2978173 (2020).
9. Goodman, T.; Said, H.; Chang, L. Local derivative estimation for scattered data interpolation. *Appl. Math. Comput.* ***68***, 41–50, doi:10.1016/0096-3003(94)00086-j, (1995).
10. Farin, G. *Curves and Surfaces for CAGD: A Practicle Guide*, 5th ed.; Palmer, C., Ed.; Morgan Kaufmann: San Diego, CA, USA, *(*2001).
11. Lodha, S.; Franke, R. Scattered Data Techniques for Surfaces. In Proceedings of the Scientific Visualization Conference (Dagstuhl '97), Dagstuhl, Germany, 9–13 June 1997; pp. 189–230 (1997).

12. Piah, A.R.M., Goodman, T.N.T., Unsworth, K. Positivity-Preserving Scattered Data Interpolation. In: Martin, R., Bez, H., Sabin, M. (eds) Mathematics of Surfaces XI. Lecture Notes in Computer Science, vol 3604. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11537908_20, (2005).