Journal of Computing Research and Innovation
e-ISSN : 2600-8793

Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 10(2) 2025

Journal of Computing Research and Innovation

Exploring the Effect of Shape Parameters on Font Design Using Rational Quadratic Trigonometric Curves

Noor Khairiah Razali[1*], Nur Ain Fitrah Azhari[2], Siti Musliha Nor-Al-Din[3], Nursyazni Mohd Sukri[4]

[1,3,4]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Terengganu Branch, Kuala Terengganu Campus, Malaysia.*
[2]*Shell Malaysia Limited, Malaysia.*

ARTICLE INFO

Article history:
Received 17 June 2025
Revised 25 August 2025
Accepted 26 August 2025
Online first
Published 1 September 2025

Keywords:
Rational Quadratic Trigonometric Bézier
Shape Parameter
Font design
Interactive tool

DOI:
10.24191/jcrinn.v10i2.541

ABSTRACT

This study investigates the impact of shape parameters on font contour design using rational quadratic trigonometric Bézier (RQTB) curves. By varying parameters (m, n) and weight (v), the research evaluates contour accuracy through pointwise deviation analysis and computational efficiency. The configuration ($m = 0.5$, $n = -0.5$, $v = 1$) achieved the lowest RMSE and CPU time, indicating optimal performance. An interactive tool was also developed to support real-time parameter adjustment. The results demonstrate that carefully tuned shape parameters enhance both visual fidelity and efficiency, making RQTB curves practical for typography and related design applications.

1. INTRODUCTION

Curve modeling is a core aspect of computer-aided geometric design (CAGD), especially in typography where both visual appeal and precision are required. Rational Bézier curves are widely used because they allow smooth transitions and flexible control through the use of weights (Ceylan et al., 2021; Salomon, 2006). However, one of their main limitations is the lack of local shape control, which makes it difficult to fine-tune detailed parts of designs such as font contours. To address this, rational quadratic trigonometric Bézier (RQTB) curves were introduced. These curves use trigonometric basis functions along with shape parameters, giving designers the ability to adjust curvature directly without having to modify control points (Bashir et al., 2012). This makes them particularly useful in type design, where small changes in curve behavior can have a big impact on readability and style.Previous studies have shown how RQTB curves can be applied in practice. Ahmad and Gobithaasan (2018) introduced RQT spirals that maintain curvature

[1*] Corresponding author. *E-mail address*: noorkhairiah@uitm.edu.my
https://doi.org/10.24191/jcrinn.v10i2.541

continuity for font strokes, while Dube and Gupta (2022) explored how shape parameters affect the appearance of letters. Ahmed et al. (2022) applied Bézier-like curves in Arabic typography, showing that the method is adaptable across different scripts. From a geometric perspective, Sánchez-Reyes (2022) proposed tools for local shape control using shape factors, and Omar et al. (2023) extended the framework with two shape parameters for even greater flexibility. Beyond typography, Munir et al. (2024) applied similar approaches to natural shape modeling, demonstrating the wider relevance of these methods.

Despite these advances, curve modeling still faces important limitations. A common approach to achieving complex shapes is to increase the number of control points. While this can improve approximation, it often results in higher computational cost, more complicated editing, and sometimes unwanted distortions in the outline. In typography, where balance and readability are critical, too many control points can actually reduce the quality of the design. Shape parameters offer a more efficient alternative by allowing designers to adjust curvature directly, without relying on additional control points. This approach not only simplifies the modeling process but also improves accuracy. However, the effectiveness of shape parameters depends on finding the best parameter settings, which has not been studied in a systematic way.

This paper addresses this gap by evaluating how shape parameters (m, n) and weight (v) influence font contour quality and CPU execution time using RQTB curves. A stylized font character was modelled using multiple curve segments, and the accuracy of each configuration was assessed through pointwise deviation analysis, which measures the Euclidean distance between the generated and reference contours. This quantitative approach provides a robust metric for evaluating contour fidelity. Additionally, an interactive tool was developed to allow real-time parameter adjustment, offering immediate feedback on both visual and computational outcomes. The findings demonstrate that carefully tuned shape parameters can significantly enhance both the accuracy and performance of font design, with broader implications for CAD and computer graphics applications.

2. METHODOLOGY

This study applies rational quadratic trigonometric Bézier (RQTB) functions to model font contours, focusing on the influence of global shape parameters on curve geometry. The methodology involves three main stages: constructing curves using predefined control points, adjusting the shape parameters m, n, v and refining the contours through iterative evaluation. Each configuration is assessed based on curvature smoothness, computational efficiency, and visual quality, with contour accuracy measured using pointwise deviation analysis to quantify the alignment between generated curves and the reference font outline.

2.1 Curve formulation

In geometric modeling, rational trigonometric curves offer a powerful framework for representing smooth and flexible shapes, particularly useful in applications like font design where stroke aesthetics are critical. The RQTB is foundational formulations that enable fine control over curve shape through the incorporation of global shape parameters. These basis functions utilize sine and cosine terms to construct curves that can closely mimic circular arcs while also offering designer-defined control over curvature transitions. The RQTB approach is computationally simpler and suitable for applications requiring moderate flexibility.

In this section, the basis sets are defined, and the respective curve formulations are constructed using a set of control points and shape parameters.

https://doi.org/10.24191/jcrinn.v10i2.541

2.1.1 Definition1: Rational Quadratic Trigonometric Bézier (RQTB) basis

According to Bashir et al. (2012), quadratic trigonometric basis functions with two shape parameters, which are m and n, where $m, n \in [-1,1]$ are defined as:

$$\begin{aligned} f_0(u) &= \left(1-\sin\frac{\pi}{2}u\right)\left(1-m\sin\frac{\pi}{2}u\right) \\ f_1(u) &= 1-\left(1-\sin\frac{\pi}{2}u\right)\left(1-m\sin\frac{\pi}{2}u\right)-\left(1-\cos\frac{\pi}{2}u\right)\left(1-n\cos\frac{\pi}{2}u\right) \\ f_2(u) &= \left(1-\cos\frac{\pi}{2}u\right)\left(1-n\cos\frac{\pi}{2}u\right) \end{aligned} \tag{1}$$

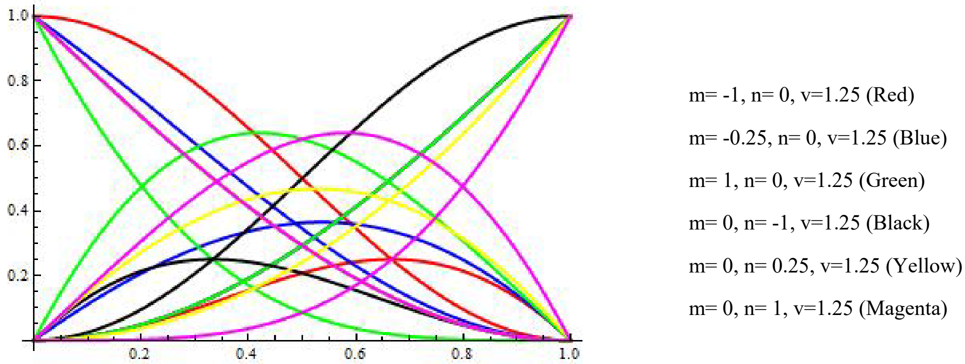


Fig 1: The Basis Graph of RQTB

2.1.2 Definition 2: Rational Quadratic Trigonometric Bézier (RQTB) curve

The RQTB Curve is describing as follows:

$$f(\mathrm{u}) = \frac{f_0P_0+f_1P_1v+f_2P_2}{f_0+f_1v+f_2}, \quad u \in [0,1], \;\; m,n \in [-1,1] \tag{2}$$

where $P_i (i = 0,1,2)$ are the control points, $f_i (i = 0,1,2)$ are the basis functions and v $(v \geq 0)$ is the scalar, which also known as the weight of the function (Bashir et al., 2012).

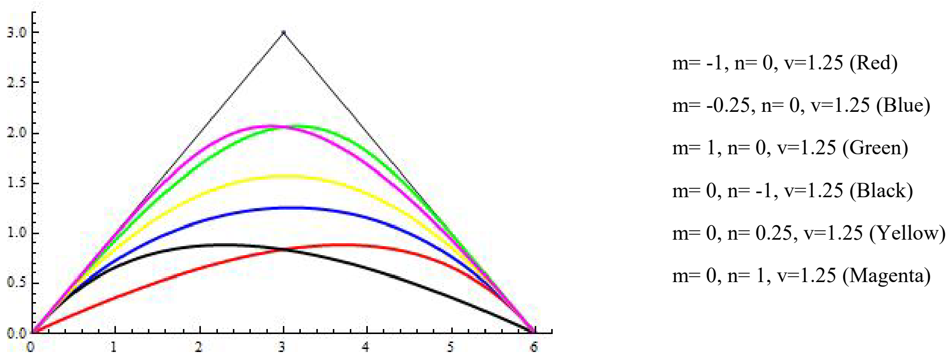


Fig. 2. The curves of RQTB

https://doi.org/10.24191/jcrinn.v10i2.541

2.2 The rational trigonometric curve in font design

A stylized script character “V” was chosen as the representative test case for this study. This character was selected because it contains both curved and angular structures, making it a challenging yet suitable subject for testing the flexibility of RQTB curves. Its geometry includes loops, smooth transitions, and sharp inflections features that allow a meaningful evaluation of how global shape parameters affect contour accuracy and smoothness.

The contour was modelled using 20 quadratic rational trigonometric curve segments and 40 control points, as listed in Table 1. These control points were carefully derived from the reference font image (FiG. 3), which served as the benchmark for comparison. Each control point was selected by tracing key structural features of the character, such as curve inflections, stroke tapering, and junctions between different contour regions. The distribution of control points aimed to balance structural fidelity which is accurately capturing the font’s geometry with computational feasibility.

By relying on this structured control point layout, the modelling process ensured that the RQTB curves captured the essential stylistic features of the letter while allowing systematic parameter variation in later analysis.

Fig 3. The real font design (reference figure)

Table 1. The control points of reference figure

Curve	Control Points P(x, y)	Curve	Control Points P(x, y)
Curve C1	p0 = (13, 20) ; p1 = (9, 15) ; p2 = (5, 21)	**Curve C11**	p20 = (29.5, 7) ; p21 = (22, -1) ; p22 = (18, 8)
Curve C2	p2 = (5, 21) ; p3 = (1, 28) ; p4 = (6, 37)	**Curve C12**	p22 = (18, 8) ; p23 = (12, 18.5) ; p24 = (17, 24.5)
Curve C3	p4 = (6, 37) ; p5 = (15, 55) ; p6 = (20, 39)	**Curve C13**	p24 = (17, 24.5) ; p25 = (21, 30) ; p26 = (17, 38)
Curve C4	p6 = (20, 39) ; p7 = (24, 27) ; p8 = (20, 21)	**Curve C14**	p26= (17, 38) ; p27 = (14, 46) ; p28 = (8.5, 38.8)
Curve C5	p8 = (20, 21) ; p9= (17, 16) ; p10 = (19, 9)	**Curve C15**	p28 = (8.5, 38.8) ; p29 = (1, 35) ; p30 = (6, 25)
Curve C6	p10 = (19, 9) ; p11 = (22, -1) ; p12 = (27, 6)	**Curve C16**	p30 = (6, 25) ; p31 = (8.5, 17) ; p32 = (13, 20)
Curve C7	p12 = (27, 6) ; p13 = (35, 13) ; p14 = (36, 22)	**Curve C17**	p32 = (35, 47) ; p33 = (39, 58) ; p34 = (42, 45)
Curve C8	p14 = (36, 22) ; p15 = (27, 37) ; p16 = (32.5, 48)	**Curve C18**	p34 = (42, 45) ; p35 = (44, 37) ; p36 = (40, 30)
Curve C9	p16 = (32.5, 48) ; p17= (39, 61) ; p18 = (43, 49)	**Curve C19**	p36 = (40, 30) ; p37 = (36.5, 23) ; p38 = (34, 32.5)
Curve C10	p18 = (43, 49) ; p19 = (50, 28) ; p20 = (29.5, 7)	**Curve C20**	p38 = (34, 32.5) ; p39 = (32.5, 39) ; p40 = (35, 47)

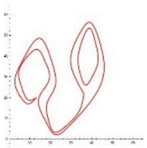
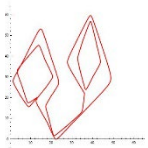
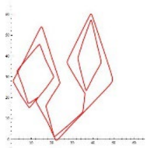
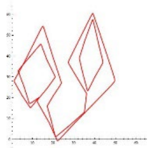
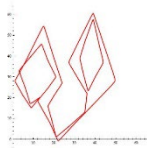
https://doi.org/10.24191/jcrinn.v10i2.541

2.3 The influence shape parameter on font design

The shape parameters play a crucial role in rational trigonometric curve modelling by providing designers with intuitive and continuous control over the overall geometry of a curve without modifying the control points. Unlike local shape parameters, which influence only a segment of the curve, global shape parameters affect the entire curve uniformly, making them particularly effective in applications like font design where consistent visual flow and smoothness are important. These parameters denoted as m, n, and weight, v in rational quadratic trigonometric formulations, modulate the influence of basis functions derived from sine and cosine terms. By adjusting these values, designers can sharpen, flatten, or soften curve segments to achieve desired stylistic effects in letter contours. This flexibility allows for the generation of font shapes with varying curvature profiles, making global shape parameters a valuable design tool in balancing artistic style and geometric accuracy.

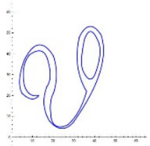
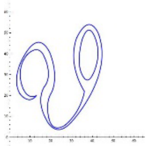
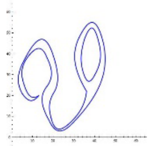
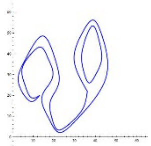
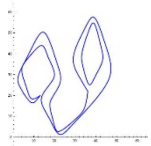
To validate the effectiveness of the generated font contour designs, a visual comparison was conducted against an actual styled font image representing the letter "V" (see Fig. 3). The reference font displays a graceful, fluid structure with a smooth rounded hook on the left and an elegant vertical curve on the right. This image serves as a valuable benchmark for assessing the degree of visual similarity between the generated and actual contours.

Table 2. The font contour for m = 0.5, n= 0.5 and various values of v

Fonts Contour Designs					
v	1	5	10	15	20

In Table 2, which illustrates the effect of varying the parameter v while holding m = 0.5 and n = 0.5 constant, the contour produced at v = 1 most closely resembles the original font. It retains the desired softness and fluidity in its structure. As the value of v increases, the contour gradually becomes more angular and rigid. Consequently, the visual appeal of the design diminishes, indicating that higher v values are less effective in replicating the original font shape.

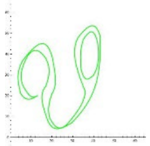
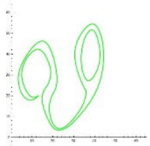
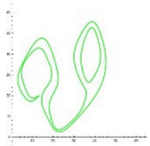
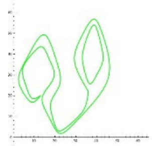
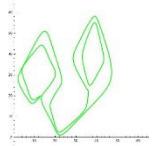
Table 3. The font contour for m = 0.5, v = 1and various values of n

Fonts Contour Designs					
n	-1	-0.5	0	0.5	1

Furthermore, Table 3 demonstrates the influence of parameter n, with m = 0.5 and v = 1 fixed. Notably, the contours generated using n = −0.5 and n = −1 exhibit a more refined and proportionate structure. These values introduce subtle inward curvature that enhances the resemblance to the original font. Conversely, when n increases to 0.5 and 1, the contours appear unbalanced and structurally inconsistent, particularly near the upper loop and lower tail. This indicates that negative values of n are more appropriate for preserving symmetry and design integrity.

https://doi.org/10.24191/jcrinn.v10i2.541

Table 4. The font contour for n = 0.5, v = 1 and various values of m

Fonts Contour Designs					
m	-1	-0.5	0	0.5	1

In addition, Table 4 highlights the effect of varying parameter m while maintaining n = 0.5 and v = 1. The designs corresponding to m = −1 and m = −0.5 generate more vertically compact and well-proportioned contours, which align more closely with the actual font's geometry. However, increasing mm to 0.5 and 1 leads to vertically stretched and distorted shapes, resulting in diminished visual quality. The loop region becomes particularly exaggerated, detracting from the overall balance and legibility.

In summary, the most accurate and aesthetically consistent results are obtained when v = 1, n = -0.5 or −1, and m = -0.5 or −1. These parameter combinations effectively preserve the stylistic features, smooth transitions, and structural harmony characteristic of the actual font. On the other hand, higher values of m and n tend to compromise these qualities, producing uneven contours that deviate from the intended design.

2.4 Pointwise deviation analysis for shape parameter evaluation

To determine the most effective combination of shape parameters and weights in constructing font contours with RQTB, this study applies a pointwise distance analysis between the generated curves and the reference contours. This technique quantitatively evaluates the fidelity of the approximation by measuring how closely each sampled point on the designed curve aligns with the actual outline.

Given a parametric curve constructed from RQTB, denoted by

$$f(\mathrm{u}) = (x(u)_c, y(u)_c), \qquad u \in [0,1] \tag{3}$$

and the actual contour curve, represented as

$$A(u) = (x(u)_a, y(u)_a) \tag{4}$$

the pointwise deviation is expressed as the Euclidean distance:

$$D_i = \sqrt{(x(u)_c - x(u)_a)^2 + (y(u)_c - y(u)_a)^2} \tag{5}$$

The results are displayed as follows, showing the computed pointwise deviations (D1, D2, D3), along with the Mean and RMSE values for different parameter settings.

Table 5. Pointwise deviation and error analysis for different values of v with m = n = 0.5

v	D1	D2	D3	Mean	RMSE
1	3.490702	1.096141	1.900421	2.162421	1.470517
5	6.820359	3.134599	5.760139	5.238366	2.288748
10	7.030348	3.241886	6.080132	5.450789	2.334692
15	8.000306	3.241886	6.080132	5.774108	2.402937
20	7.350333	3.671241	6.620121	5.880565	2.424988

https://doi.org/10.24191/jcrinn.v10i2.541

When fixing m = n = 0.5 and varying v, the RMSE values range from 1.47 to 2.42. The smallest error is observed at v= 1 (RMSE = 1. 470517), indicating that the curve generated under this setting best matches the reference contour. As v increases from 5 to 20, the RMSE consistently rises, suggesting that larger values of v reduce the approximation fidelity and lead to a greater deviation from the target outline.

Table 6. Pointwise deviation and error analysis for different values of n with v = 1, m = 0.5

n	D1	D2	D3	Mean	RMSE
-1	0.912688	1.160078	1.220656	1.097807	1.047763
-0.5	0.485077	0.838492	0.056569	0.460046	0.678267
0	1.23199	0.138351	0.711126	0.693822	0.83296
0.5	3.270749	1.203039	2.22036	2.231383	1.493781
1	5.10048	1.846677	3.610222	3.519126	1.875933

For the case where v = 1, m = 0.5 and n varies, the results demonstrate that negative values of n provide better approximation accuracy. Specifically, the lowest RMSE occurs at n = −0.5 (RMSE = 0.678267), which highlights the effectiveness of slight negative adjustments in improving the curve's alignment. In contrast, positive n values, such as n = 0.5 or n = 1, show considerably larger RMSE values (1.493781 and 1.875933, respectively), indicating poorer approximation.

Table 7. Pointwise deviation and error analysis for different values of m with v = 1, n = 0.5

m	D1	D2	D3	Mean	RMSE
-1	0.703491	0.946322	0.900888	0.850234	0.922081
-0.5	0.783135	0.838492	0.056569	0.559398	0.747929
0	1.991231	0.054231	1.140702	1.062054	1.03056
0.5	3.490702	1.905656	2.320345	2.572234	1.603819
1	4.990491	2.061406	3.400235	3.484044	1.866559

Similarly, for v = 1, n = 0.5 with varying m, the accuracy trend is consistent. The best performance is achieved at m = −0.5 (RMSE = 0.747929), followed by m = −1 (RMSE = 0.922081). Positive values of m, particularly m = 0.5 and m = 1, result in higher RMSE values (1.603819 and 1.866559), indicating reduced fidelity of the constructed curve.

Overall, the findings indicate that lower or slightly negative values of the parameters m and n, along with smaller values of v, are more effective in minimizing pointwise deviations. This suggests that fine-tuning the shape parameters towards negative or near-zero ranges contributes to better alignment of the constructed curves with the actual contours, enhancing the precision of the RQTB method in font contour design.

2.5 The computation time on font design

This study focuses on evaluating the computational efficiency of RQTB methods in font curve design by analysing CPU time with varying global shape parameters (m, n) and weight (v). By systematically adjusting these parameters, the study examines how changes in curve flexibility and stroke refinement impact the time required for curve generation and curvature computation. The CPU time metric serves as a secondary aspect for assessing the computational cost associated with different shape parameter settings.

https://doi.org/10.24191/jcrinn.v10i2.541

This focused analysis aims to identify which parameter configurations optimize performance, offering insights into the trade-offs between shape control and computational efficiency in rational trigonometric curve formulations.

Table 7. The Computation Time of the Image

Shape Parameter, v	CPU Time	Shape Parameter, n	CPU Time	Shape Parameter, m	CPU Time
1	0.07	-1	0.069	-1	0.085
5	0.115	-0.5	0.068	-0.5	0.088
10	0.111	0	0.063	0	0.068
15	0.102	0.5	0.069	0.5	0.091
20	0.103	1	0.069	1	0.088
Average CPU Time	**0.1002**	**Average CPU Time**	**0.0676**	**Average CPU Time**	**0.084**

Table 7 shows that the choice of shape parameters significantly affects both CPU time and contour quality. For parameter v, the best contour with the lowest computation time was achieved at v = 1, compared to the overall average of 0.1002 seconds. For parameter n, although the minimum CPU time occurred at n = 0, the best contours were observed at n = –0.5 and –1, both close to the average of 0.0676 seconds. Similarly, for parameter m, values of –0.5 and –1 yielded better contours than other settings, with CPU times slightly above the average of 0.084 seconds. In conclusion, the shape parameter of v = 1, n = –0.5, and m = –0.5 provides the best trade-off between high-quality contours and efficient computation.

2.6 The interactive tool for shape parameter

To facilitate a deeper understanding of the influence of the shape parameters on rational trigonometric font curves, an interactive tool has been developed. This tool allows users to dynamically adjust shape parameters and observe the immediate effects on the resulting curves. By incorporating this tool, users can experiment with different values for the shape parameters, specifically m, n, and weight, v, and see how these changes impact the smoothness, curvature, and overall appearance of font contours. Additionally, the tool provides real-time feedback, including curvature plots and CPU time measurements, offering valuable insights into the trade-offs between computational efficiency and visual aesthetics in font design.

https://doi.org/10.24191/jcrinn.v10i2.541

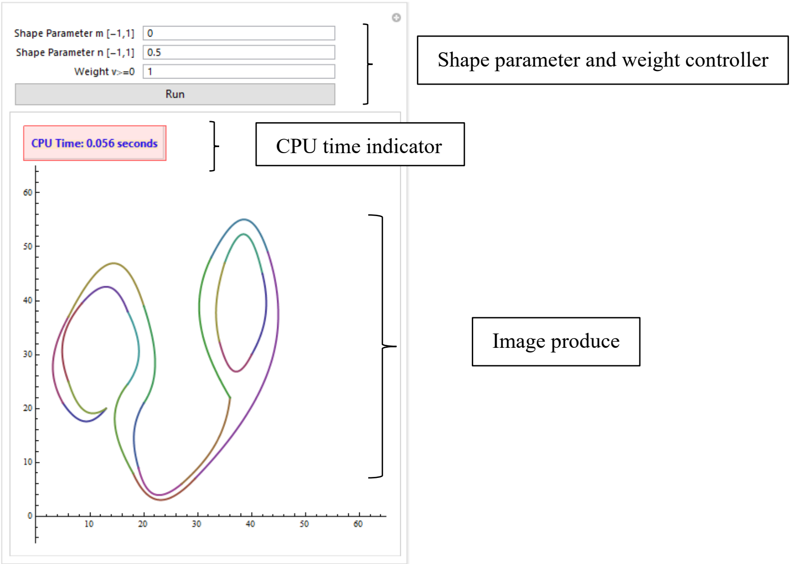


Fig 4. The interactive tool for shape parameter

3. RESULTS AND DISCUSSION

In this study, a selected set of shape parameter values was tested to reconstruct the font character using 20 RQTB segments and 40 control points. Among the tested configurations, a particular set of parameters produced the closest approximation to the actual font shape with smooth curvature and visually consistent strokes. This design also recorded the lowest CPU time, indicating efficient computation. Although not all possible parameter values were explored, the results show that high-quality font curves can be achieved with minimal adjustment, demonstrating the practicality and effectiveness of the RQTB method for font design.

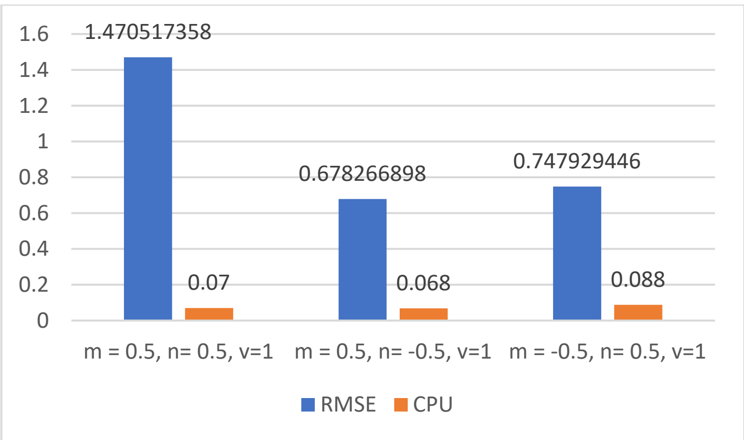
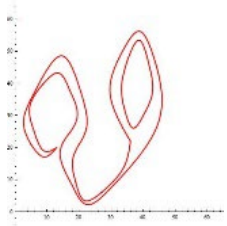
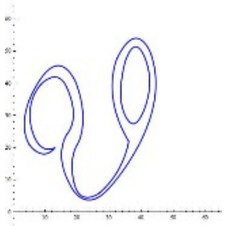
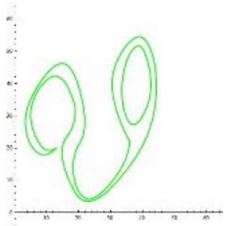


Fig 5. Evaluation of shape parameters based on RMSE and CPU time

https://doi.org/10.24191/jcrinn.v10i2.541

Table 8. The result of font contour design

IMAGE				
SHAPE PARAMETER		**M = 0.5, N= 0.5, V=1**	**M = 0.5, N= -0.5, V=1**	**M = -0.5, N= 0.5, V=1**

The analysis of shape parameter combinations reveals a clear trade-off between contour accuracy and computational efficiency. Among the tested configurations, the combination $m = 0.5$, $n = -0.5$, $v = 1$ achieved the lowest RMSE value (0.678), indicating the highest fidelity to the original font contour. This configuration also recorded the lowest CPU time (0.068s), making it the most efficient and accurate overall.

Similarly, the combination $m = -0.5$, $n = 0.5$, $v = 1$ produced a smooth and visually consistent contour with a slightly higher RMSE (0.748) and CPU time (0.088s), still within acceptable performance limits. In contrast, configurations such as $m = 0.5$, $n = 0.5$, $v = 1$ resulted in significantly higher RMSE (1.4705), indicating poor contour approximation despite a relatively low CPU time (0.07s). The combined RMSE and CPU time graph clearly illustrates that lower or slightly negative values of m and n, paired with $v = 1$, consistently yield better performance in terms of both accuracy and speed. These findings support the use of RQTB curves with optimized shape parameters for efficient and high-quality font design.

4. CONCLUSION

This study investigated the impact of shape parameters m, n and weight, v on font contour design using RQTB curves. Through systematic evaluation of contour fidelity and computational efficiency, the findings demonstrate that optimal font reconstruction is achieved when shape parameters are carefully tuned. Specifically, the combinations $m = 0.5$, $n = -0.5$, $v = 1$ and $m = -0.5$, $n = 0.5$, $v = 1$ produced the most accurate contours with minimal deviation from the original font shape, while also maintaining low CPU time. The comparative analysis, supported by RMSE and CPU time graphs, highlights a clear trade-off between accuracy and performance. Lower or slightly negative values of m and n, paired with $v = 1$, consistently yielded superior results. These configurations not only preserved the stylistic integrity of the font but also ensured efficient computation, making them ideal for practical applications in digital typography, CAD, and computer graphics. Future work may explore adaptive parameter selection techniques, integration with machine learning models for automated tuning, and the extension of RQTB curves to higher-degree or fractional forms to enhance flexibility and visual fidelity in complex font designs.

5. ACKNOWLEDGEMENTS/FUNDING

The authors would like to acknowledge Universiti Teknologi MARA Cawangan Terengganu Kampus Kuala Terengganu for their support.

https://doi.org/10.24191/jcrinn.v10i2.541

6. CONFLICT OF INTEREST STATEMENT

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare the absence of conflicting interests with the funders.

7. AUTHORS' CONTRIBUTIONS

Noor Khairiah Razali: Conceptualisation, methodology, formal analysis, investigation and writing-original draft; **Nur Ain Fitrah Azhari**: Conceptualisation, methodology, and formal analysis; **Siti Musliha Nor-Al-Din**: Conceptualisation, formal analysis, and validation; **Nursyazni Mohd Sukri**: Conceptualisation, supervision, writing- review and editing, and validation.

8. REFERENCES

Ahmad, A., & Gobithaasan, R. (2018). Rational quadratic Bézier spirals. *Sains Malaysiana*, *47*(9), 2205–2211. https://doi.org/10.17576/jsm-2018-4709-31

Ahmed, I. A. A., Omar, A. H. A., & Ali, J. M. (2022). Arabic font design using quadratic Bézier-like curve. *Journal of Pure & Applied Sciences*, *21*(4), 50–56.

Bashir, U., Abbas, M., Abd Majid, A., & Ali, J. M. (2012, July). The rational quadratic trigonometric Bézier curve with two shape parameters. In *2012 Ninth International Conference on Computer Graphics, Imaging and Visualization* (pp. 31–36). IEEE. https://doi.org/10.1109/CGIV.2012.21

Ceylan, A. Y., Turhan, T., & Tukel, G. O. (2021). On the geometry of rational Bézier curves. *Honam Mathematical Journal*, *43*(1), 88–99. https://doi.org/10.5831/HMJ.2021.43.1.88

Dube, M., & Gupta, N. (2022). Font design through RQT Bézier curve using shape parameters. In *Advances in Mathematical Modelling, Applied Analysis and Computation*. In *Proceedings of ICMMAAC 2021* (pp. 319–335). Springer. https://doi.org/10.1007/978-981-16-8641-1_23

Munir, N. A. A. A., Hadi, N. A., Nasir, M. A. S., & Halim, S. A. (2024, January). Modelling fruit outline using cubic trigonometric spline. In *AIP Conference Proceedings* (Vol. 2905, No. 1). AIP Publishing. https://doi.org/10.1063/5.0172468

Omar, A., Ahmed, I., & Abuzaiyan, F. (2023). The quadratic trigonometric Bézier-like curve with two shape parameters. *Journal of Pure & Applied Sciences*, *22*(3), 215–221.

Salomon, D. (2006). *Curves and surfaces for computer graphics*. Springer. https://doi.org/10.1007/0-387-28452-4

Sánchez-Reyes, J. (2022). The uniqueness of the rational Bézier polygon. *Computer Aided Geometric Design*, 96, 102118. https://doi.org/10.1016/j.cagd.2023.102266