

A Comparative Study of Convex Combination and Inner Ordinate Methods For Scattered Data Interpolation Using Quartic Triangular Patch

Owen Tamin
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Jalan UMS
Kota Kinabalu, 88400
Sabah, Malaysia
owentamin1996@gmail.com

Samsul Ariffin Abdul Karim
Data Technologies and Applications (DaTA) Research Lab
Faculty of Computing and Informatics
Jalan UMS
Kota Kinabalu, 88400
Sabah, Malaysia
samsulariffin.karim@ums.edu.my

Mohammad Khatim Hasan
Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
Bangi, 43600 UKM
Bangi, Selangor, Malaysia
mkh@ukm.edu.my

ABSTRACT

In this study, we perform a comparative evaluation and assessment for the scattered data interpolation using a quartic polynomial triangular patch with ten control points on a triangular domain. The comparison is made using two different convex combinations and inner ordinates methods, i.e., cross derivative and cubic precision. Statistical Goodness-fit measurements used are maximum error, coefficient of determination (R^2), CPU time (in seconds), and contour plot. From the result, the cubic precision method with linear convex combination methods gave better results with smaller CPU times and higher R^2 value. All numerical and graphical results are presented using MATLAB programming.

Keywords

Quartic polynomial, convex combination, inner ordinates, scattered data, interpolation, triangular, patch

1 INTRODUCTION

Scattered data interpolation is a pivotal technique in modern computational science, offering a means to estimate values at arbitrary locations within a dataset characterized by irregularly distributed data points. Unlike structured datasets arranged on grids or meshes, scattered data often emerges from real-world observations or simulations, where data points are irregularly positioned in multidimensional space. Interpolating scattered data entails the construction of a continuous function that approximates the underlying behaviour of the dataset, enabling the inference of values at unobserved locations within the dataset's domain [Fran91].

There are two common categories for scattered data interpolation. The first type involves triangulating the

data sets, while the second type, mesh-free interpolation, does not require the triangulation. Standard basis functions utilized in interpolation include Bézier, B-spline, and radial basis functions (RBFs). Previous studies [Cav19, Dell20, Dell18, Dell16, Ska23] have explored the application of the Shepard triangular method, a part of the meshless approach for surface reconstruction. However, these methods often demand significant computational time to generate interpolated surfaces. Alternatively, Bézier or spline triangular methods can construct piecewise smooth surfaces with desired degrees of smoothness (e.g., C^1 or C^2) while requiring less computational time, provided that certain continuity conditions are met at the adjacent boundary.

Goodman and Said [Good91] developed a C^1 triangular interpolant appropriate for interpolating scattered data. They achieved this by employing a convex combination scheme that combines three local schemes. Their work differs from that of Foley and Opitz [Foley92]. Furthermore, both works established a C^1 cubic triangular convex combination scheme. However, limited research has been conducted to compare the relative

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

performance of these two methods. Foley and Opitz [Foley92] asserted that employing the cubic precision method resulted in enhanced visual smoothness of the surface for small scattered data sets, as well as a smaller root mean square (RMS) error compared to the method proposed by Goodman and Said [Good91]. Furthermore, two versions of convex combinations exist, differing only in the degree of the rational function. However, less research is conducted on determining which version is better when dealing with large scattered data sets.

Several studies have investigated the practical application of cubic Bézier curves in handling real-life data, primarily due to their simpler complexity and lower computational demands. For instance, Karim et al. [Karim18a] delved into spatial interpolation methods for rainfall data by employing cubic Bézier triangular patches to interpolate scattered data points. They also introduced a novel cubic Bézier-like triangular patches designed explicitly for interpolating scattered data [Karim20a]. Additionally, Karim and Saaban [Karim18b] utilized cubic Ball triangular patches to generate terrain data. However, the inherent simplicity of cubic Bézier patches is suitable only for achieving moderate smoothness in large datasets. A higher-degree polynomial is necessary to achieve a smoother surface. Quartic Bézier triangular patches have received less attention from researchers in the past due to their requirement of 15 control points, which need optimization techniques to ensure C^1 continuity is met on each adjacent triangles. This optimization process consumes more computational time, and the quartic scheme is global and not local [Piah06, Aziz05, Huss14a, Huss14b].

A recent study by Karim et al. [Karim20b] has successfully developed a C^1 scattered data interpolation scheme without requiring any optimization, building upon the extension work by Zhu and Han [Zhu13]. This newly proposed scheme is characterized as local, and the central processing unit (CPU) time needed for constructing the surface is notably faster compared to the quartic Bézier triangular patches in previous studies [Piah06, Aziz05, Huss14a, Huss14b]. Furthermore, the proposed scheme ensures positivity preservation in scattered data interpolation, leading to improved interpolated surfaces on the real-life data based on coronavirus disease 2019 (COVID-19) cases at Selangor State and Klang Valley in Malaysia. This advancement overcomes the limitations of previous works by [Ali20, Dra20, Karim20a], which did not apply positivity preservation in their interpolations. However, it is worth noting that this novel scheme has not yet been tested on other real-life datasets, such as geological data.

Motivated by these developments, this study aims to compare two version of convex combination and methods for forming local schemes based on the inner ordinates, focusing on the quartic Bézier surface interpolation scheme developed by Karim et al. [Karim20b]. The analysis will include error assessment using metrics such as R^2 , maximum error, and CPU time. Additionally, graphical representations showcasing the surfaces and their corresponding contour plots will be presented. Finally, the proposed scheme will be applied to construct real-life geologic data, specifically Seamount data, to demonstrate its applicability in practical scenarios.

This paper is organized into five main sections. The first section introduces the research topic and outlines its objectives. The second section provides a review of related literature and methodologies from related studies. The third section elaborates on the methods utilized in the research. Subsequently, the fourth section presents the findings and engages in a comprehensive discussion. Finally, the fifth section concludes the paper and suggests areas for future work.

2 RELATED WORK

This section discusses the related work on the methods for forming local schemes for inner ordinates and convex combinations.

Goodman and Said's method

This method, also known as the cross derivative method, achieves a C^1 surface by prescribing first-order derivatives at the vertices. This method involves blending three cubic polynomials to create a rational function interpolant within each triangle of the domain. One of the significant advantages of this method is its locality; the surface at any point depends only on the data "close" to that point. This local nature makes the method efficient and well-suited for surface interpolation applications. The method begins by triangulating the domain using the data points as vertices. Within each triangle, a local interpolant is defined. To ensure a smooth surface, specific derivative values are typically employed on the boundaries of the triangles. The inner Bézier ordinates, denoted as b_{111} , are determined by the local scheme to satisfy the C^1 requirement along all edges ε_i , $i = 1, 2, 3$ of the triangular respectively as shown in Fig. 1.

Foley and Opitz's method

This method is also known as the cubic precision method. This method is best represented in Fig. 2 in which the two adjacent triangles with the vertex of V_i and \tilde{V}_i where $i = 1, 2, 3$. The hybrid patch on the left triangle with boundary control point b_{ijk} is identical to the right triangle with boundary control point b_{ijk} ,

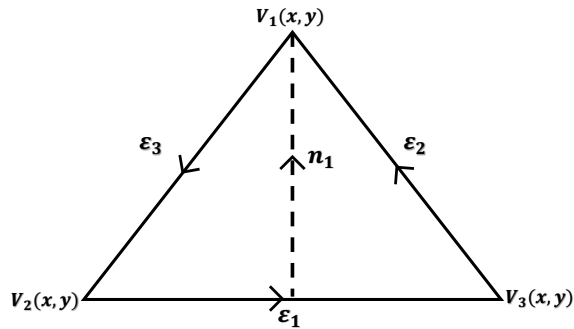


Figure 1: Triangle with side-vertex blending

where $i + j + k = 3$. The C^1 continuity is achieved by making sure the sets of four control points along the edge from both sides of the triangle are co-planar. Complete derivations can be found in Karim et al. [Karim20b].

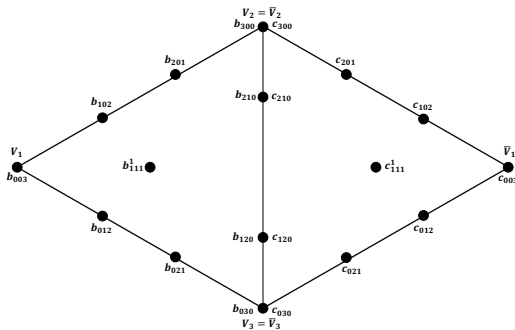


Figure 2: Two adjacent hybrid cubic patches

Convex Combination

In the literature, there are two common convex combination schemes that can be used in scattered data interpolation for calculating the inner ordinates $b_{111}^i, i = 1, 2, 3$ where the barycentric coordinate (u, v, w) on the triangle is defined by $u + v + w = 1$, where $u, v, w \geq 0$. Eqn. (1) represents a linear convex combination, whereas Eqn. (2) represents a square linear convex combination which can be shown as below:

$$b_{111} = \rho_1 b_{111}^1 + \rho_2 b_{111}^2 + \rho_3 b_{111}^3$$

where

$$\rho_1 = \frac{vw}{uv + vw + wu}, \quad \rho_2 = \frac{wu}{uv + vw + wu}, \quad (1)$$

$$\rho_3 = \frac{uv}{uv + vw + wu}$$

or

$$\rho_1 = \frac{v^2 w^2}{u^2 v^2 + v^2 w^2 + w^2 u^2}, \quad \rho_2 = \frac{w^2 u^2}{u^2 v^2 + v^2 w^2 + w^2 u^2},$$

$$\rho_3 = \frac{u^2 v^2}{u^2 v^2 + v^2 w^2 + w^2 u^2} \quad (2)$$

3 METHOD

This section describes the construction of quartic triangular patches on a triangular domain.

3.1 Quartic Triangular Patches

Since we are dealing with triangulation, the barycentric coordinates (u, v, w) on the triangle T with vertices V_1, V_2 and V_3 is defined by $u + v + w = 1$, where $u, v, w \geq 0$. Set the point inside the triangle as $V(u, v, w) \in R^2$ (as shown in Fig. 3), which can be expressed as:

$$V = uV_1 + vV_2 + wV_3$$

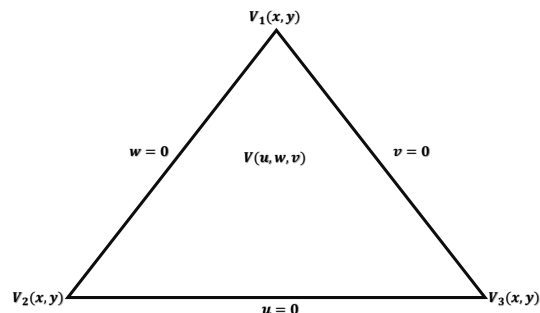


Figure 3: Triangle T

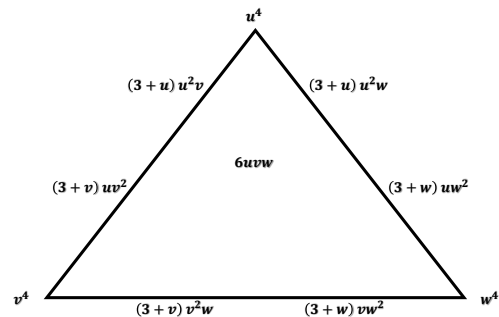


Figure 4: Basis functions for quartic triangular patch

Fig. 4 shows the basis functions for quartic triangular patch. The basis functions satisfy the properties of non-negativity, partition of unity and symmetry (full details of the properties can be referred to Zhu and Han [Zhu13]). Thus, the quartic triangular patch from Zhu and Han is further defined by

$$R(u, v, w) = u^4 b_{300} + v^4 b_{030} + w^4 b_{003} + u^2 v (3 + u) b_{210} +$$

$$(3 + u) u^2 w b_{201} + (3 + v) v^2 u b_{120} +$$

$$(3 + v) v^2 w b_{021} + (3 + w) w^2 u b_{102} +$$

$$(3 + w) w^2 v b_{012} + 6 u v w b_{111} \quad (3)$$

3.2 Scattered Data Interpolation Using Quartic Triangular Patches

In this section, the following scattered data interpolation scheme is constructed based on Karim et al. [Karim20b].

3.2.1 Boundary Ordinates

The boundary ordinates are calculated based on Goodman and Said [Good91] method for each triangle, as represented in Fig. 5.

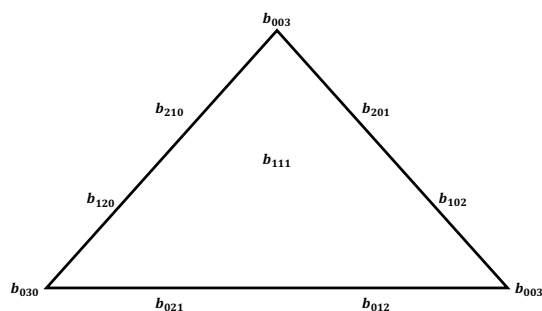


Figure 5: Control points for quartic triangular patch

Based on the literature review from Sec. 2, the vertex is given as $F(V_1) = b_{300}$, $F(V_2) = b_{030}$, and $F(V_3) = b_{003}$. The other six boundary ordinates are expressed as:

$$\begin{aligned} b_{210} &= b_{300} + \frac{1}{4} ((x_2 - x_1) F_x(V_1) + (y_2 - y_1) F_y(V_1)) \\ b_{201} &= b_{300} - \frac{1}{4} ((x_1 - x_3) F_x(V_1) + (y_1 - y_3) F_y(V_1)) \\ b_{021} &= b_{030} + \frac{1}{4} ((x_3 - x_2) F_x(V_2) + (y_3 - y_2) F_y(V_2)) \\ b_{120} &= b_{030} - \frac{1}{4} ((x_2 - x_1) F_x(V_2) + (y_2 - y_1) F_y(V_2)) \\ b_{102} &= b_{003} + \frac{1}{4} ((x_1 - x_3) F_x(V_3) + (y_1 - y_3) F_y(V_3)) \\ b_{012} &= b_{003} - \frac{1}{4} ((x_3 - x_2) F_x(V_3) + (y_3 - y_2) F_y(V_3)) \end{aligned}$$

3.2.2 Inner Ordinates

The remaining inner ordinates, $b_{111}^i, i = 1, 2, 3$, are obtained by first using the cross derivative method by Goodman and Said [Good91]. Then, after the inner ordinates are calculated, the second inner ordinates are then calculated using cubic precision by Foley and Opitz [Foley92].

3.2.3 Final Scheme

The final interpolating scheme can be written as

$$\begin{aligned} R(u, v, w) &= \sum_{\substack{i+j+k=3 \\ i, j, k \neq 1}} b_{ijk} B_{i,j,k}^3(u, v, w) \\ &+ 6uvw (\rho_1 b_{111}^1 + \rho_2 b_{111}^2 \\ &+ \rho_3 b_{111}^3) \end{aligned} \quad (4)$$

where $B_{i,j,k}^3(u, v, w)$ represents the Bernstein polynomial of degree 3, and ρ_1, ρ_2 , and ρ_3 are determined by Eqn. (1) or (2).

3.3 Algorithm

This section shows the overall algorithm used in this study, as represented in Algorithm 1.

Algorithm 1 Quartic Triangular Patches for Scattered Data Interpolation

- 1: Input: scattered data points
- 2: Estimate the partial derivative at the data points by using [Karim20a];
- 3: Triangulate the data points by using Delaunay triangulation;
- 4: Calculate the boundary control points;
- 5: Calculate inner control points for the local scheme, $b_{111}^i, i = 1, 2, 3$ by using the cubic precision method as in Foley and Opitz [Foley92];
- 6: Construct the interpolated surface using the convex combination method of three local schemes defined by Eqn. (4)

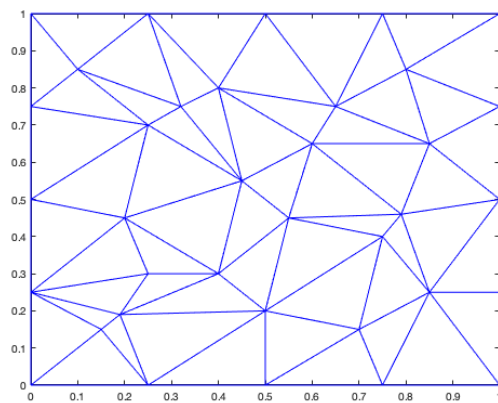
4 RESULTS AND DISCUSSION

In this section, we will compare the performance between the different convex combination and methods for forming the local scheme. The proposed scheme is tested with one well-known function, Franke's exponential function:

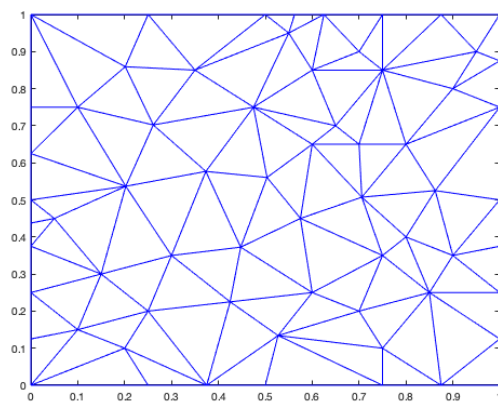
$$\begin{aligned} F_1(x, y) &= 0.75e^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + 0.75e^{-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}} \\ &+ 0.5e^{-\frac{(9x-7)^2+(9y-3)^2}{4}} - 0.2e^{-(9x-4)^2 - (9y-7)^2} \end{aligned}$$

The error norms are computed using a 33×33 uniform rectangular grid of evaluation points in the unit square for 36, 65 and 100 data points. Fig. 6 shows the Delaunay triangulation for sample of 36, 65 and 100 data points. The error measurements used are (a) Coefficient of determination, R^2 , (b) maximum errors and (c) Central Processing Unit (CPU) time in seconds. The simulation is conducted using MATLAB 2023a version.

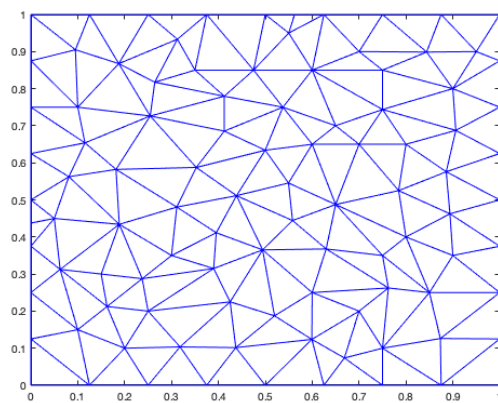
Fig. 7 shows an example of scattered data interpolation using quartic triangular patches based on $F_1(x, y)$ function based on 65 data points. The scattered data interpolation is performed within each mesh triangle, which is then evaluated at points on a 33×33 uniform rectangular grid within the unit square domain. This evaluation involves applying the interpolant function to each grid point to estimate the function value based on the test function F_1 . The contour plot of the respective surface interpolation is shown in Fig. 8.



(a)



(b)



(c)

Figure 6: Delaunay triangulation for (a) 36 data points; (b) 65 data points and (c) 100 data points

Based on Fig. 7, there are no significant differences between the surface of the F_1 function and the four interpolated surfaces. However, when it comes to the contour plot, which can be referred to in Fig. 8, there are no exact similarities between the four contour plots for the interpolated surface. These differences indicate that different methods with different convex combinations give different appearances, and it's worth further investigating which one of them is actually the best. A better comparison between the methods is represented with error analysis, which is shown in Table 1.

Based on Table 1, the cubic precision method with linear convex combination shows the best performance in terms of higher R^2 value and smaller CPU time (in seconds) as compared to the three methods. Interestingly, the linear convex combination also favors the cross derivative method, where its performance is better than the one with the square convex combination. This observation tells us that selecting the convex combination depends on which method is acquired for scattered data interpolation. This observation will help researchers to be able to select the best choice for selecting which convex combination to choose when it comes to different methods for calculating the inner ordinates.

Another interesting observation is that the performance of both proposed methods significantly improved as the number of data points increased from 33 to 100, as evidenced by higher R^2 values, reduced maximum errors, and more efficient CPU utilization. This improvement can be attributed to the increase in triangulation density from coarse to fine, enabling the capture of subtle variations and features of the underlying surface of the test function F_1 . This pattern is also reflected in the contour plot shown in Fig. 8, where visual smoothness is observed to increase with the higher density of data points. Finer triangulation are adept at accurately representing intricate details of the surface but may lead to increased computational complexity and runtime. On the other hand, coarser triangulation simplify the surface representation but can potentially result in a loss of detail and accuracy.

Nevertheless, the cubic precision method with linear convex combination still performs the best based on the four-method comparison. By having this result, the best performing method will be used to reconstruct real-life geologic data, which is the Seamount data. Seamount data sets obtained from MATLAB represent the surface of the Mount under the sea located on Louisville Ridge in the South Pacific in 1984. Fig. 9 shows the Delaunay triangulation of Seamount with 294 data points as well as the 3D interpolation of those data points. Fig. 10 shows the surface of the Seamount data points as well as the interpolated surface using cross derivative and cubic precision with linear combination. The interpolated surface consists of 566 triangles and is evaluated

by subdivision of the triangles. Table 2 shows the CPU time comparison between the two methods. The CPU time is taken by averaging three runs of the experiment to prevent bias. Based on the table, the cubic precision with linear convex combination takes a lower time to reconstruct the Seamount data points at 3.3901 seconds faster than cross derivative with the linear combination. This observation shows that the cubic precision method developed by Foley and Opitz [Foley92] is more accurate and faster than the cross derivative by Goodman and Said [Good91].

A claim by Foley and Opitz [Foley92] mentioning that their method works better than cross derivative is valid in this case even though the differences are not significant. These small differences can lead to big changes in performance when it comes to interpolating more big data points. Furthermore, the CPU time required to reconstruct seamount data by using a cubic Timmer triangular patch developed by Ali et al. [Ali20] scheme is 123.9761 seconds. Meanwhile, the proposed quartic triangular patch has achieved an average CPU time of 83.4646 (cross derivative) and 80.0745 (cubic precision) for linear convex combination. Based on this result, the proposed scattered data interpolation scheme by using quartic polynomial is at least 1.5 time faster than scattered data interpolation scheme based on cubic triangular Timmer developed by Ali et al. [Ali20]. This result is significant because the proposed scheme has quartic degree, meanwhile the existing schemes are cubic degree.

5 CONCLUSION

In conclusion, this study conducted a comparative evaluation of scattered data interpolation using quartic polynomial triangular patches with ten control points on a triangular domain. The comparison focused on two different methods: convex combination with inner ordinates using cross derivatives and cubic precision. The results of our evaluation indicate that the cubic precision method, coupled with linear convex combination schemes, outperformed the cross derivative method. Specifically, the cubic precision method exhibited shorter CPU times and higher R^2 values, demonstrating its effectiveness in interpolating scattered data using quartic polynomial triangular patches. Furthermore, our study highlights the efficacy of the linear convex combination scheme in conjunction with the cubic precision method for scattered data interpolation. This combination yielded superior results, particularly in reconstructing geologic real-life data, such as Seamount, and outperformed the linear convex combination scheme with the cross derivative method in terms of computational efficiency. Future studies will focus on implementing these successful schemes on GPU platforms using machine learning techniques.

This strategic approach aims to enhance computation time for surface reconstruction while simultaneously improving accuracy. Overcoming challenges in GPU implementation and leveraging machine learning algorithms will contribute significantly to advancing the field of scattered data interpolation.

6 ACKNOWLEDGMENTS

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.

7 REFERENCES

- [Fran91] Franke, R., & Nielson, G. M. (1991). Scattered data interpolation and applications: A tutorial and survey. *Geometric Modeling: Methods and Applications*, 131-160.
- [Cav19] Cavoretto, R., De Rossi, A., Dell'Accio, F., Di Tommaso, F. (2019). Fast computation of triangular Shepard interpolants. *Journal of Computational and Applied Mathematics*, 354, 457-470.
- [Dell20] Dell'Accio, F., Di Tommaso, F. (2020). On the hexagonal Shepard method. *Applied Numerical Mathematics*, 150, 51-64.
- [Dell18] Dell'Accio, F., Di Tommaso, F., Nouisser, O., Zerroudi, B. (2018). Increasing the approximation order of the triangular Shepard method. *Applied Numerical Mathematics*, 126, 78-91.
- [Dell16] Dell'Accio, F., Di Tommaso, F., Hormann, K. (2016). On the approximation order of triangular Shepard interpolation. *IMA Journal of Numerical Analysis*, 36(1), 359-379.
- [Good91] Goodman, T. N. T., Said, H. B. (1991). A C^1 triangular interpolant suitable for scattered data interpolation. *Communications in Applied Numerical Methods*, 7(6), 479-485.
- [Foley92] Foley, T. A., Opitz, K. (1992). Hybrid cubic Bézier triangle patches. In *Mathematical methods in computer aided geometric design II* (pp. 275-286). Academic Press.
- [Karim18a] Karim, S. A., Saaban, A., Hasan, M. K., Sulaiman, J., Hashim, I. (2018). Interpolation using cubic Bézier triangular patches. *Int. J. Adv. Sci. Eng. Inf. Technol*, 8(4-2), 1746-1752.
- [Karim20a] Karim, S. A. A., Saaban, A., Skala, V., Ghaffar, A., Nisar, K. S., Baleanu, D. (2020).

Convex combination	Data Points (F_1)	R^2		Max Error		CPU Time (s)	
		CD ¹	CP ²	CD ¹	CP ²	CD ¹	CP ²
Linear	36	0.9917	0.9918	0.1146	0.1110	0.2948	0.2910
	65	0.9960	0.9962	0.0789	0.0786	0.3238	0.3382
	100	0.9990	0.9991	0.0632	0.0643	0.4246	0.3752
Square	36	0.9916	0.9916	0.1151	0.1111	0.4644	0.3492
	65	0.9960	0.9961	0.0785	0.0787	0.4828	0.3331
	100	0.9989	0.9990	0.0629	0.0655	0.5160	0.4962

¹ Cross Derivative

² Cubic Precision

Table 1: Overall error measurements for 36, 65 and 100 data points for Quartic Bézier Triangular Surface with different method and convex combination

Method	Convex combination	CPU Time (s)			
		First run	Second run	Third run	Average
Cross Derivative	Linear	81.1796	89.1468	80.0674	83.4646
Cubic Precision	Linear	79.9700	80.2119	80.0416	80.0745

Table 2: CPU time comparison between cross derivative and cubic precision method using linear convex combination

Construction of new cubic Bézier-like triangular patches with application in scattered data interpolation. *Advances in Difference Equations*, 2020(1), 151.

[Karim18b] Karim, S. A. B. A., Saaban, A. (2018). Visualization terrain data using cubic Ball triangular patches. In *MATEC Web of Conferences* (Vol. 225, p. 06023). EDP Sciences.

[Piah06] Mt Piah, A. R., Saaban, A., Abd Majid, A. (2006). Range restricted positivity-preserving scattered data interpolation. *Journal of Fundamental Sciences*, 2(1-2), 63-75.

[Aziz05] Azizan, S., Piah, A. R. M., Ahmad, A. M. (2005). G^1 scattered data interpolation with minimized sum of squares of principal curvatures.

[Huss14a] Hussain, M., Abd Majid, A., Hussain, M. Z. (2014). Convexity-preserving Bernstein-Bézier quartic scheme. *Egyptian Informatics Journal*, 15(2), 89-95.

[Huss14b] Hussain, M., Hussain, M. Z., Buttar, M. (2014). C^1 positive Bernstein-Bézier rational quartic interpolation triangle, 1(2), 3.

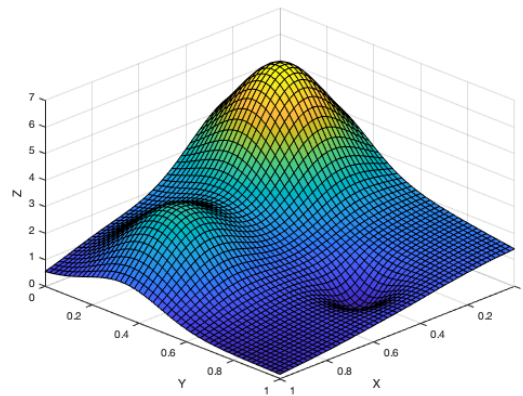
[Karim20b] Karim, S. A. A., Saaban, A., Nguyen, V. T. (2020). Scattered data interpolation using quartic triangular patch for shape-preserving interpolation and comparison with mesh-free methods. *Symmetry*, 12(7), 1071.

[Zhu13] Zhu, Y., Han, X. (2013). A class of $\alpha\beta\gamma$ -Bernstein-Bézier basis functions over triangular domain. *Applied Mathematics and Computation*, 220, 446-454.

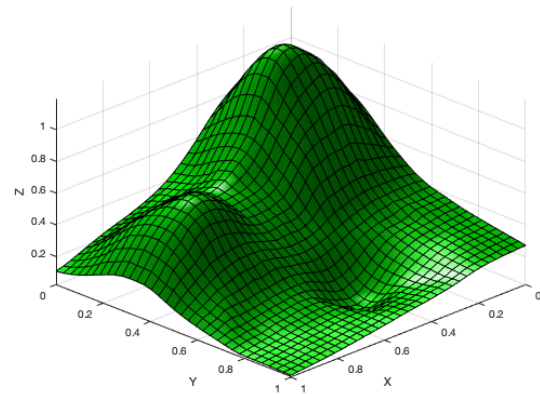
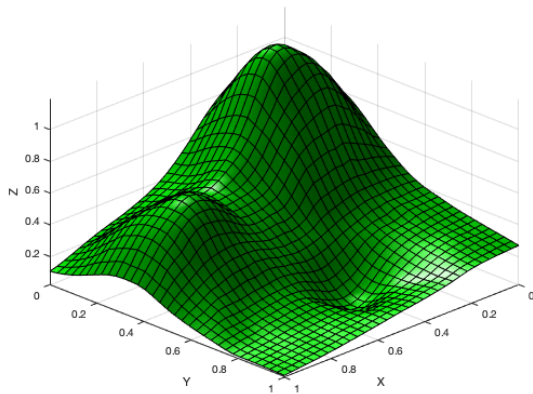
[Ali20] Ali, F. A. M., Abdul Karim, S. A., Saaban, A., Hasan, M. K., Ghaffar, A., Nisar, K. S., Baleanu, D. (2020). Construction of cubic Timmer triangular patches and its application in scattered data interpolation. *Mathematics*, 8(2), 159.

[Dra20] Draman, N. N. C., Karim, S. A. A., Hashim, I. (2020). Scattered data interpolation using rational quartic triangular patches with three parameters. *IEEE Access*, 8, 44239-44262.

[Ska23] Skala, V.; Mourycova, E. Meshfree Interpolation of Multidimensional Time-Varying Scattered Data. *Computers* 2023, 12, 243. <https://doi.org/10.3390/computers12120243>

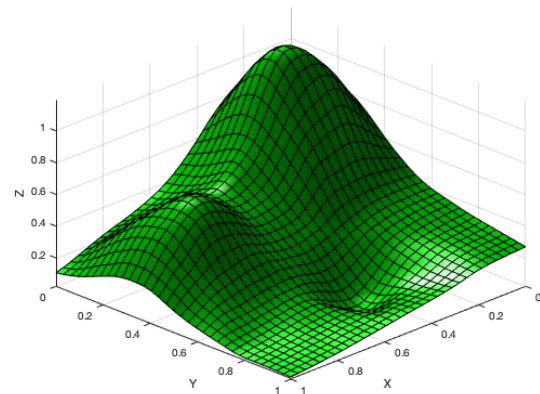
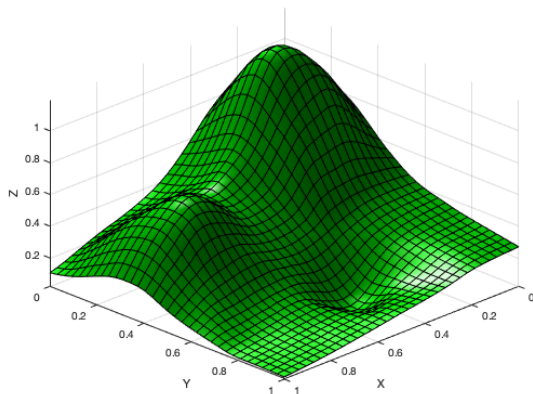


(a) F_1 test function



(b) Cross derivative method with linear convex combination

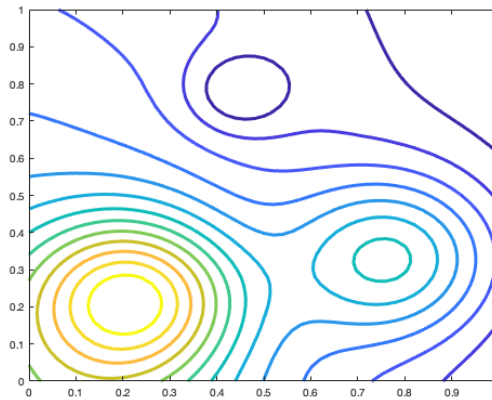
(c) Cross derivative method with square convex combination



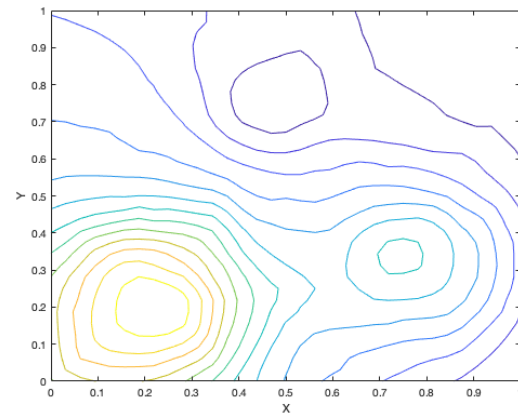
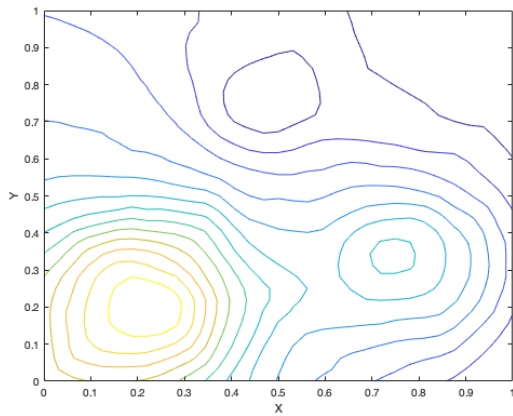
(d) Cubic precision method with linear convex combination

(e) Cubic precision method with square convex combination

Figure 7: Example of surface interpolation based on F_1 test function

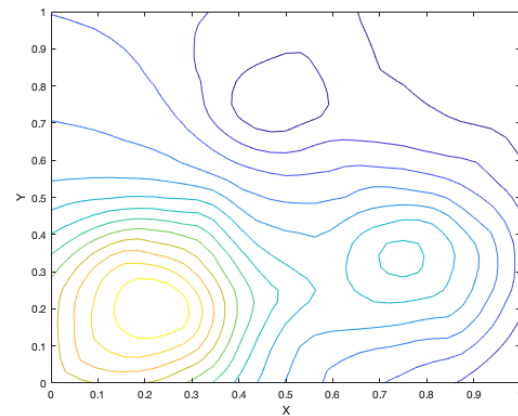
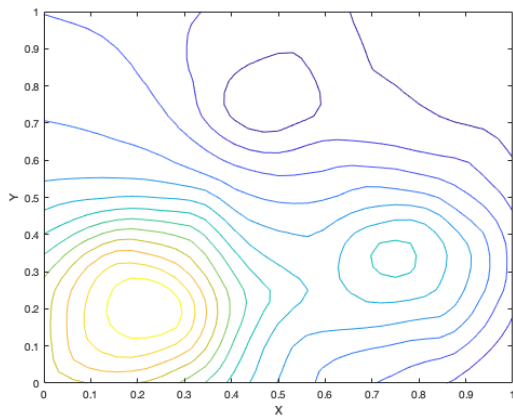


(a) F_1 test function



(b) Cross derivative method with linear convex combination

(c) Cross derivative method with square convex combination



(d) Cubic precision method with linear convex combination

(e) Cubic precision method with square convex combination

Figure 8: Contour plots of surfaces presented in Fig. 5

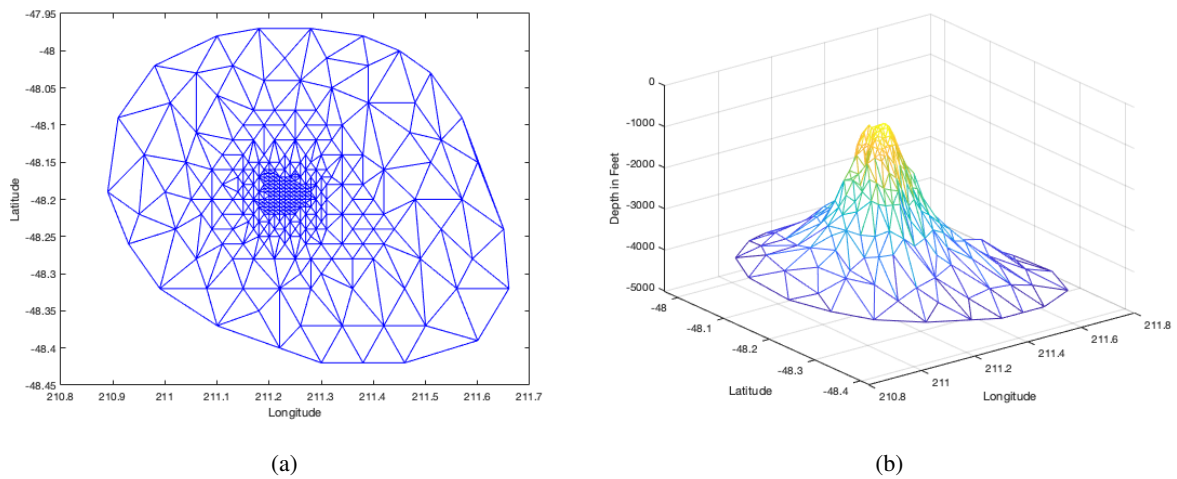
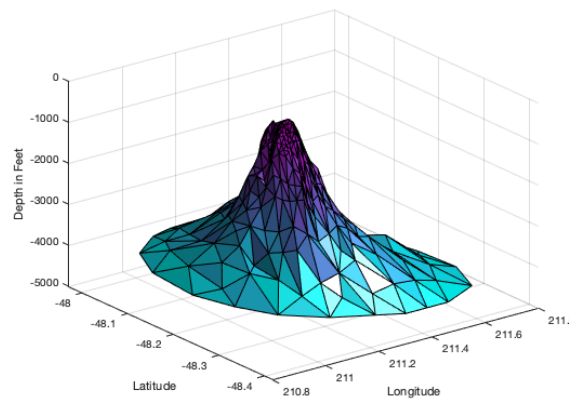
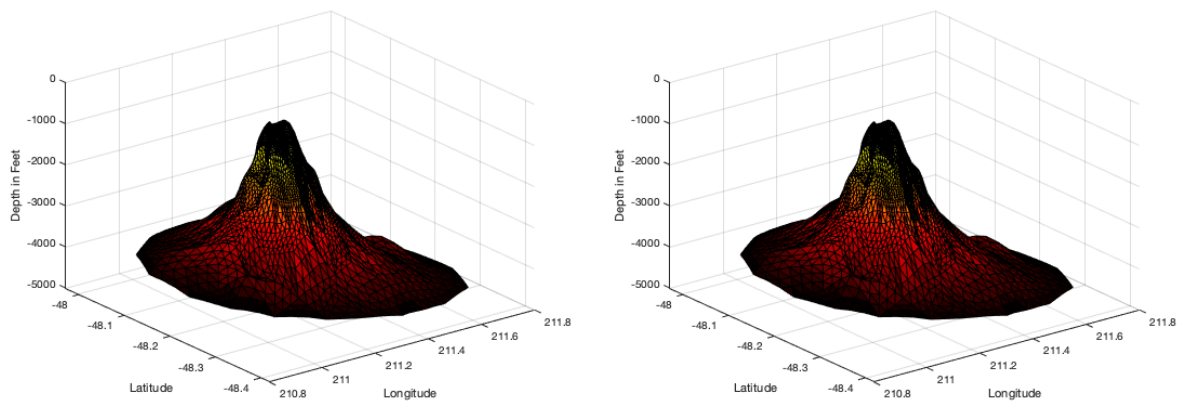


Figure 9: Example of (a) Delaunay Triangulation of Seamount with 294 data points and (b) 3D interpolation of Seamount data points



(a) Seamount surface generated in Matlab



(b) Cross derivative method with linear convex combination (c) Cubic precision method with linear convex combination

Figure 10: Example of surface generated based on Seamount real data points