UNIVERSITI TEKNOLOGI MARA

G^2 CUBIC TRIGONOMETRIC SPLINE FOR CURVE AND SURFACE FITTING

NUR AZLIANA AZLIN BINTI AHMAD MUNIR

PhD

March 2026

UNIVERSITI TEKNOLOGI MARA

G^2 CUBIC TRIGONOMETRIC SPLINE FOR CURVE AND SURFACE FITTING

NUR AZLIANA AZLIN BINTI AHMAD MUNIR

Thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy
(Mathematics)

Faculty of Computer and Mathematical Sciences

March 2026

CONFIRMATION BY PANEL OF EXAMINERS

I certify that a Panel of Examiners has met on 24 April 2025 to conduct the final examination of Nur Azliana Azlin binti Ahmad Munir on her Doctors of Philosophy thesis entitled "G^2 Cubic Trigonometric Spline for Curve and Surface Fitting" in accordance with Universiti Teknologi MARA Act 1976 (Akta 173). The Panel of Examiners recommend that the student be awarded the relevant degree. The Panel of Examiners was as follows:

Azlan Bin Ismail, PhD
Associate Professor
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
(Chairman)

Adem Kilicman, PhD
Professor
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
(Internal Examiner)

Zainor Ridzuan Yahya, PhD
Associate Professor
Faculty of Intelligent Computing
Universiti Malaysia Perlis
(External Examiner)

PROFESSOR DR HJH ZURAEDA IBRAHIM
Dean
Institute of Postgraduates Studies
Universiti Teknologi MARA

Date: 27 March 2026

AUTHOR'S DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the results of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any degree or qualification.

I, hereby, acknowledge that I have been supplied with the Academic Rules and Regulations for Post Graduate, Universiti Teknologi MARA, regulating the conduct of my study and research.

Name of Student	:	Nur Azliana Azlin binti Ahmad Munir
Student ID. No.	:	2015353435
Programme	:	Doctor of Philosophy (Mathematics) – CS952
Faculty	:	Computer and Mathematical Sciences
Thesis Title	:	G^2 Cubic Trigonometric Spline for Curve and Surface Fitting
Signature of Student	:	……..
Date	:	March 2026

ABSTRACT

Free-form curves and surfaces are frequently required in various industry for a wide range of applications in the fields of science and engineering. Nevertheless, there are difficulties in producing complex curves and surfaces with both smoothness and great precision. Thus, this thesis proposed a novel cubic trigonometric spline with a single shape parameter to address the given circumstances. The basis functions of cubic trigonometric spline are constructed based on Bezier-like functions, that satisfied all the properties needed. The presence of shape parameter plays an essential role in modifying the shape of the curve. Various shapes of curves and surfaces could be created even with the presence of limited shape parameter. Parametric and geometric continuity conditions have also been implemented and achieved to guarantee the smoothness of joining the piecewise curves and surfaces. There are C^0, C^1, C^2, G^0, G^1 and G^2 continuity. The presence of scalar factor in continuity conditions allows the degrees of freedom in handling the formation of curves and surfaces. The conditions for shape preservation of positive data set are derived. The process of modelling and handling complex product using continuity constraint is quick and easy to control, making it easier to meet the actual need even applying to real data. The mathematical outline of 2D boundary images of 'ى', 'epsilon', and 'delta' are reconstructed using proposed cubic trigonometric spline with other three different cubic trigonometric spline in literatures. All the chosen cubic trigonometric spline shared same criteria regarding the type of spline, number of shape parameter, degrees of spline and having Bezier-like properties. The splines undergo the curve fitting process including corner points detection, control points calculation, curve fitting, and error calculations. The approximation errors are calculated by computing the distance from the actual image and the fitted curves to determine each spline's effectiveness. The performance of each spline is compared visually and numerically. The results show that the proposed functions provide higher accuracy representations of curves which are continuous, smooth and pleasant.

ACKNOWLEDGEMENT

In the name of Allah S.W.T, the Most Gracious and the Most Merciful. Alhamdulillah, I am most grateful to Allah, the One who gives me strength and blessings throughout my studies. Peace and blessings of Allah be upon the last Prophet Muhammad. For this opportunity, I would like to thank those people involved who extended their help and encouraged me to complete this Ph.D. journey.

First and foremost, I would like to express my deepest gratitude and appreciation to my respected main supervisor, Prof. Madya Ts. Dr. Normi binti Abdul Hadi, for her patience, immense knowledge, motivation, and quality supervision. It has been an honour to be her student. I am thankful for her time and valuable guidance, which helped me understand the research area in writing this thesis.

Secondly, I would like to thank my former supervisor, Dr. Fatimah binti Yahya, for all the knowledge and encouragement during the beginning of my study. My sincere thanks also for my co-supervisor, Prof. Madya Ts. Dr. Mohd Agos bin Salim Nasir for his constructive comments, ideas, and encouragement in order to improve the quality of my thesis.

Next, special thanks are dedicated to the loved ones around me. For my parents, Ahmad Munir bin Taeh, thank you for the endless love, everlasting prayers, and encouragement. My deepest thanks to my beloved husband, Mohd Amirul Aizzad bin Hamdan for his patience, sacrifice, and greatest support. I would like to express my heartfelt appreciation to my beloved son, Muhammad Adam Ayyash bin Mohd Amirul Aizzad, who always been my source of strength and inspiration throughout this journey. Not forgetting my sisters, brothers, parents-in-law, and friends for supporting me spiritually throughout writing this thesis and my life in general. Thank you very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbols

θ	Angle in degree
u	Variable
C^0	Position parametric continuity
C^1	Tangent parametric continuity
C^2	Curvature parametric continuity
C^3	Third derivative parametric continuity
C^4	Fourth derivative parametric continuity
G^0	Position geometric continuity
G^1	Tangent geometric continuity
G^2	Curvature geometric continuity
G^3	Geometric Continuity
B	Bezier Curve
P_i	Control points of cubic trigonometric spline
S_i	Quadratic trigonometric basis functions
π	Pi
β	Shape parameter for quadratic trigonometric spline
V_i	Control points of S_i
F	Quadratic trigonometric curve
m	Shape parameter for cubic trigonometric spline

μ_i	Scalar factor
ε	Epsilon
δ	Delta

LIST OF ABBREVIATIONS

Abbreviations

CAGD	Computer Aided Geometric Design
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
2D	2-Dimensional
3D	3-Dimensional
T-Bezier	Cubic Trigonometric Bezier
GBTB	Generalized Blended Trigonometric Bernstein-Like Basis Functions
TB curves	Cubic Trigonometric Polynomial
GHT-Bezier	Generalized Hybrid Trigonometric Bezier
GT-basis	Generalized Trigonometric Basis
DOF	Degrees of Freedom
SG-Bezier	Shape-adjustable Generalized Bezier
GDCT-Bezier	Developable Cubic Trigonometric Bezier
LSM	Least-Squares Method
PCACSS	Principle Component Analysis Based Curvature Scale Space
BOA	Butterfly Optimization Algorithm
DE	Differential Evolution
SSE	Sum Square Error

SCA	Sine Cosine Algorithm
WOA	Whale Optimization Algorithm
NURBS	Non-uniform Rational B-spline

CHAPTER 1
INTRODUCTION

1.1 Research Background

In synchrony with the advanced technology development and revolution nowadays, industries demand geometric tools and methods which are able to reduce processing time and product quality. Computers are the primary graphic tools in industries that highly contribute in developing and modelling geometric shapes and physical surfaces in numerous fields, including electricals, automotive, engineering and healthcare. The advancement of computer technology allows for storing data for geometric modelling, making it easy to visualize the algorithm of the systems. The approaches about the mathematical and computational methods of describing entities, modifying, and developing any particular design are identified as "Computer Aided Geometric Design (CAGD)." CAGD involves expressing the shape of curves and surfaces using mathematical algorithms that are able to generate desired smooth and pleasant representation. The construction of curves forms the surface and the performance of the curve affects the development of the surface.

Through the years, 2-dimensional (2D) and 3-dimensional (3D) models have been developed, which are both critical in contributing to the industries. Scientific visualization applications require representations of the data in terms of lines, curves, or surfaces like waves and wind patterns, 3-D imaging of rainfall density over a region, geographical contours and mapping, animation and games designs, pattern recognition and matching, anatomical reconstruction of organs and tumours in the medical industry, manufacturing, and engineering designs, mortality and longevity for actuarial purposes and tracking trend and popularity for business considerations. Accurate representations are not only insightful but also impactful and effective.

Chambers et al., (1995) stated that in CAGD, complex objects could be described as simple primitives using points and lines. Most of the curves in CAGD or CAD are a blend of some points P_i known as control points at some i^{th} location. The control points and control polygon are essential in determining the shape of the curve. Literature shows various types of splines involved in designing curves and surfaces;

Lagrange, Newton, Hermite spline curve, Bezier, B-splines, Beta-splines, rational, and other specific types. The spline is a mathematical model that enables users to generate and develop the shape and the contours of complicated curves and surfaces. In CAGD, Bezier and B-splines are preferably used due to the flexibility properties in design purposes compared to other splines. B-spline offers local control points, indicating that when the coordination of a control points is modified, this will impact some parts of the curve; nevertheless, for Bezier, it ultimately affects all the curve's structure. A Bezier-like curve is chosen in this research because it is mathematically much simpler to deal with.

Generating complex shapes may be challenging and are unable to accurately depicted using only one curve but by many connected smoothly as splines. When merging two connected curves, specific continuity requirements must be met for the purpose to maintain the geometrical features. Two different types of continuity, parametric and geometric, are required to ensure the smoothness of the two merging curves and their parameterization. In other circumstances, when two curves are merged without appropriate continuity, the curve's continuous curvature will not be guaranteed.

Another critical component of CAGD, particularly for tackling reverse engineering issues and image processing, is curve reconstruction (Usman et al., 2020). It has emerged as one of the key issues addressed by current academics, particularly in computer graphics pattern recognition and CAGD, for reversing engineering difficulties. Reconstruction is defined as a process of rebuilding a curve or spline from a given collection of data points using a mathematical equation. Curve fitting is a step that must be taken into account in accurately discovering the set of data points to construct an original image. In other words, the process of creating a curve that passes through or fulfils known non-noisy data points.

Throughout this research new trigonometric formulas, specifically quadratic and cubic basis function are generated. This study focuses on the construction, analysis, and application of cubic trigonometric splines that satisfy up to G^2 continuity. Despite rapid progress in computing, good basic mathematical models are still needed. Technology can process data faster, but it cannot replace accurate and flexible geometric formulas. These new basis functions are employed to perform desired curves and surfaces by modifying the shape parameters which are useful for modern design. A mathematical framework has developed that can accurately approximate or interpolate sets of data

points in both curve and surface settings. The discussion has focused on four distinct sorts of design difficulties, which are the development of

i) Quadratic and cubic trigonometric curves with contour adjustability.
ii) Cubic trigonometric curves using parametric and geometric continuity conditions.
iii) Cubic trigonometric spline to develop surface-shaped data.
iv) Cubic trigonometric spline in the reconstruction of 2D image.

1.2 Motivation and Statement of the Problem

Numerous trigonometric spline schemes have been proposed in the literature, but a new formulation is still needed when existing methods do not fully address certain modelling requirements. Most available schemes often rely on multiple shape parameters, which increases their complexity and can result in inconsistent performance across different shapes. In applications such as font design reconstruction, where the geometry varies significantly, these limitations reduce the effectiveness and usability of current methods. Therefore, there is need for a spline scheme that is simpler, easier to control, and still capable of producing accurate and smooth curves.

Besides that, in line with the needs of the industry related to image construction, designing, and modelling objects, many types of basis functions have been developed for curve and surface fitting. Meanwhile, the effectiveness and accuracy are not well concerned. This absence of evaluation may result in unreliable curve or surface reconstruction when applied to real-world industrial modelling tasks. Therefore, this research aims to extend the work by comparing and evaluating the performance of various spline methods in literature.

Therefore, the driving force that led to the implementation of this research is to fulfil the needs of the industry. Although computer technology is advancing rapidly, the fundamental mathematical models for geometric representation remain crucial in design and modelling applications. Hardware improvements cannot substitute for accurate, flexible, and controllable curve and surface formulations. The proposed cubic trigonometric basis functions provide improved shape control and smoothness, making them highly relevant for modern CAD/CAM systems, 3D printing, image reconstruction, and digital modelling. In fact, as technology advances, the demand for

more robust geometric algorithms increases, and this study contributes directly to that need.

Schoenberg's early 1964 with the introduction of the trigonometric spline has attracted much interest from researchers due to its superior performance. Next, in previous literature on the CAGD area, numerous researchers performed their new basis trigonometric basis function that complied with specific criteria to generate high-quality spline curves and surfaces. The interpolation of a given collection of data points while preserving smoothness and continuity is the mark of a high-quality curve (Schneider & Kobbelt, 1999). The other crucial elements for any curve building in computer graphics are local control over the shape, stability, and simplicity in representation.

In this study, the new cubic trigonometric basis function is generated with Bezier-like criteria and satisfies the requirements properties needed to be valuable and improve the efficacy for the designers. The term “Bezier-like” refers to curve or surface schemes that imitate the core geometric and mathematical properties of classical Bezier curve, even if they are based on different basis functions such as trigonometric, exponential, or modified polynomials. These schemes are designed to preserve key characteristics of Bezier curves, including endpoint interpolation (the curve passes through the first and last control points), the convex hull property (the curve remains within the polygon formed by its control points), variation diminishing behaviour (the curve does not oscillate more than the control polygon), and affine invariance (transformations like translation or rotation applied to control points produce the same effect on the curve). Examples of Bezier-like schemes include trigonometric Bezier curves, rational Bezier curves, Ball curves, and generalized Bernstein polynomial schemes. All of the given examples able to maintain the Bezier structure while offering enhanced shape control or modelling flexibility. The researchers widely used the Bezier technique since it can cope with many difficulties in approximating, interpolating, preserving the shape, and generating curves and surfaces.

In contrast, non-Bezier-like schemes such as Lagrange interpolation curves, Hermite curves, and Catmull-Rom splines do not maintain these properties. For instance, Lagrange curves interpolate all control points and often violate the convex hull property, while Hermite curves rely on derivatives rather than control points. A Bezier-like scheme, therefore, regardless of its mathematical formulation, behaves in a way that offers the intuitive and controllable design advantages of classical Bezier-like curves.

Despite the existing of many trigonometric spline schemes, developing a new trigonometric scheme remains significant, as existing schemes frequently do not achieve an optimal balance among shape flexibility, mathematical simplicity, continuity, and computational efficiency. Numerous earlier schemes exhibit limitations such as insufficient local shape control, poor handling of curved or periodic features, or increased complexity when generalized to rational or higher-order forms.

The inclusion of shape parameters in trigonometric Bezier offers a useful replacement to the traditional rational Bezier. A new scheme is adequate, even it develops with only one shape parameter, it provides effective local shape control without significantly increasing complexity. The development of new scheme allows for smooth and intuitive adjustments to the curve without changing control points. By including the shape parameters in trigonometric Bezier, it is more beneficial in providing simpler computation particularly compared to rational Bezier. In addition, according to research by (Romani et al., 2014) the inclusion of shape parameters by the trigonometric curve representations improved results and made the proposed trigonometric curves more flexible than polynomial interpolation.

Trigonometric functions are chosen as the basis functions since it ensures that curves are smooth even at high acceleration levels (Menasri et al., 2014). The trigonometric spline can resolve the challenges in performing conic shapes and transcendental curves. Bashir et al., (2013) claimed that polynomial splines are incapable of generating circular arcs and conical shapes, which are the predominant geometric elements found in most modelling systems. Moreover, trigonometric curves are compatible with many formulations. In areas such as architecture, engineering, and medicine, their equations and systems are frequently stated in trigonometric terms.

Motivated by those researches, a new development of quadratic and cubic trigonometric basis functions is proposed which aims to deal with the curve and surface fitting. Moreover, the proposed scheme supports higher-order continuity (such as G^2), preserves Bezier-like characteristics, and operates efficiently with a minimal number of control parameters, making it a practical and elegant choice for real-world modelling applications. In this context, introducing a new trigonometric spline scheme with even a single well-behaved free parameter can substantially improve usability and flexibility over existing methods that may lack adequate control, simplicity, or smoothness.

In addition, the construction and development of many kinds of curve and surface involve the parametric and geometric continuity up to second order. By establishing appropriate continuity schemes ensures that the resulting curve and surface remain smooth at the joints.

1.3 Significant Findings and Benefits

This research contributes to geometric modelling applications, particularly in producing appealing 2D shapes and objects, which also aids designers in obtaining highly sought curves and surfaces. The developed functions help to enhance the curve and surface capabilities that a polynomial cannot by involving trigonometric terms in the functions for building curves. Trigonometric function application also enables the production of a smooth curve even at a high parametric speed or higher degree continuity, which is thought appropriate for curve and surface design.

Moreover, the incorporating of shape parameters within the basis functions and continuity conditions brings convenience in forming the shape of the curve and surface by easily updating the values instead of altering control points. Besides, the implementation of this method is uncomplicated for both 2D and 3D data since the computation is simple due to its non-rational structure.

This research holds particular significance in reconstructing images, as the usage of cubic trigonometric curves outperform alternative methods in terms of accuracy. Besides, further research is possible since the developed trigonometric basis function is newly invented.

1.4 Research Framework and Objectives

The generated of new cubic trigonometric Bezier-like basis functions that include a shape parameter are presented in this thesis. These functions will then be used to generate the curves and surfaces. The objectives of this research are:

i) To develop a new cubic trigonometric basis function with an adjustable shape parameter.
ii) To employ parametric and geometric continuity conditions for cubic trigonometric piecewise curves and surfaces.

iii) To construct cubic curves and cubic trigonometric surfaces using the developed method in (i) and (ii).
iv) To fit cubic trigonometric curves and surfaces on 2D and 3D dataset image reconstruction algorithm using the proposed cubic trigonometric spline.
v) To analyse the performance of all the generated cubic trigonometric splines in automatically reproducing the shape of contour-based images in 2D.

1.5 Scope and Limitation

This research is concerned with developing trigonometric spline up to the degree of three. New quadratic and cubic trigonometric basis functions with a shape parameter are constructed in non-rational forms.

This study addresses two types of continuity conditions in the connection of curves and surfaces, namely parametric and geometric continuity. Within the scope of this work, continuity is considered up to the second order. The continuity conditions are denoted as C^0, C^1, C^2, G^0, G^1 and G^2. While it is mathematically feasible to extend the continuity to higher levels such as $G^3, G^4, .., G^n$, the present study restricts its focus to G^2 for reasons of practically and relevance. Yet, despite improvements through higher-order continuity, the measurement of surface smoothness remains difficult to quantify. For data shape preservation, the issue that has been discovered is only in maintaining positivity.

For the 2D image reconstruction, this study focuses on the outline of captured fonts, specifically the Greek letters epsilon ε, delta, δ, and the Arabic letter 'ى'. The proposed method is compared with other existing cubic trigonometric splines developed by researchers with similar characteristics, including the works of Majeed & Qayyum (2020), (Han et al., 2010), and Usman et al., (2020). The comparison primarily considers properties such as the degree of the splines, the number of shape parameters, and their Bezier-like behaviour. The reconstruction of the selected letters epsilon, ε, delta, δ, and 'ى' using these splines is conducted to identify the method that yields the smallest total distance between the fitted curve and the actual image data points.

All four basis functions are developed using trigonometric Bezier-type curves, exploiting trigonometric basis functions for better modelling of circular or wavy shapes.

All aim to improve flexibility, local control, and accuracy in geometric modelling, particularly useful in fields like CAD, animation, and industrial design. Each basis functions generated preserves essential Bezier-like properties such as the endpoint interpolation, convex hull property and variation diminishing property. The degree of the basis functions is similar that is cubic which consists of four control points. Besides, each method introduces one shape parameter but with different range values. Meanwhile, for both Majeed & Qayyum (2020) and Usman et al., (2020) use rational curves, allowing the use of weights to further influence curve shape.

The letters ε, δ, and '$ى$' are reconstructed using four different spline methods based on the given digitized data points. The accuracy of each method is evaluated through the computation of the approximation error, defined as the minimum deviation between the generated curve and the corresponding data points. A comparative analysis of the results is subsequently performed, and the entire reconstruction procedure is implemented using MATLAB software.

1.6 Thesis Outline

The thesis is structured into seven chapters, including this introductory chapter. The following provides an overview and concise concepts for each chapter of this thesis:

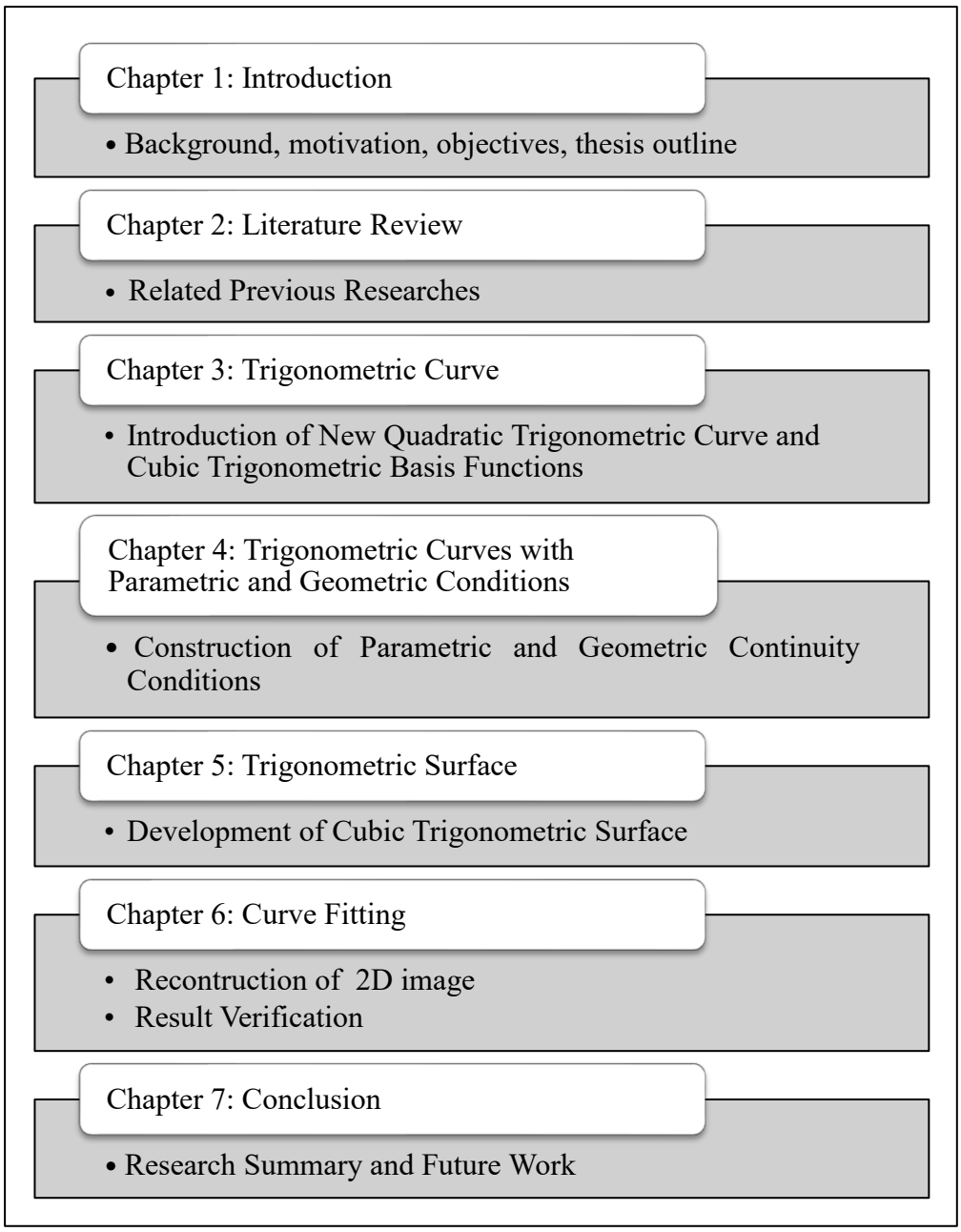


Figure 1. 1 Thesis Outline

Chapter 1 includes the introduction, motivation, objectives, and outline of this research.

In Chapter 2, the basic concepts related to the area of research, which is developing quadratic and cubic trigonometric spline, are presented. The literature review section in this chapter discusses previous works by other researchers. The review starts with the essential factors involved in developing cubic trigonometric spline, 2D image reconstruction algorithms, and the development of 3D-designed data.

Chapter 3 presents the development of new quadratic and cubic trigonometric spline basis functions with one shape parameter. The basis functions that satisfy most of the properties of Bernstein and Bezier are discussed and proved in detail. The alteration of the shape parameter values illustrates the behaviour of the quadratic and trigonometric spline.

Next, the parametric and geometric continuity conditions are developed in Chapter 4. Two or more curve segments of cubic trigonometric curves are constructed by using the continuity conditions, C^0, C^1, C^2, G^0, G^1 and G^2. Smoothness at the joints is an essential component of this depiction. The smoothness of the curve is preserved by using parametric and geometric continuity conditions with convenient values of shape parameters. For each continuity condition, the shape parameter's value is adjusted to observe the implications of the curve. Some examples of proposed curves are also given to demonstrate the effectiveness of curve design.

Chapter 5 concerns the construction of cubic trigonometric surface equations. The developed curve fitting algorithm in Chapter 4 is extended to the surface fitting algorithm. The surface equations developed will be fitted into 3-D objects, and the values of the shape parameter and scaling factor are altered to observe the effects on each surface generated. The surface equations are also applied in reconstruction of real data of human face and pelvic.

Chapter 6 deals with the process of image reconstruction. In this chapter, some new approaches are involved: a method for determining control points, a method for calculating fitting errors, and a way to refine curves. The algorithm process flow is presented. The technique is applied to font images. Four kinds of cubic trigonometric basis functions proposed by different researchers include the proposed basis functions used in reconstructing font images. The result is tabulated and discussed both visually and numerically.

Lastly, chapter 7 concludes with an overview of the entire body of research and offers recommendations for further research.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

The conceptual frameworks for this study are presented in this chapter. It begins with an overview of the Bezier curve, as the proposed cubic trigonometric basis functions are Bezier-like functions. Understanding the Bezier structure provides foundation for the formulation and analysis of the trigonometric curves introduced in this research.

Next, this chapter offers an exploration and extensive review of the literature related to cubic trigonometric curves and surfaces. This includes the discussions on the development of trigonometric spline functions, the theoretical basis of trigonometric formulations, the role of the shape parameter, and the formulation of parametric and geometric continuity conditions. Moreover, applications of trigonometric splines in curves and surface, surface interpolation, shape-preserving interpolation, and 2D image reconstruction are examined to establish the broader relevance of the study.

Furthermore, the advantages and limitations associated with constructing trigonometric basis functions are critically assessed. These discussions help situate the proposed method within the existing literature and point out the weakness of current approaches.

2.2 Structure of Bezier Curve

The Bézier form is a widely used theoretical framework for representing parametric curves and surfaces in CAGD (Gerald Farin, 2002). In scholarly literature, the terms “Bezier” and “Bézier” are often used interchangeably; however, for consistency, this thesis adopts the spelling “Bezier.” The concept of the Bezier curve was independently developed in the late 1950s and early 1960s by Pierre Étienne Bezier at Renault and Paul de Casteljau at Citroën, both working in the automotive industry. Their objective was to create a mathematical formulation that would enable non-specialists to design complex shapes with ease. The Bezier curve’s simplicity and

applicability have made it extensively applied by researchers and designers across various fields.

The Bezier curve is formulated as presented below.

Definition 2.1. For any $t\epsilon[0,1]$, the general Bezier curve of degree n is defined by

$$B(t) = \sum_{i=0}^{n} P_i B_{i,n}(t) \tag{2.1}$$

where P_i are the control points or Bezier points for $i = 0,1,\dots,n$ and $B_{i,n}(t)$ are the Bernstein polynomials. Bernstein's polynomial basis is the primary function of the Bezier model with degree n specified by formula (2.2).

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{2.2}$$

where $\binom{n}{i}$ is the binomial coefficient. The configuration of the curve is dictated by the control points and the basis functions. The Bernstein polynomials hold the following properties:

2.2.1 Properties of Bernstein Polynomials

1. Non-negativity: Bernstein polynomials are non-negative, that is,

$$B_{i,n}(t) \geq 0 \qquad t \in [0,1].$$

2. Partition of unity: For $t \in [0,1]$,

$$\sum_{i=0}^{n} B_{i,n}(t) = 1.$$

3. Symmetry:

$$B_{n-i,n}(t) = B_{i,n}(1-t), \qquad i = 0,1,2,\dots,n.$$

4. Recursion:

$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),$$

with $B_{0,0}(t) = 1$ and $B_{i,n}(t) = 0$, for $i < 0$ or $i > n$.

2.2.2 Properties of Bezier Curve

The Bezier curve described in Equation (2.1) possesses the subsequent properties:

1. Terminal Properties

   i) Endpoint interpolation property: Typically, the first and last control points of the curve correspond to the endpoints, whereas the intermediate control points are not located on the curve.

$$B(0) = P_0 \text{ and } B(1) = P_n$$

   ii) Endpoint tangent property: The curve is tangent to the control polygon at the endpoints. That is,

$$B'(0) = n(P_1 - P_0) \text{ and } B'(1) = n(P_n - P_{n-1})$$

2. Convex Hull Property: The Bezier curve always lies within the convex hull of their control points, as the Bernstein polynomials are all non-negative.

$$\sum_{i=0}^{n} B_{i,n}(t) = 1\,,\ B_{i,n}(t) \geq 0, 0 < t < 1, \qquad i = 0, \dots n.$$

3. Symmetry:

$$B_{n-i,n}(t) = B_{i,n}(1-t), \qquad i = 0,1,2, \dots, n.$$

4. Variation Diminishing: The intersection count between a specific line and a planar Bezier curve must not exceed the intersection count between the line and a control polygon.

5. Invariance under affine transform: If ω is an affine transformation (translation, rotation, or scaling), then

$$\omega\left(\sum_{i=0}^{n} P_i B_{i,n}(t)\right) = \sum_{i=0}^{n} \omega(P_i) B_{i,n}(t).$$

2.3 Trigonometric Polynomial and Trigonometric Spline Curve

2.3.1 Background of Trigonometric Polynomials

Polynomial interpolation such as Bezier, B-spline, and Beta spline is an elementary concept in the CAGD area. It is simple and commonly used as a mathematical tool since it can be computed efficiently by computer systems and precisely represents many functions. Additionally, polynomials are simply differentiated, integrated, and stitched together to generate spline curves. Polynomial is limited to low degrees, but approximation methods can be applied to make the interpolation successful if a large number of data is used (Farin, 2002). In other circumstances, polynomials are used to describe the fundamental geometries of the models. In these cases, the polynomial must be in a rational form, which is more challenging to compute and may result in a greater degree of inaccuracy. According to (Bashir et al., 2013), polynomial splines lack the ability in creating curved arcs and conical shapes, which are the geometrical elements that are most prevalent in practically any behaviour system.

The required design cannot be completed in a single phase, even by applying the continuity conditions, due to the lengthy nature of generating curves, particularly when the complex curves and surfaces are needed to be formed by Bezier curves. To address the troublesome problem, generalized trigonometric basis functions with shape parameters are employed as an alternative to the use of multiple parameters for generating Bezier curves and surfaces (Maqsood et al., 2020).

2.3.2 Development of Trigonometric Spline Curves

Trigonometric splines emerged as an important extension of polynomial splines due to their flexibility and analytical properties. Scholars have shown a significant interest in the trigonometric spline by proposed quadratic, cubic, and higher order trigonometric spline schemes to model complex shapes. The proposed spline often demonstrating improved behaviour over classical polynomial splines. These schemes are particularly effective in maintaining smooth curvature transitions and can represent periodic behaviour in the data. Trigonometric functions are compatible with engineering and medical applications, such as wavelength and signal properties

Besides that, the trigonometric function is significant because it ensures the presence of smooth curves regardless of the parametric speeds are large, (Menasri et al., 2014). Furthermore, the trigonometric spline can overcome the obstacle of forming conic shapes and transcendental curves. In literary works, particular focus has been placed to their geometric behaviour and related applications. Trigonometric spline could carry out a variety of curve and surface modelling issues, as was further and completely discovered. Due to this, numerous trigonometric splines with particular properties appropriate for CAGD applications have been introduced. These are summarized in Table 1. 1. Since the research is specifically centred around the Bezier-like spline method, it is imperative to thoroughly examine the existing research that concerns the advancement of trigonometric Bezier-like spline

Table 1. 1
The Literature on Curves and Surfaces of Trigonometric Spline

No.	Author	Degree of curve					Continuity						No. of Shape Parameters in the Basis Functions	Surface
		1	2	3	4	5	C^1	C^2	C^3	G^1	G^2	G^3		
1.	(X. A. Han et al., 2009)	-	-	/	-	-	-	-	-	-	-	-	2	No
2.	(X. A. Han et al., 2010)	-	-	/	-	-	-	-	-	-	-	-	1	No
3.	(L. Yang et al., 2011)	-	-	-	/	-	-	-	/	-	/	-	2	No
4.	(Dube & Sharma, 2013)	-	-	-	/	-	/	-	-	-	-	-	1	No
5.	(Bashir et al., 2013)	-	/	-	-	-	/	/	-	/	/	-	2	No
6.	(Bashir & Ali, 2014)	-	-	/	-	-	/	/	-	/	/	-	2	No
7.	(Dube & Yadav, 2014)	-	-	-	-	/	-	-	-	-	-	-	1	No
8.	(Sharma Rani Durgawati Govt G College Mandla & Sharma Assistant Professor, 2016)	-	-	/	-	-	-	-	-	-	-	-	1	Yes
9.	(J. Li & Pakdemirli, 2016)	-	-	/	-	-	-	/	-	-	-	-	2	Yes

10.	(Hu et al., 2017)	-	-	/	-	-	-	-	-	-	/	-	1	Yes
11.	(Sarfraz et al., 2018)	-	/	-	-	-	-	/	-	-	-	-	1	No
12.	(Levent, 2018)	-	-	/	-	-	-	-	-	/	/	-	1	No
13.	(Zhu et al., 2018)	-	-	-	-	-	-	-	-	-	-	-	2	Yes
14.	(Bibi et al., 2019)	-	/	-	-	-	/	/	-	-	-	-	4	Yes
15.	(L. Peng & Zhu, 2019)	-	-	/	-	-	-	-	-	/	-	-	6	Yes
16.	Karim et al., 2020)	-	-	/	-	-	/	-	-	-	-	-	3	Yes
17.	(Maqsood et al., 2020)	-	/	-	-	-	/	/	/	/	-	-	2	No
18.	(Sun & Ji, 2020)	-	-	/	-	-	-	-	-	/	/	-	2	No
19.	(Ammad & Misro, 2020)	-	-	-	-	/	-	-	-	-	/	-	2	Yes
20.	(BiBi et al., 2020)	-	/	-	-	-	/	/	/	/	/	/	3	No
21.	(Hu et al., 2020)	/	-	-	-	-	-	-	-	/	/	-	3	Yes
22.	(Majeed et al., 2021)	-	-	/	-	-	-	-	/	-	-	/	1	No
23.	(Ammad et al., 2021)	-	-	/	-	-	-	-	-	/	/	-	2	Yes
24.	(Maqsood, Abbas, Miura, Majeed, BiBi, et al., 2021)	-	/	-	-	-	-	-	-	/	/	-	2	Yes

25.	(Maqsood, Abbas, Miura, Majeed, Hu, et al., 2021)	-	/	-	-	-	/	/	/	-	-	-	2	Yes
26.	(R. Sharma, 2022)	-	-	-	/	-	-	-	-	-	-	-	2	Yes
27.	(Ameer et al., 2022)	-	/	-	-	-	-	-	-	-	-	-	2	Yes
28.	(Lu & Guicang, 2022)	-	/	-	-	-	-	-	-	/	-	-	3	Yes
29.	(Mahzir & Misro, 2023)	-	-	-	-	/	-	/	-	-	-	-	2	No
30.	Proposed cubic trigonometric spline	-	-	/	-	-	/	/	-	/	/	-	1	Yes

Table 1.1 shows the list of trigonometric Bezier-like spline developed in the literature across the year. Among 29 reviewed papers, which excluded the proposed spline, most focus on cubic which is degree of three splines. For continuity, the preferred in scholars is geometric continuity with only degree of two for allowing the smooth join between adjacent curves and surfaces. Many researchers use only a few shape parameters, and 15 papers out of 29 papers extend their work to surface construction. Further details are discussed in the following section.

Over time, many kinds of trigonometric Bezier-like splines have been constructed by researchers as tabulated in Table 1. 1. The differences between the developments of each trigonometric Bezier-like spline are their level of smoothness, the overall amount of shape parameters, and the order of the continuity.

The existences of new development of Bezier basis functions are able to resolve the circumstances in constructing some curves. Besides, there are shortcomings in using the traditional Bezier method since they are polynomial based. Several literary works have been proposed to construct new basis functions with different degrees of spline and a number of the shape parameters while continuing to satisfy all the properties of Bezier-like spline. Those studies have a similar objective to improve the curve designing techniques in the geometric modelling sector.

In the earlier research by (Sharma, 2007), quartic trigonometric Bezier has been implemented with no structural differences with the classical quartic Bezier curve which can easily be adapted to CAD/CAM system. The formulation has been developed with one shape parameter. Then, the research has been extended by possessing two shape parameters in (Sharma, 2019). Both studies include illustrations of several quartic trigonometric Bezier curves and surface models in their paper.

Trigonometric Bezier curves can be created by following the same design principles as cubic Bezier curves. Two specific types of trigonometric Bezier curves and quasi-quartic trigonometric Bezier curves, have been proposed by (Han et al., 2009) and (Yang et al., 2011), respectively. These curves have the ability to represent the circular arcs, cylinders, cones, tori, parabolas, cardioids and elliptic arcs precisely. By extending to surface, the corresponding tensor product can generate quadratic surfaces, including cylindrical surfaces, spheres, ellipsoids, parabolic surfaces, and torus. The method that discovered in the paper improves the efficiency of developing surfaces and enhances the accuracy of depiction across a wide range.

Another finding by using new generalization of trigonometric Bezier curve that is trigonometric Bernstein-like (GBTB) basis functions with two shape parameters able to approach the convex hull better than the standard Bezier surfaces; besides, it is simpler to design complicated objects (Maqsood et al., 2021). These findings are also agreed by (Maqsood et al., 2021) whereby employing the same basis functions of GBTB, enable the creation of a more complex GBT-Bezier surfaces design. The geometric continuity requirements must be determined to construct and combine two

adjacent, developable GBT-Bezier surfaces. Modifying the shape parameters of a composite developable GBT-Bézier surface can be done without resetting the control planes. This allows for the adjustment of both the local and global shape of the surface while maintaining an even connection. The proposed GBT-Bezier showed greater performance than the traditional Bezier.

The use of higher-degree basis functions has also been examined in previous research. A quintic trigonometric Bezier curve with a single shape parameter has been established by (Dube & Yadav, 2014). The correlation between the cubic Bezier curve and quintic trigonometric Bezier curves has been demonstrated and the result showed that the quintic trigonometric Bezier curve generated nearer to the control polygon. In the study of (Ammad & Misro, 2020) the generated quintic trigonometric Bezier curve with shape parameters which have similar features to cubic trigonometric polynomials and NURBS (non-uniform rational B-splines), able to produce smooth Bezier surface, wept surface and swung surface.

Next, another trigonometric formulation generates to replace and enhance classical polynomial Bezier/B-spline approaches. (Han et al., 2010) provide a foundation by analysing shape behaviour for a simple trigonometric curve. In other research, the rational cubic trigonometric curve is generated by (Majeed & Qayyum, 2020) that incorporates both trigonometric and rational extensions. While, (Usman et al., 2020) propose trigonometric Bezier like curve, designed to overcome limitations of classical cubic Bezier representations, particularly their inability to represent conics. These three basis functions have been compared with the proposed cubic trigonometric spline in this study.

A review of (Usman et al., 2020), (Majeed & Qayyum, 2020) and (Han et al., 2010) reveals several fundamental similarities in their motivations, theoretical foundations, and methodological approaches. Despite focusing on different curve formulations, the three works share a common objective which are improving the flexibility, smoothness, and capability of trigonometric curve models for geometric design. The curves generated with one shape parameter which consider having the same capability in handling curves shape representations. Next, all papers also deal with continuity conditions up to the degree of five. This will ensuring smooth transitions between the connecting curves for CAGD applications. Each paper includes modelling demonstrations in proving the practical advantages in engineering. The papers establish

complete shape diagrams, providing an intuitive tool for designers to understand how control points and the shape parameter influence global shape behaviour.

(Han et al., 2010) provides rigorous shape analysis framework related to inflection points, cusps and loops) that will help the designers understand curve behaviour mathematically and visually. Meanwhile, (Usman et al., 2020) give a complete derivation of C^2 and G^2 continuity conditions between adjacent curve segments, providing a rigorous foundation for constructing long composite curves. In terms of continuity, (Majeed & Qayyum, 2020) achieves C^2 (uniform), C^3 and C^5 (non-uniform knots) which is the highest smoothness among the tree also the propose cubic trigonometric functions. This will give advantage for engineering applications that require high smoothness such as automotive and biomedical modelling. Compared with (Usman et al., 2020) and (Han et al., 2010) , this paper offers greater modelling flexibility and higher continuity, making it more suitable for complex surfaces of objects requiring smooth transitions. However, none of the papers fully extend their methods to surface modelling, despite the need for 2D surface representations in CAD.

2.3.3 Development of Trigonometric Spline Curves and Surfaces

In the industries sector, developable surfaces are needed in constructing and manufacturing goods, for example, in modelling the ship's compartments, including ship hulls, apparel ducts, and automobiles (Ferris, 1968). The first product in space was designed using developable surfaces that are flat in shape. Developable surfaces form the attachment of a collection of consecutive patches that emphasize the significant role of continuity conditions for adjacent patches. The critical issue that would be focused on is ensuring the smoothness joint for every adjacent patch along the common boundary.

Chu & Chen (2004) examined the development of surfaces with degree n by the joint of Bezier patches. For surface designing, the number of degrees of freedom (DOF) is evaluated, not depending on the degree of the surface. By randomly choosing the first boundary curve, $(2m + 3), (m + 4)$, and five DOF's remain in the developing surface containing m patches for the second boundary curve when the surface is G^0, G^1 and G^2. For C^1 and C^2 continuity, there are five and $(7 - 2m)$ DOF's. The paper indicates the practical methods in designing objects with limited DOFs that do not influence the restriction in the last procedure. (Maqsood et al., 2020) designed effective curves and

surfaces applications that meet the conditions required for parametric continuity (C^0, C^1, C^2, C^3) and geometric continuity (G^0, G^1, G^2, G^3). The presence of shape parameters is beneficial in altering the shape of the curve and surface. The advantage of this technique is the flexibility and ability in developing curves without increasing the degree and mathematical complication that give timely, practical submissions.

De Casteljau algorithm is applied to generate four developable surface boundary curves using continuity (Li et al., 2021). The construction of the surfaces is given with the constraints of the control points that must be fulfilled. Hu et al., (2020) introduced a new type of Bezier curve entitled shape-adjustable generalized Bezier (SG-Bezier) with incorporates locally and globally shape parameters. The selection values of the suitable parameter for the shape modification of the composite surfaces are investigated, satisfying the conditions for G^1 and G^2 continuity. Next, developable cubic trigonometric Bezier (GDCT-Bezier) surfaces were introduced by (Ammad et al., 2021) who demonstrate the ability to design geodesic interpolating surfaces. The sufficient conditions for generating intricate structures in engineering and architectural design are determined, including the requirements for G^1, Farin-Boehm G^2 and G^2 Beta continuity. The conditions also have been derived by Bibi et al., (2021) with the addition of G^3 continuity conditions for the modification of GHT-Bezier surfaces. The new invention of G^3 continuity allows for modelling complex outcomes that were previously unexplored in the fabrication of developable GHT-Bezier surfaces.

Furthermore, Ammad & Misro (2020) came out with biquintic Bezier surfaces with different types of adjustable shape surfaces, such as general Bezier, swept, and swung surfaces. The paper demonstrates the impact of shape parameters on surface design using a mean curvature nephogram. The boundary curves were retained unchanged, while the inside of the surface form was modified in accordance with the variations in shape parameter values. Yang et al., (2012) proposed quasi-biquartic trigonometric Bezier surfaces and joining many patches of C^1 and C^4 continuous to construct right angle pipe, bowl-shaped surfaces and a tread surface.

Hu et al., (2017) constructed a kind of generalized Bezier-like surface that performed better than existing Bezier-like surfaces in terms of shape adjustment and approximation ability. By achieving G^2 continuity conditions, complex curves can be developed easily yet offer additional flexibility. Sun & Ji (2020) proposed a class of cubic T-Bezier curves that incorporate two shape parameters for the purpose of

designing and simulating model kitchen products as well as other goods in curved shapes. The method possesses the ability to create items with cutting edge designs in a shorter timeframe and yield superior results in comparison to the conventional design approach. The benefits of the suggested approach include shape modification, simplicity of shape optimization design realization, and the ability to adapt or expand to the parametric creation of different goods.

2.3.4 Shape parameter

Shape parameters are widely used in CAGD as it helps the designers control the shape of the curves and surfaces intuitively and easily. The fundamental idea of the existing shape parameter, together with classical basis functions, comes from the constraint in adjusting the shape of the curves. The Bezier curve, or surface generated by Bernstein basis functions, can no longer be changed once the control points are provided. For example, Bezier curves are generated using specifically defined control points that are locality fixed depending on the control polygon. The curve's shape is capable of being altered by manipulating the control points (Han & Zhu, 2012).

Most researchers agree that to satisfy this need, the basis functions have to be set up with shape parameters to allow the curve to move as flexibly as the designers want it to be. Without repositioning control points, the curve's shape and the surface can vary (Yan, 2016; Usman, 2020; Ammad, 2020). In Majeed et al., (2021), the curve is adjusted using two ways; using control points and shape parameters to discover the effect of the curve. By using the shape parameter, the value varies. It is observed that when the value of the shape parameter greater, the curve moves nearer the control polygon and vice versa.

Razali et al., (2021) stated that spline has been enhanced by performing shape parameters as the manipulative in generating curve. By including shape parameters, the versatility and efficiency of Bezier trigonometric surfaces have increased (Ammad et al., 2021). In addition, Bibi et al., (2020) mentioned that the physical and geometrical figures are unaffected by shape parameters.

The outcome of the generated spline influenced by the number of shape parameters. Using only one shape parameter gives the limited capability to manipulate the curve but it is adequate and easier enough to adjust the shape of the curve as needed. In (Han, 2004), the cubic trigonometric polynomial curves with a shape parameter were

performed. Modifying the shape parameters will have a global impact on the curve. However, by incorporating a changeable shape parameter into the trigonometric spline curves, it becomes possible to control the shape of the curve on a local level within each segment (Choubey & Ojha, 2008). Han (2005) introduced local shape parameters to enable the generation of curves flexibly. Han (2002) and Han (2004) presented trigonometric polynomial curves with global shape parameters.

Numerous research has been carried out using only one shape parameter in their new proposed basis functions. Examples include by Majeed et al., (2021), Usman et al., (2020),Levent (2018), Hu et al., (2017), Sarfraz et al., (2018), and Sharma et al., (2016). Most researchers agreed that by using limited shape parameters, it is easier to decide the value of shape parameters in constructing the shape of curves and surfaces besides requiring simpler computation. Moreover, in joining two or more curves and surfaces, the presence of a scalar factor in the continuity conditions is very helpful in shape generation.

The availability of numerous shape parameters allows for modifying the curve's shape without repositioning control points (Zain et al., 2021). As the number of shape parameters increases, the curve becomes easier to customize and adjustable. However, this necessitates the manipulation of numbers of values to be varied. Maqsood et al., (2021) discovered that their proposed basis functions with two shape parameters showed simpler and extra succinctness compared to other Bezier curves and surfaces techniques that rely on multiple shape parameters.

Quadratic trigonometric spline curves have been developed by Wu et al., (2007) with multiple shape parameters that enable the adjustment of the curve globally and locally. Peng & Zhu (2019) developed a trigonometric Bezier patch using six practical shape parameters. All shape parameters need to be restricted to adjust the shape of the patch. The border of the curves is unaffected by the six shape factors, according to the boundary property.

2.3.5 Continuity

CAGD is highly reliant on mathematical representations of things that are derived from parametric functions. A piecewise parametric spline function consists of many segments, each of which is a parametric function. A vital element of this purpose is the manner in which the segments are interconnected. Continuity constraints are the

equations that define this joining. The majority of industrial design applications demand smoothness and shape authenticity. The continuity constraints are often achieved by selecting the specified level of spline smoothness. The application determines the smoothness order to be used. The theoretical concept of parametric and geometric continuity is presented in Chapter 4 while the related literature is discussed in this chapter.

Most of the works of literature apply convenient continuity conditions in constructing desired curves and surfaces. For example, Bibi et al., (2019) constructed parametric continuity conditions for the GHT-Bezier curve. Various kinds of symmetrical shapes, including symmetrical curves, and rotational surfaces have been designed to demonstrate the effectiveness of the modelling by employing the GHT Bezier curves and the continuity requirements and symmetric formulas. Karim et al., (2020) also used parametric continuity conditions, C^1 for the construction of a new cubic Bezier-like triangular basis function. C^2 and G^2 continuity is needed for two contiguous TC Bezier-like curves to resolve the issue that composite functions cannot always be built by a single curve (Usman et al., 2020). Next, Dube & Sharma (2013) discussed the composition of two quartic trigonometric Bezier curves using C^1 continuity. Sufficient conditions are generated in achieving continuity. Piecewise quadratic trigonometric polynomial curves which have C^2 continuity was discovered by Han (2005).

Geometrical continuity is preferable to the B-spline because of the shape parameter at each vertex that will provide more degrees of freedom for modifying the shape of the curve (Diao et al., 2010). (Yahya, et al., 2006) favoured a geometric continuity curve for font fitting because loops easily develop to maintain equal tangent at joints for parametric continuity. These loops will later perceive themselves as surface artefacts and require additional fairings to make them smooth. Bercovier & Volpin (1999) also describes a technique for generating composite surfaces from a group of quadrilateral patches using the smoothing constraints C^1 and G^1. Hu et al., (2020) derived conditions for G^1 and G^2 continuity between two adjacent SG-Bezier and modelling examples demonstrated that the SG-Bezier was more effective than existing Bezier surfaces in terms of form adjustability and approximation ability.

CAD/CAM systems are used to build a lot of sophisticated engineering surfaces, yet it might be challenging to create a complex surface from a single surface. According

to Ammad & Misro (2020), geometric continuity which is G^1 continuity is needed by the designers for generating complex surfaces. Nevertheless, for the higher requirement in smoothness, G^2 continuity must be achieved where the surfaces share a single tangent plane at every point along their shared boundary. Ammad et al., (2021) utilized G^1, Farin-Boehm G^2, and G^2 Beta continuity to assure smoothness between the novel approach of two neighbouring GDCT-Bezier. The outcomes of the research implied that the GDCT-Bezier with continuity provides a high level of flexibility in designing evolving surfaces. Additionally, it is simple and convenient to incorporate into the design process. This claim is reinforced by Hu et al., (2017), who have shown that the proposed G^2 smooth continuity leads to the formation of piecewise generalized Bezier-like surfaces. In addition, the methods are practical, simple for handling, and provides additional variable in generating complicated surfaces in engineering design.

2.4 Application of Trigonometric Curves and Surfaces

CAGD deals with constructing parametric curves and surfaces that accurately pass through a given set of data points which is called interpolation. The interpolation techniques can be categorized into two distinct groups; global and local methods. By using global approaches, curves or surfaces are produced utilizing all the data points simultaneously. The issue arises when modifying any data points will affect the whole kind of surface. This situation is different from interpolation using local techniques where the interpolation is done locally. Any changes in the data point will only affect a part of the interpolated surface and only values from neighbouring dispersed point set points will be taken into account.

The fundamental requirement is that the interpolating curve or surface retains the data's form properties (Kvasov, 2000). Natural phenomenon that occurs which can be distinguished into real-shaped data is always stated in three conditions namely, positive, monotone, and convex. This study will specifically focus on the positive real data. The graphic representation of positive curves and surfaces is essential for seeing objects which cannot have negative values, as the positivity of curves and surfaces is derived from regular data. The positive form is mostly used to acquire shape-preserving properties. Some of the real positive data examples are resistance offered by an electric circuit, population statistics, solvability of solute in solvent, and stability of radioactive substances and chemical reactions.

2.4.1 Shape-Preserving Interpolation

Shape preservation and interpolation are vital processes for graphical data display in scientific visualization. One of the shape characteristics scholars most often mentioned in this subject is positivity. In many physical situation applications, entities can only be meaningful when a positive physical amount depiction is used. In order to interpolate or approximate this data while keeping the same distinctive properties, a suitably smooth univariate or bivariate function should be established.

Numerous studies have been conducted on generating positivity-preserving interpolants. The method started with the development of constraint schemes applied to the shape parameters within the basis functions. Bashir & Ali (2014) devised a method for interpolating positive data using a piecewise C^1 rational trigonometric curve. This method ensures that the positivity of the data is maintained. The scheme is evaluated on 2D positive data sets, where the results of cubic trigonometric Bezier curves are displayed as smooth, appealing positive curves in graphics. The values of two shape parameters are constrained by the scheme, whereas the selection of the other two shape parameters is based on trial-and-error basis.

Hussain et al., (2010) studied the issue of C^1 rational cubic Bezier interpolation with shape preservation and generated a function with only one free parameter to maintain the shape of the data but no adjustable parameters to allow users for adjusting the curve. Studies by Karim (2014) offered the GC^1 cubic interpolant with two shape parameters, one is constrained while the other shape parameters are free and used rational cubic Bezier splines to demonstrate the positive shape preserving. Later, using a cubic numerator and a quadratic denominator, Karim & Pang (2016) generated a new C^2 rational cubic Bezier spline interpolant. The newly suggested technique, which included the preservation of positive data, featured three parameters, where one parameter served as a sufficient condition, and the other two allowed users to alter the final curve form freely.

The development of new partially blended rational bi-quartic spline interpolation by Karim et al., (2021) with C^1 continuity managed to ensure a positive surface on the full specified interval. The proposed positivity scheme is simpler in computation compared to other scheme since the required numbers of mathematical derivations has been decreased. Besides, the scheme provides shorter CPU times (in seconds) when constructing a positive interpolating surface.

Next, to interpolate positive scattered data, Hussain & Hussain (2011) suggested a rational cubic Bernstein-Bézier technique. Similar to Piah et al., (2006), which derive the necessary condition on the Bezier ordinates to preserve the shape of positive data, Hussain & Hussain (2011) provides a positivity condition that is derived in the same approach. The primary distinction in their studies is that the approach developed by Hussain & Hussain (2011) include a weight function in their formulation. This weight function allows the user to change the shape of the data if the Bezier ordinates in any triangular patch fail to meet the specified lower bounds.

The topic of positive data visualization was also covered by Brodlie et al., (2005), who also created an approach employing interpolated scattered data by applying the dimension-modified quadratic Shepard technique. The proposed approach ensured that the quadratic basis function remained positive and retained its properties automatically. The procedure maintained the lowest and upper boundaries of every interpolant between the two provided functions. A piecewise rational cubic Bezier function with two shape parameters was constructed by Hussain (2006) work in order to maintain the shape of positive data. The proposal does not offer free parameters to change the positive curve's shape. Despite the scheme's lack of flexibility, the C^1 continuous curve was visible and appeared smooth.

By applying compactly supported radial basis functions, Wu et al., (2010) have studied positivity-preserving approximation and interpolation for curves and surfaces (CSRBFs). The optimization problem must be resolved to attain positivity. The inherited shape of positive data was then visualized by Sarfraz et al., (2010) using a rational bi-cubic partially blended function that was extended from a rational cubic Bezier function. A rational quadratic trigonometric function with three shape parameters was designed by Dube & Rana (2014) to preserve positivity. The function was able to maintain positivity, although some of the interpolants made from several data sets were not smooth.

A cubic form piecewise rational function with four shape parameters in each interval was proposed by Sarfraz et al., (2012). In order to maintain the shape of convex, monotone, and positive data, two of these shape parameters are constrained; the other two, nevertheless, are controlled by adjusting positive, monotone, and convex curves to produce a more aesthetically pleasing curve.

The non-rational trigonometric quadratic spline was employed by Sarfraz et al., (2015) to keep the data's positivity, monotonicity, and convexity. In the trigonometric

development, three shape parameters were employed. Two of these parameters were utilized to retain the shape of the data, while the remaining shape parameters was dedicated to adjusting the shape of the preserving curve. Han (2015) introduced a piecewise rational interpolant with a quartic numerator and a quadratic denominator. To satisfy the interpolant's shape-preserving features, particularly positivity and monotonicity, a shape parameter was chosen for each subinterval.

To maintain the inherited positive shape, Abbas et al., (2013) have been using the cubic/quadratic piecewise rational cubic Bezier function. The proposed curve technique by the authors is suitable for both data with and without derivatives. The derivative value was solved using LU decomposition. In producing a smooth and positive curve, a piecewise rational cubic interpolation with three shape parameters has been established by Faris et al., (2016). The data's shape positivity is dependent on one of these shape parameters, while the other two are left up to the curve's designer to adjust as necessary. Modifications of the bounded shape parameter values give impact on the curves used for data visualization's smoothness and positivity.

(Karim, 2018) developed a new C^1 rational cubic spline with three shape parameters. The automatic selection of one parameter is established through some mathematical reasoning in order to employ the proposed rational cubic spline for limited data that are above any arbitrary straight line and positivity maintaining interpolation. This approach provides two new free parameters to each sub-interval, which are highly helpful for the user to alter the curve's shape while still preserving the data's shape.

Automated methods that adhere to the shape-preserving constraint have been established by Mahzir & Misro (2023) on one of the shape parameters, meanwhile on the other shape parameter, the smoothness of the end interpolations is guaranteed. The proposed positivity-preserving techniques, which performed well when applied to larger datasets, are utilized to interpolate the actual data, which is the daily COVID-19-related death data. The devised scheme produces accurate outcomes with only minor differences from the actual data when compared to actual statistics. According to these results, missing data points in time-series analysis can be interpolated using the suggested methods.

2.4.2 Image Reconstruction and Data Fitting

Curve fitting has been carried out using a variety of approaches. Trigonometric form, genetic algorithm, simulated annealing, particle swarm optimization (PSO), least square method (LSM), gradient-descent method, neutral network, and Bayesian are some of the terms used to describe these techniques. The chapter includes reviews a range of previous studies that have reconstructed of 2D images.

Mostly researchers chose to reconstruct different kind of fonts for example Thai, Chinese characters and Arabic letters. Each works has different challenging in terms of continuity chosen, constructing algorithm and organizing flow to reconstruct curves similar like the real images. By using G^1 rational Bezier cubic curves Yahya et al., (2006) fitted Arabic outline characteristics, 'qaf' dan 'dal'. In the fitting process, choosing continuity is essential since an Arabic font is cursive with curves and cusps. G^1 curves are used at the corner points instead of C^1 as loops may be presented to maintain the equal tangent at the joining point. In other cases, reconstructing Arabic fonts as studied by Zainudin et al., (2021), by using Butterfly Optimization Algorithm (BOA), the results showed that BOA does not do well in addressing curve fitting and reconstructing and needs a hybrid approach in enhancing the techniques. Thai consonant contour has been presented by Dejdumrong & Tongtar (2007) using cubic Bezier curve fitting. By choosing Jasmine UPC font, G^1 continuity has been achieved where the features are not complicated and only have the head, closed-loop, and twist specifications. In the situation when the curve joining does not meet G^1 continuity condition, the curve subdivisions should be made more. Li et al., (2019), Razali et al., (2021) and Zhang et al., (2007) redesigned Chinese calligraphy using different kind of spline and approaches algorithm. The comparisons between the splines and algorithms have been made in their research and each of them has its behaviour in modifying the curvature of curves. The findings displayed the effective method and algorithms in constructing a collection of Chinese character.

Throughout the research, in reconstructing 2D images, some phases should be followed for the curve fitting process. First and foremost, the initial step in the curve fitting process is the image digitization and boundary extraction process. These are the steps in getting the outline features of the image by extracting the contour points and removing the interior points. The important point in this part is to get a clear and smooth image by eliminating the noise that is not used in this process. The spatial filter was

introduced by Yahya (2009) to smooth the digital image before the threshold for converting the image from a digital to a bi-level image. The process continued using Mathematical morphology to obtain the boundary image. Raseli & Ali (2012) included the steps for editing the image using Adobe Photoshop and Adobe Illustrator to eliminate noise production. These will result in a smooth and clear image yet help produce good outcomes because the extraction process becomes more manageable. The image's boundary, which is obtained through the boundary extraction method, is particularly helpful for other pre-processing techniques including data extraction and corner detection. The boundary points of the image are applied to carry out the corner detection process.

Next, corner detection is an essential part of the curve fitting process, which will help the designers to set up the control points. Corners will determine the division points of the boundaries for the curve segments. Therefore, for cubic trigonometric spline, this is likely Bezier spline, with corners practiced as the endpoints of the curve segments and the end of control points. By using contour, the character's corner points calculated by Sarfraz & Khan (2002). Both interactive and automatic processes are used to determine the corner points. The contoured outline is divided at the corner points for optimal curve fitting.

Finding the control points for the Bezier curve appears not to be a big deal since the properties interpolate at the beginning at the last control points as referring to (Hadi, 2014). In the situation of a cubic Bezier curve, it has four control points P^0, P^1, P^2 and P^3. The curve will interpolate at the endpoints, which are at the control points P^0 and P^3 and approaching the other two intermediate points P^1 and P^2. For evaluating the control points of P^1 and P^2, one preferred technique is the LSM as stated in (Rusdi & Yahya, 2015) and (Yahya et al., 2006). Parameterization techniques are applied to estimate the value of t before getting the values of control points P^1 and P^2.

Those points will generate the best-fitted curve for specified values of t. It was considered an achievement when the gap between the generated fitted curve with the contour point of the original image was smaller. The results' validity could be improved by calculating the distance using Hausdorff distance and Euclidean distance. The fitness of the curve is assessed by Yang et al., (2001) using the total square distance between each contour segment point and the Bezier curve correspondingly. The control points are modified based on the intersection points of the perpendicular line for both the

Bezier curves and the contour segments to regulate the cost. If the cost of curve fitting is sufficiently low, the shape is created with great precision.

Amat et al., (2019) applied Sum Square Error (SSE) to compute the error generated by the algorithms used in their research. Two algorithms are applied which are the Sine Cosine Algorithm (SCA) and the Whale Optimization Algorithm (WOA) to maximize the solution of the curve fitting issue utilizing the cubic Bezier curve. The findings indicates that WOA's performance surpasses that of SCA due to its minimal error. On the other hand, the corresponding spline function parameter is optimized via differential evolution (DE) by Hasan et al., (2018) who compared the effectiveness of two different types of curves, cubic Bezier and cubic Ball. In order to discover the optimum curve to fit the data, DE minimized the SSE. The outcomes indicate the reconstruction generated a visually identical image; however, the cubic Ball gives a remarkable outcome compared to cubic Bezier which produced a lesser error for all four examples presented.

Several scholars have worked on 3D reconstruction in recent years. In order to interpolate irregular data sets, specifically rainfall data sets, Karim et al., (2024) set up scattered data interpolation method utilizing the cubic Bezier-like triangular patches. The outcomes of the proposed scheme provided a smooth surface in addition to speedy performance when fixing with a particular parameter value. The study employed the positivity-preserving criterion, increasing the smoothness form C^1 to C^2 parametric continuity, since rainfall cannot be negative.

A B-spline face surface can be covered with distinct structure and physical features by combining the low rank and B-spline, which takes advantage of the 3D structure form images taken “in the wild” (Peng et al., 2016). By employing a novel geometric method, Zhu et al., (2018) able to reduce the requirement for data storage, resulting in an efficient 3D developing process. Tensor product forms of rational and non-rational W-spline surfaces are used to construct the three-dimensional graphic. The 3D reconstruction result is generated by utilising the new control matrix and modifying the parameters based on the photographs.

A new approach had discovered by Ai et al., (2010) for 3D reconstruction of liver using auto-segmentation and the advanced Marching cubes algorithm. The approach resulting a fast speed in practice typically less than one second besides simply extended to apply in any other tissue by adjusting suitable values of the parameters. Saini & Kumar (2014) worked on Non-Uniform Rational B-Spline (NURBS) model to

reconstruct of 3D shapes. By solving a system of linear equation, control points in two image planes are acquired besides the weight vectors are also discovered by resolving the quadratic programming problem. This resulting the essential outcomes of surface reconstruction and the errors has been compared between proposed and triangulation-based approach.

Heo & Chae (2004) had presented an image segmentation algorithm based on B-spline (Han & Zhu, 2016) curve fitting to produce smooth tooth regions form such CT slices. The algorithm provided the fitting procedure with an accurate initial tooth boundary. A highly effective threshold approach is proposed in this study, which utilizes both the level and shape information from the previous slice for the initial boundary generation. Additionally, an effective B-spline fitting based on genetic algorithm are used in this research. This works contributes to the industries by stimulating orthodontic surgery and treatment. The proposed method accurately and smoothly identifies the shape of each individual tooth, resulting in a 3D model.

By using identical control polygon, (Halim et al., 2024) conducted a comparison of four distinct established trigonometric triangular spline curves and surfaces. A higher number of surface points indicates a smoother surface, according to the data. Compared to the Han & Zhu (2016) and Peng & Zhu (2019) , the surface points produce by Zhu & Han (2015) and Wang & Zhang (2018) significantly closer to the control polygon.

2.5 Summary and Gap Analysis

In this chapter, it is clear from the literature on generating new spline construction, various approaches have been developed, each with its benefits and drawbacks. In designing curves and surfaces, the aim is to place a strong emphasis on generating aesthetically refined curves or surfaces in addition to improve the data interpolation. Most researchers preferred to use a Bezier spline that is extended into a rational form to simulate and interpolate curves and surfaces. Besides that, trigonometric polynomials had also been extensively used since they provide advantages that allow for the representation of basic trigonometric curves without the necessity of the rational form. All the proposed spline provides an alternative to generate curves spline. The differences in terms of the degrees of the spline, the number of shape parameters, and the continuity conditions used in connecting curves and

surfaces are showing the uniqueness of each developed spline. The important part in generating new basis functions is that it satisfies the properties needed for constructing Bezier basis functions. New proposed splines are developed possessing comparable features to the traditional Bezier.

Most of the researchers agreed that by using trigonometric Bezier-like basis functions together with the shape parameter gives extra freedom in generating complex curves and surfaces compared to the traditional ones. According to Table 1. 1, the majority of the paper employs two shape parameters, followed by one shape parameter, three shape parameters, and multiple shape parameters. An increased number of shape parameters provides greater flexibility and customization for the curve, but it also requires significant variances in the parameter values. Reducing the number of shape parameters allows for quicker and simpler computation, but it also restricts the ability to adjust the shape of the curve.

The trigonometric surface is a method that is judged ideal for producing surfaces in accordance with the market's current trend and aesthetic desire based on the analysis of prior literature and studies. As earlier research tended to concentrate on generating new basis functions, it neglected to extend the curve equation into the surface equation. Trigonometric surface equation is not widely discussed by researchers. Therefore, by satisfying several continuity conditions, numerous contiguous surfaces will be joined throughout this work.

For reconstruction of image, different researchers used various approaches and algorithms in curve fitting process. The results reveal that the generated spline allows to regenerate the image intended. However, most of the papers did not examine the accuracy of the recreated image with respect to the original image. The efficiency of the proposed spline should be known to achieve a good quality of curves in terms of smoothness, formation shapes, processing time, and accuracy.

To sum up, numerous studies have advanced the development of trigonometric curve and spline formulations but there is no works offer a systematic comparison between the generated trigonometric curves. Specifically, no study that evaluates the important criteria such as computational efficiency, numerical stability, modelling accuracy or error metrics. Consequently, the practical advantages of trigonometric representations remain largely qualitative rather than quantitatively validated. This lack of evaluation can lead to inaccurate curve and surface reconstruction in application.

So, a systematic comparative analysis has been considered in this research. The presented trigonometric cubic spline will be compared with other cubic trigonometric spline in literature in terms of modelling accuracy.

CHAPTER 3
TRIGONOMETRIC CURVES WITH SHAPE CONTROL

3.1 Introduction

The principal purpose of CAGD is to create, modify, analyse, and optimize any design needed by the designer. The flexibility, continuity, and smoothness of the curve or spline produced are considered to ensure the objective is fulfilled. Scholars have researched constructing a basis for generating object shape models using various methods such as non-uniform rational B-splines, B-spline, Bezier, and Hermite. The standard practice uses non-uniform rational B-splines; however, accuracy has some shortcomings since it does not interpolate all object points to preserve continuity. Furthermore, this standard method consumes a longer time and large storage in representing object shapes. Bezier is one of the methods that the researcher widely uses, given its capability in designing free-form curves and surfaces according to the designer's desire. Nevertheless, using the polynomial bases of Bezier fails to perform well in higher order. This circumstance motivated the use of the trigonometric function in the basis function. In addition, the presence of shape parameters and scalar factors that act as the controlling factors is vital for the designers to achieve desired outcomes while prioritizing smoothness.

This chapter presents the development of quadratic and cubic trigonometric splines incorporating a shape parameter. The construction of new basis functions for generating smooth trigonometric splines requires the fulfilment of specific properties, ensuring their effectiveness. Essential spline properties, such as non-negativity, partition of unity, and endpoint interpolation are examined to demonstrate that the proposed basis functions satisfy the criteria needed for reliable curve construction.

The characteristics of these splines exhibit strong similarities to those of conventional Bezier splines. A key aspect of the formulation is the inclusion of shape parameters, which provide local shape control and serve as an essential mechanism in curve design. By adjusting the values of these parameters, the desired curve can be obtained without altering the positions of the control points. Examples are provided to demonstrate the influence of the shape parameters to the behaviour of the trigonometric

spline. This property enables designers to achieve precise and intuitive control over the final shape.

This study findings show the strengths of the proposed quadratic and cubic trigonometric basis functions, such as flexibility in contour shaping, as well as potential limitations or constraints. The outcomes presented in this section serve as a transition to the following chapters, where the proposed trigonometric spline are applied to interpolation tasks, surface generation, and image reconstruction problems.

3.2 Quadratic Trigonometric Spline

First, the implementation involves utilizing quadratic trigonometric basis functions are implemented that have only one shape parameter, β. The pattern and style for developing the spline are inspired by (Bashir et al., 2013) where both focus on trigonometric Bezier curve constructions with shape parameters that used quadratic trigonometric blending functions defined over the interval $[0, \theta]$, where $\theta \in \frac{\pi}{2}$. But there are differ significantly in the mathematical formulation, degree and the number of shape parameters. The (Bashir et al., 2013) scheme is based on rational quadratic trigonometric Bezier curve that uses two shape parameters to allow local shape manipulation meanwhile the proposed scheme used a single shape parameter.

All the necessary properties and characteristics required for the construction of trigonometric basis functions are thoroughly outlined and proven. Additionally, graphical illustrations highlight the influence of the shape parameter β, on the spline, with endpoint interpolation also taken into account.

3.2.1 Quadratic Trigonometric Basis Functions

Definition 3.1. For $u \in [0,1]$, quadratic trigonometric basis functions with one shape parameter β, where $\beta \geq 0$ are defined as:

$$\begin{aligned} H_0(u) &= \frac{\beta}{\beta+1} - \frac{2\beta}{\beta+1}\sin\frac{\pi}{2}u + \frac{\beta}{\beta+1}\sin^2\frac{\pi}{2}u; \\ H_1(u) &= 1 - H_0(u) - H_2(u); \\ H_2(u) &= \frac{\beta}{\beta+1} - \frac{2\beta}{\beta+1}\cos\frac{\pi}{2}u + \frac{\beta}{\beta+1}\cos^2\frac{\pi}{2}u. \end{aligned} \tag{3. 1}$$

when $\beta = 0$, the curve shows a straight line.

Figure 3.1 shows plots of quadratic trigonometric basis functions $H_i(u)(i = 0,1,2)$ in (3. 1) using variation shape parameter values, β.

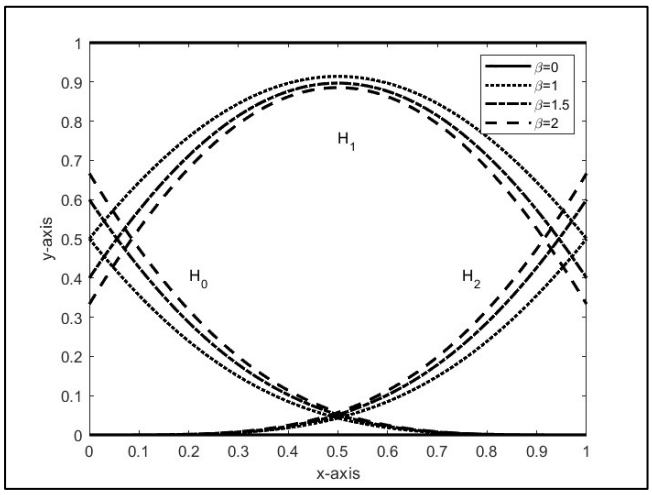


Figure 3.1 Quadratic Trigonometric Basis Functions with Varying Shape Parameter, β

3.2.2 Properties of Quadratic Trigonometric Basis Functions

The basis functions (3.1) above have all the following properties:

i) Non-negativity:

$$H_i(u) \geq 0, i = 0,1,2$$

ii) Partition of unity:

$$\sum_{i=0}^{2} H_i(u) = 1, u \in [0,1]$$

iii) Monotonicity: For the given value of the parameter, β, $H_0(u)$ is monotonically decreasing and $H_2(u)$ is monotonically increasing.

iv) Symmetry:

$$H_i(u;\beta) = H_{2-i}\left(\frac{\pi}{2} - u;\beta\right), i = 0,1,2$$

Proof.

i) For $u \in [0,1]$, when $\beta \geq 0$,

$H_0(u)$:

$$\frac{\beta}{\beta+1}\left(1 - \sin\left(\frac{\pi}{2}u\right)\right)^2 \geq 0$$

$H_2(u)$:

$$\frac{\beta}{\beta+1}\left(1 - \cos\left(\frac{\pi u}{2}\right)\right)^2 \geq 0$$

Thus, it follows that:

$$H_1\left(\frac{\pi}{2}u\right) \geq 0$$

So, $H_i\left(\frac{\pi u}{2}\right) \geq 0, i = 0,1,2.$

ii) Partition of unity is evident.

For all u, β

$$H_0\left(\frac{\pi}{2}u\right) + \left(1 - H_0\left(\frac{\pi}{2}u\right) - H_2\left(\frac{\pi}{2}u\right)\right) + H_2\left(\frac{\pi}{2}u\right) = 1$$

iii) For $u_0, u_1 \in [0,1]$, such that $u_0 \leq u_1, H_0(u_0) \geq H_2(u_1)$ which shows that $H_0(u)$ is monotonically decreasing.

For $u_0 \leq u_1$, $H_2(u_0) \leq H_2(u_1)$ which shows that $H_2(u)$ is monotonically increasing.

iv) For $i = 2$,

$$H_2\left(\frac{\pi u}{2};\beta\right) = \frac{\beta}{\beta+1} - \frac{2\beta}{\beta+1}\cdot\cos\frac{\pi}{2}u + \frac{\beta}{\beta+1}\cdot\cos^2\frac{\pi}{2}u$$
$$= \frac{\beta}{\beta+1} - \frac{2\beta}{\beta+1}\cdot\sin\left(\frac{\pi}{2} - \frac{\pi}{2}u\right) + \frac{\beta}{\beta+1}\cdot\sin^2\left(\frac{\pi}{2} - \frac{\pi}{2}u\right)$$
$$= H_0\left(\frac{\pi}{2} - \frac{\pi}{2}u;\beta\right)$$

The quadratic trigonometric spline used in this study is defined using a set of quadratic trigonometric basis functions with a single shape parameter. The spline is expressed as

$$F(u) = \sum_{i=0}^{2} H_i(u)V_i, \qquad u\epsilon[0,1] \tag{3.2}$$

where $H_i(u)$ denote the quadratic trigonometric basis functions and V_i represent the associated control points. This formulation generate curve that does not interpolate the endpoints. This behaviour is expected, because quadratic trigonometric type schemes generally lack enough degree of freedom to enforce endpoint interpolation unless additional constraints are imposed. The quadratic spline only approximated the data rather than interpolating it. Further examples are shown in the next section.

3.2.3 Shape Control of Quadratic Trigonometric Spline

The impact of the shape parameter β, on the basis functions of the quadratic trigonometric spline is examined by varying its value. By using control points $P_i = (i = 0,1,2)$, where $P_0 = (0,0)$, $P_1 = (1.5,1.5)$ and $P_2 = (3.0,0)$, a quadratic trigonometric spline curve is formed. The quadratic trigonometric curve is given by $F(u) = \sum_{i=0}^{2} H_i(u)P_i$. The behaviour of the curve is restrained by the shape parameter of the quadratic trigonometric spline.

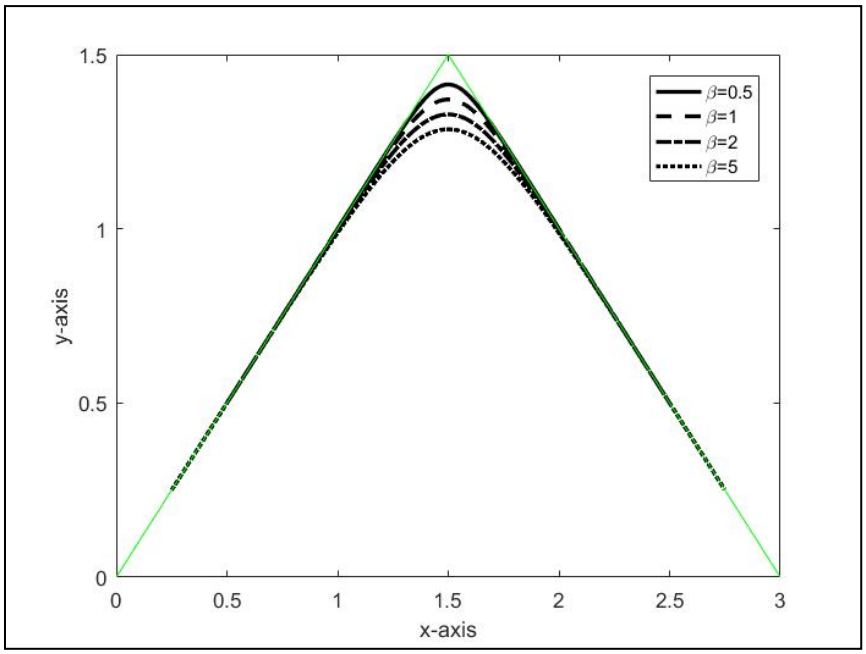


Figure 3. 2 The Impact of Different Values of β on the Quadratic Trigonometric Spline

Figure 3.2 shows the quadratic trigonometric spline as the values of shape parameter β increase, the distance of the quadratic trigonometric curves from the control polygon is also increased. When the shape parameter $\beta = 0$, $H_0(u)$ and $H_2(u)$ vanish, while $H_1(u)$ becomes identically equal to one for all parameter values u. A pleasing and smooth curve is produced by choosing various values of the shape parameter. Bashir et al., (2013) stated that it is beneficial when a curve retains the control polygon's shape.

3.2.4 Endpoint Condition

The importance of curve interpolation is to identify the endpoints of the interpolating. The control points $P_i(i = 0,1,2)$ define the starting and ending points.

For control points $P_i(i = 0,1,2)$, we define the curve as in equation (3.2). So, equation gives,

$$F(u) = H_0P_0 + H_1P_1 + H_2P_2$$

For the first control point where $u = 0$,

$$F(0) = P_0 \tag{3. 3}$$
$$H_0P_0 + H_1P_1 + H_2P_2 = P_0$$

After computations, we obtain

$$\frac{1}{\beta + 1}(P_1 - P_0) = 0 \tag{3.4}$$
$$\lim_{\beta \to \infty} \left(\frac{1}{\beta + 1}(P_1 - P_0 \right) = 0$$

While at the last point where $u = \frac{\pi}{2}'$

$$F\left(\frac{\pi}{2}\right) = P_2 \tag{3. 5}$$
$$H_0P_0 + H_1P_1 + H_2P_2 = P_2$$

So, we get

$$\frac{1}{\beta + 1}(P_1 - P_2) = 0 \tag{3. 6}$$
$$\lim_{\beta \to \infty} \left(\frac{1}{\beta + 1}(P_1 - P_2 \right) = 0$$

From the limit, there are no exact values of β that makes the curve touch both endpoints. The curve only interpolates nearer to the control points as the value of β is increased. Otherwise, phantom and double control points are the techniques that allow the interpolating curve to develop at the endpoint.

Particular examples are illustrated in Figure 3.3:

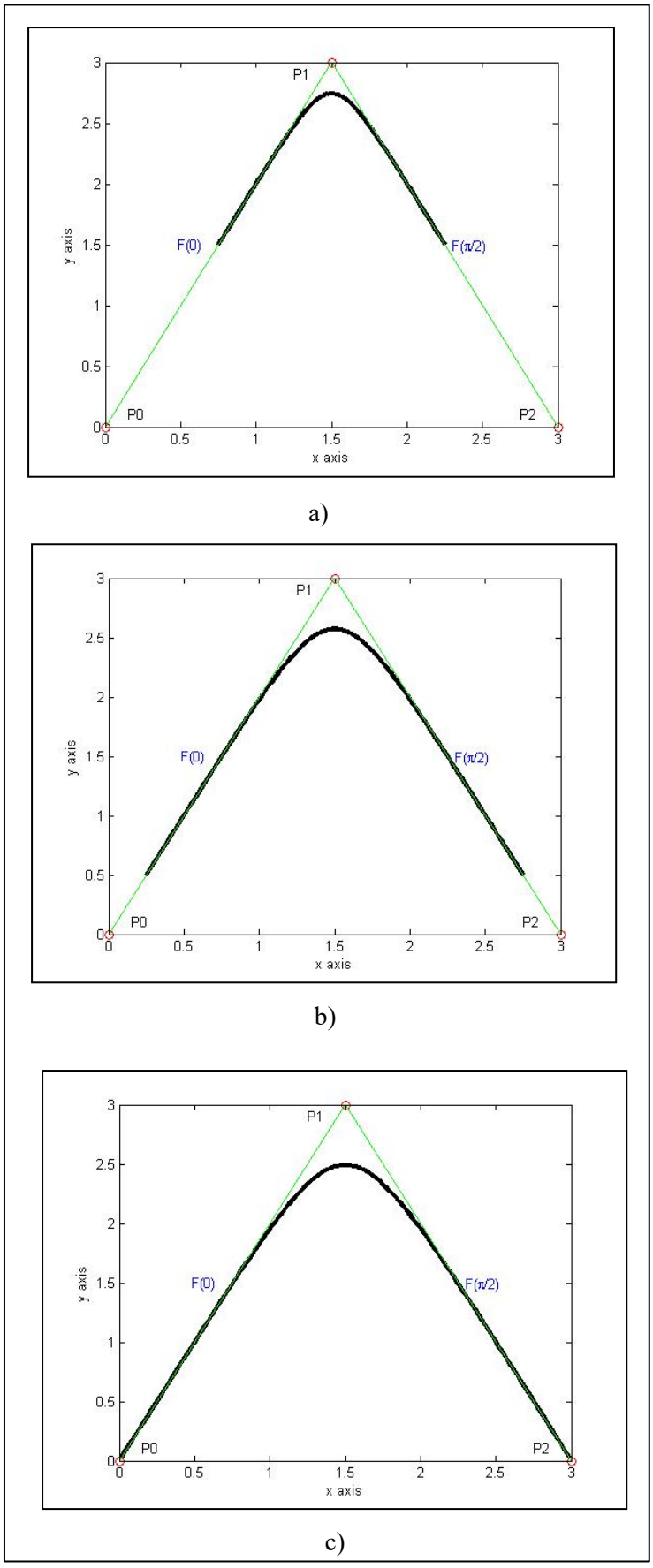


Figure 3. 3 Curve End Conditions by Varying Values of β. (a) $\beta = 1$, (b) $\beta = 5$, (c) $\beta = 100$

Figure 3.3 illustrated the effect of varying the shape parameter β on the end-point behaviour of the quadratic trigonometric spline. The curves are generated using control point P_0, P_1 and P_2. As expected for this type of spline, the curve starts at the first control point P_0 and ends at the last control point P_2. The curve will not interpolate the middle point P_1 instead the point only influences the shape of the curve.

In Figure 3.3(a), for $\beta = 1$, the resulting curve lies inside the control polygon and the curve fitted at the middle of the edge P_0P_1 and P_1P_2 for the value of shape parameter $\beta = 1$. Figures 3.3(b) and Figure 3.3(c) display the curve for larger shape parameter values, specifically, $\beta = 5$ and $\beta = 10$. As the values of β increases, the curves are generated towards the endpoints. Although the curve increasingly aligned with the edges of the control polygon but it does not interpolate the end control points P_0 and P_2. This shows that the quadratic trigonometric spline does not satisfy the endpoint interpolation requirement. The basis functions of the quadratic trigonometric spline do not possess the Bezier-like properties required for guaranteed endpoint interpolation. Consequently, although the shape parameter β provides some control over the curve bends, it cannot force the curve pass exactly through the endpoint control points.

3.2.5 Representation of Piecewise Quadratic Trigonometric Spline

However, by using the piecewise quadratic trigonometric spline, many kinds of shape can be represented. As the spline formulation yields open curves and does not impose endpoint interpolation, it enables a high degree of geometric flexibility. Below in Figure 3.4 and Figure 3.5, the heart-shape and star-shaped open curves are represented.

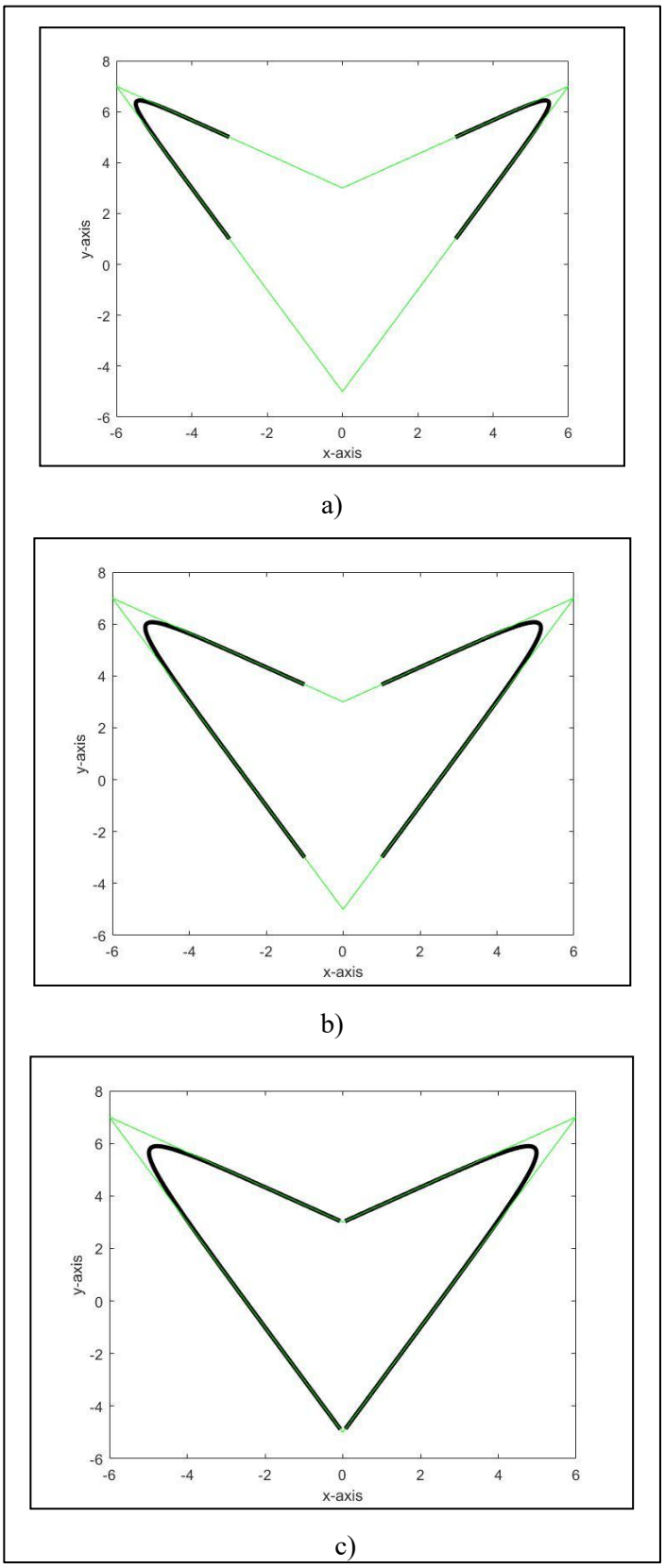


Figure 3.4 Heart Design by Quadratic Trigonometric Spline by Varying Values of Shape Parameters. (a) $\beta = 1$, (b) $\beta = 5$, (c) $\beta = 100$

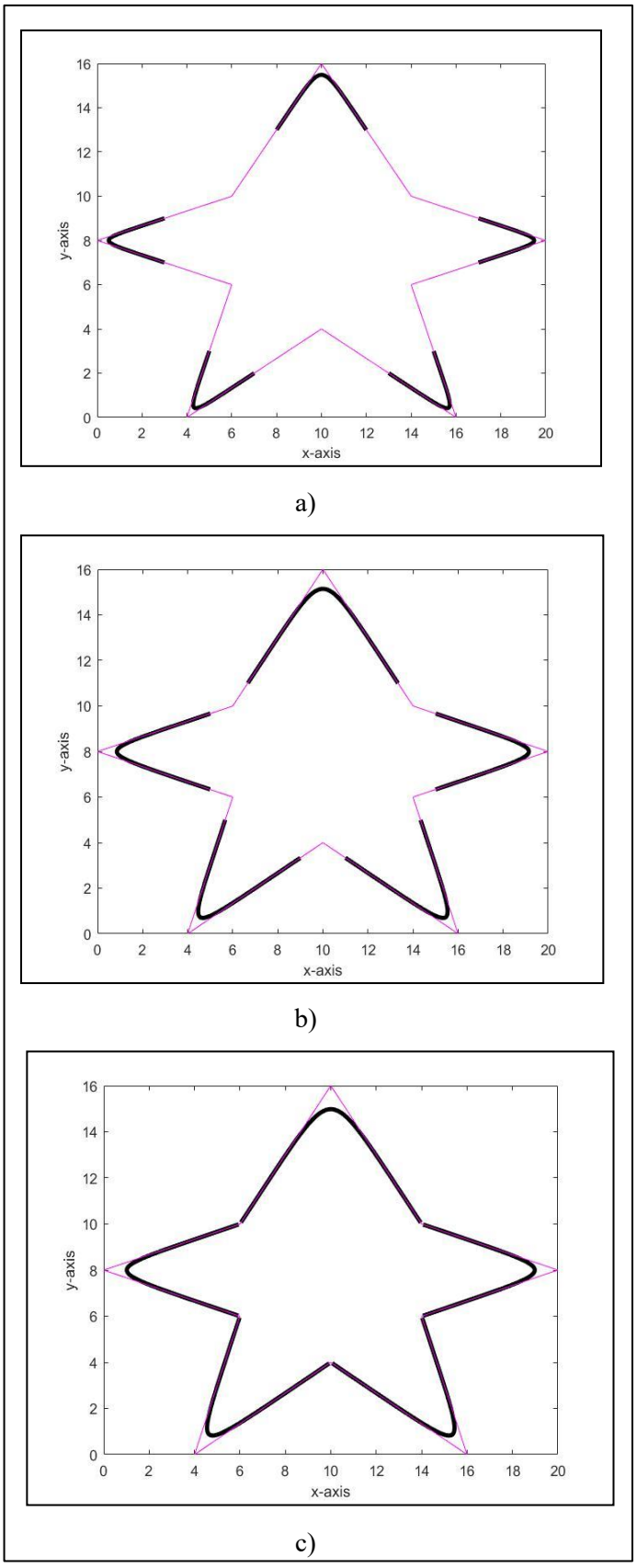


Figure 3.5 Star Design by Quadratic Trigonometric Spline by Varying Values of Shape Parameters. (a) $\beta = 1$, (b) $\beta = 5$, (c) $\beta = 100$

Figure 3.4 and 3.5 represent the effects on quadratic trigonometric spline by using the various shape parameter values on the heart and star design. As the shape parameter β increase, the curve is generated towards both endpoints.

3.3 Cubic Trigonometric Spline

A new cubic trigonometric spline has been generated with four basis functions. The spline acquires some of the properties needed to be presented in this section. The presence of shape parameters in the splines helps the designers in achieving the desired shape without changing the control points.

3.3.1 Cubic Trigonometric Basis Functions

First, the definition and properties of cubic trigonometric spline basis functions are presented.

Definition 3.2. Cubic Trigonometric basis functions with a shape parameter, m where $m \in [-2,1]$ are defined for $u \in [0,1]$:

$$\begin{aligned} b_0(u) &= \left(1-\sin\frac{\pi}{2}u\right)^2\left(1-m\sin\frac{\pi}{2}u\right); \\ b_1(u) &= m\sin\frac{\pi}{2}u\left(1-\sin\frac{\pi}{2}u\right)^2 - 2\sin\frac{\pi}{2}u\left(\sin\frac{\pi}{2}u-1\right); \\ b_2(u) &= m\cos\frac{\pi}{2}u\left(1-\cos\frac{\pi}{2}u\right)^2 - 2\cos\frac{\pi}{2}u\left(\cos\frac{\pi}{2}u-1\right); \\ b_3(u) &= \left(1-\cos\tfrac{\pi}{2}u\right)^2\left(1-m\cos\tfrac{\pi}{2}u\right); \end{aligned} \tag{3. 7}$$

When,

$$s = \sin\frac{\pi}{2}u, \qquad c = \cos\frac{\pi}{2}u;$$

So, equation (3.7) also can be written as

$$\begin{aligned} b_0(u) &= (1-\mathrm{s})^2(1-ms); \\ b_1(u) &= ms\,(1-\mathrm{s})^2 - 2\,\mathrm{s}(\mathrm{s}-1); \\ b_2(u) &= mc\,(1-\mathrm{c})^2 - 2\,\mathrm{c}(\mathrm{c}-1); \\ b_3(u) &= (1-\mathrm{c})^2(1-mc); \end{aligned}$$

Figure 3.6 below depicts a graphical representation of the basis functions $b_{i,3}(u)(i = 0,1,2,3)$ in (3. 7) of cubic trigonometric spline that is built with various values of shape parameter m.

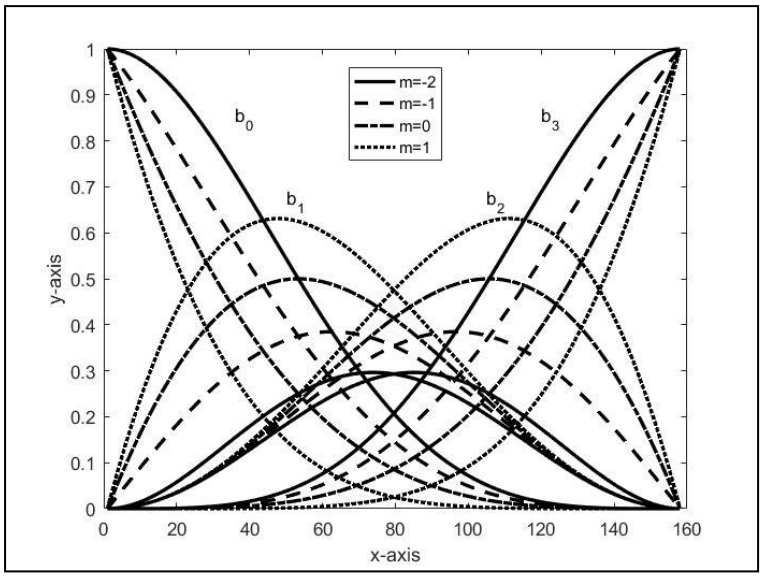


Figure 3.6 The Cubic Trigonometric Basis Functions with Varying Values of Shape Parameters of m

3.3.2 Properties of Cubic Trigonometric Spline

Cubic trigonometric basis functions have the following properties:

i) Non-negativity:

$$b_i(u) \geq 0, \text{ for } i = 0,1,2,3.$$

ii) Partition of unity:

$$\sum_{i=0}^{3} b_i(u) = 1$$

iii) Monotonicity: For the given value of the parameter, m, $b_0(u)$ is monotonically decreasing and $b_2(u)$ is monotonically increasing.

iv) Symmetry:

$$b_i(u;m) = b_{2-i}\left(\frac{\pi}{2} - u;m\right), i = 0,1,2,3$$

Proof.

i) For $u \in [0,1]$, and $m \in [-2,1]$, then

$$\left(1-\sin\frac{\pi}{2}u\right)^2 \geq 0, \qquad 1-m\sin\frac{\pi}{2}u \geq 0,$$

$$\left(1-\cos\frac{\pi}{2}u\right)^2 \geq 0, \qquad 1-m\cos\frac{\pi}{2}u \geq 0,$$

$$m \leq 1$$

$\therefore b_0(u)$ and $b_3(u)$ are always non-negative.

$$m\sin\frac{\pi}{2}u\left(1-\sin\frac{\pi}{2}u\right)^2 - 2\sin\frac{\pi}{2}u\left(\sin\frac{\pi}{2}u-1\right) \geq 0$$

$$m \geq \frac{-2}{\left(1-\sin\frac{\pi}{2}u\right)}$$

$$m \geq -2$$

$$m\cos\frac{\pi}{2}u\left(1-\cos\frac{\pi}{2}u\right)^2 - 2\cos\frac{\pi}{2}u\left(\cos\frac{\pi}{2}u-1\right) \geq 0$$

$$m \geq \frac{-2}{\left(1-\cos\frac{\pi}{2}u\right)}$$

$$m \geq -2$$

$\therefore b_1(u)$ and $b_2(u)$ are always non-negative. So, the interval values of m are $[-2,1]$.

ii) For all $u \in [0,1], m \in [-2,1]$,

$$b_0(u)+b_1(u) = 1-\sin^2\frac{\pi}{2}u$$

$$b_2(u)+b_3(u) = 1-\cos^2\frac{\pi}{2}u$$

$$b_0(u)+b_1(u)+b_2(u)+b_3(u) = 2-\left(\sin^2\frac{\pi}{2}u+\cos^2\frac{\pi}{2}u\right)$$

$$= 2-1$$

$$= 1$$

$\therefore \sum_{i=0}^{3} b_i(u) = 1$ is proved.

iii) For $u_0, u\epsilon[0,1]$, such that $u_0 \leq u_1, b_0(u_0) \geq b_0(u_1)$
This demonstrates that $b_0(u)$ is monotonically decreasing.
For $u_0, u_1\epsilon[0,1]$, such that $u_0 \leq u_1, b_3(u_0) \geq b_3(u_1)$
This demonstrates that $b_3(u)$ is monotonically increasing.

iv) For $i = 3$,

$$b_3(u,m) = \left(1 - \cos\frac{\pi}{2}u\right)^2\left(1 - m\cos\frac{\pi}{2}u\ \right)$$
$$= \left(1 - \sin\left(1 - \frac{\pi}{2}u\right)\right)^2\left(1 - m\sin\left(1 - \frac{\pi}{2}u\right)\right)$$
$$= b_0\left(1 - \frac{\pi}{2}\mathrm{u};\mathrm{m}\right)$$

3.3.3 Shape Control of Cubic Trigonometric Spline

A cubic trigonometric curve is a type of parametric curve defined using trigonometric basis functions of cubic order, instead of polynomial basis functions like in Bezier or B-spline curves. The curve is a piecewise or single curve segment expressed as a linear combination of control points and cubic trigonometric blending (basis) functions.

General form:

$$\sum_{i=0}^{3} b_i(u)P_i$$

P_i : control points
$b_i(u)$: cubic trigonometric basis functions that are constructed to satisfy certain properties like non-negativity, partition of unity, monotonicity, symmetry and endpoint interpolation.

The cubic trigonometric curve is $Q(u)$ where,

$$Q(u) = \sum_{i=0}^{3} b_i(u) P_i$$

The resulting curves of the cubic trigonometric spline curve are displayed in Figure 3.7. Considering control points as stated as $P_i, (i = 0,1,2,3)$, that is $P_0 = (0,0), P_1 = (7,3)$, $P_2 = (10.5,8)$, $P_3 = (12,16)$, Figure 3.7(a) displays the effects on cubic trigonometric spline when the values of the shape parameter, m, are varied. In Figure 3.7(b), the control points are $P_0 = (0,0), P_1 = (1.5,1.5), P_2 = (3,0), P_3 = (4.5,1.5)$ also shows the behaviour of the curves when modifying the value of m. Both figures show the curve interpolates approaches and aligns closely with the control polygon as the value of shape parameter m expands.

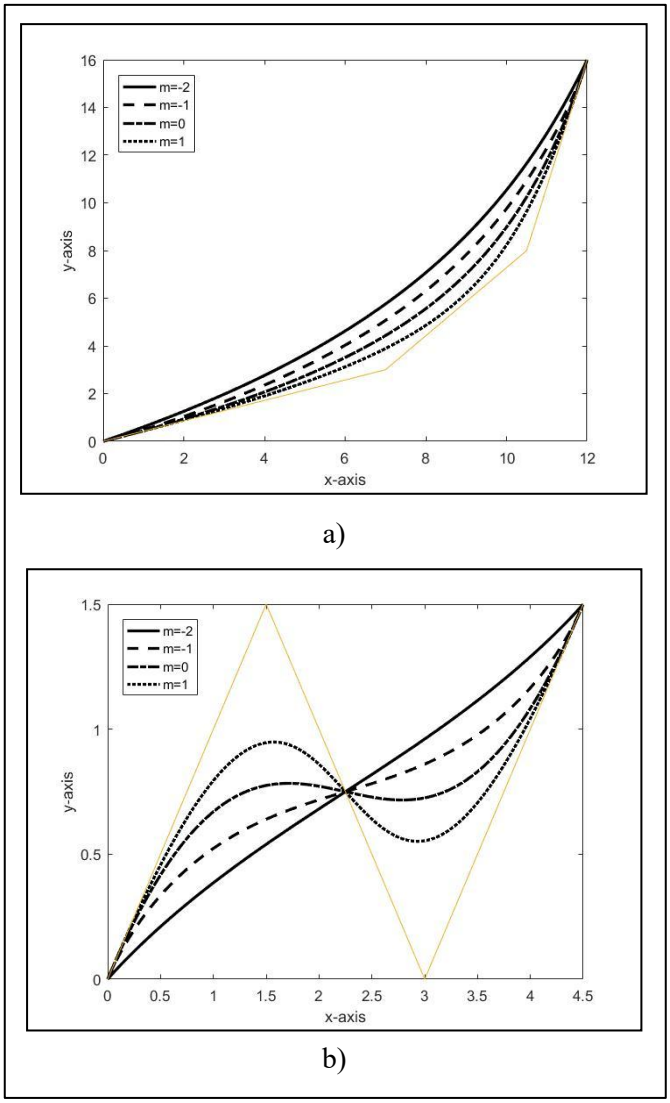


Figure 3.7 The Effects of Shape Parameter on the Shape of Cubic Trigonometric Basis Functions

3.3.4 Representation of Cubic Trigonometric Spline

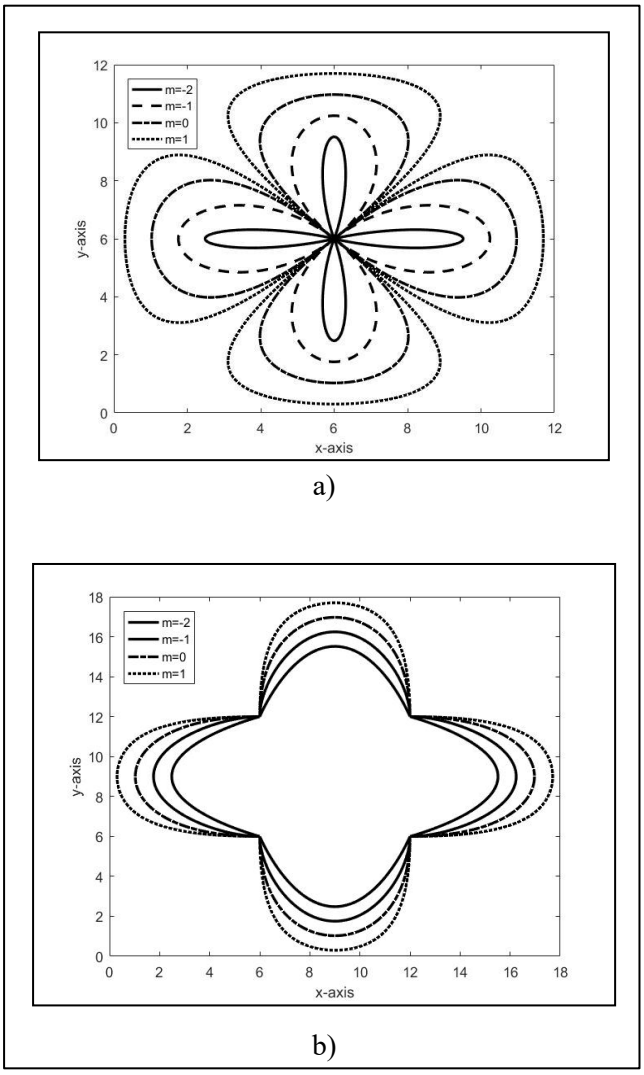


Figure 3.8 Flower Design by Cubic Trigonometric Spline by Varying Values of Shape Parameters (a) Closed curves (b) Opened Curves

Figure 3.8 shows the impact of the shape parameter, m, in representing our desired shape of curves without repositioning the control points. The shape parameter, m incorporated in the structure cubic trigonometric curve, helps to control its shape locally. Local modification can be made without influencing the rest of the other curve. Figures 3.8 (a) and 3.8 (b) developed closed and opened curves by cubic trigonometric spline.

3.4 Summary

New quadratic and cubic trigonometric basis functions that incorporate a single shape parameter have been developed and described in this chapter. A crucial step in developing the basis functions is ensuring that they satisfy and preserve the fundamental properties of Bezier curves. For quadratic trigonometric curves, the resulting curves uphold nearly all principal geometric properties such as, convex hull containment, geometric invariance, intuitive shape control controllability. However, it does not satisfy the endpoint-interpolation property which is does not pass through the first and endpoints. For cubic trigonometric curves, the curves generated meet all the geometric properties including the endpoint interpolation. Thus, only cubic trigonometric curves are prioritised for subsequent research as it fulfils the Bezier-like behaviour.

An analysis is conducted on the behaviour of the quadratic and cubic trigonometric curves, each of which incorporates with a single shape parameter. Smooth spline curves are developed using both quadratic and cubic trigonometric basis functions. The presence of shape parameter allows precise control over the shape characteristic, which is highly advantageous in the context of design. This feature is particularly useful in geometric design, as even small adjustments to the parameter can result in observable changes in the behaviour of the curve. It is beneficial in geometric design applications, where smoothness, flexibility, and, local control is crucial. Consequently, a diverse range of shapes can be constructed depending on the intended in use. This work can be extended to the development of continuity condition required for generating smoothly connecting curves and surfaces patches, which are discussed in Chapter 4 and Chapter 5.

CHAPTER 4
TRIGONOMETRIC CURVES WITH PARAMETRIC AND GEOMETRIC CONTINUITY CONDITIONS

4.1 Introduction

In modelling complex and complicated curves, using a higher-order polynomial is not an excellent course of action since it requires complex mathematical computations and takes a longer processing time. However, it is also impossible to design by using only a single curve. Every curve part is called a segment; the surfaces are known as patches. Constructing more than one segment and patch may be challenging but crucial in designing complex shapes and objects. In this case, designers find ways to represent a smooth joint of curves and surfaces. This section highlights the necessity of smooth joints between segments and establishes continuity as a fundamental requirement for achieving smooth, visually coherent connections.

Smoothness is an essential prerequisite for producing curves in this manner. The term for conducting this joining is called continuity. This section highlights the necessity of smooth joints between segments and establishes continuity as a fundamental requirement for achieving smooth, visually coherent connections. Continuity also plays a role in enabling flexible modification of curves and surfaces while keeping the control points unchanged. Additionally, it ensures that smoothness is preserved across junctions, avoiding the creation of sharp corners at each connecting point. The challenging part is to preserve the continuity and fulfilling specific shape constraints. The first and second curves and surfaces will not be smoothly merged, affecting the curvature continuity if the required continuity requirements between the two merging curves and surfaces are not satisfied.

The degree of smoothness varies depending on the utilisation. In architectural design, it is only necessary for the curve segments to have positional continuity. If it relates to designing mechanical parts, it is essential to have a high level of smoothness, either first or second order as stated by Barsky and DeRose (1989). By simply matching the curves only at their connecting points is insufficient. A seamless connection can only be achieved if additional requirements are applied at those specific points.

Two types of continuity are prominent in modelling curves in CAGD: parametric and geometric. The parametric and geometric continuity conditions are the common measuring standards to ensure smooth and continuous connecting of the piecewise parametric curves to the next (Zain et al., 2021). The continuity conditions enable the curves and surfaces to be connected at two fixed points, the endpoints. Continuity generates a smooth curve without gaps, holes, and breaks between points in the curve (Chambers et al., 1996). Continuity requirements must be met for transition curves to join two non-adjacent cubic curves (Levent, 2018).

The symbol used to represent nth order parametric and geometric continuity is C^n and G^n, respectively. For parametric continuity, the direction and magnitude for the first n derivatives must be equal at the common point. Meanwhile, in geometric continuity, only the directions for the two curves must be similar, while the magnitude can be contrasted for the first n derivatives. The level of continuity at the junctions decides and impacts the smoothness of the connecting curves.

This chapter presents the basic concepts of parametric continuity and geometric continuity. The basic concepts are utilized in the development of cubic trigonometric curves that are interrelated by parametric and geometric continuity requirements. The development of piecewise trigonometric curves that satisfy the first and second order continuity requirements is presented. Mathematical derivations and conditions used to ensure smooth connections are detailed.

Lastly, this section provides practical applications on the constructed curves. Examples are included to validate the approach, demonstrating its usefulness in design tasks and real-world datasets.

4.1.1 Parametric Continuity

The most common way to represent how smoothly parameter values change with distance along a curve in computer graphics is via parametric continuity**.** Parametric continuity expressed as C^n in mathematical notation, refers to the level smoothness exhibited by of both the curve and its parameterization up to the n^{th} order. The following examples presented two curves $P(u), u \in [u_0, u_1]$ and $Q(u), u \in [u_1, u_2]$ with order parametric continuity, C^n.

Definition 4.1. A curve $P(u)$ is said to be C^n continuous if the derivatives up to order $n, i.e., \frac{d^k P}{du^k}$ for $k = 0,1, \ldots, n,$ are continuous values throughout the curve.

There are three kinds of parametric continuities most commonly used:

i) Zero-order parametric continuity, C^0. The curve segments intersect at one endpoint. C^0 continuous occurs when two curve segments are simply connected as shown in Figure 4.1.

$$P(1) = Q(0)$$

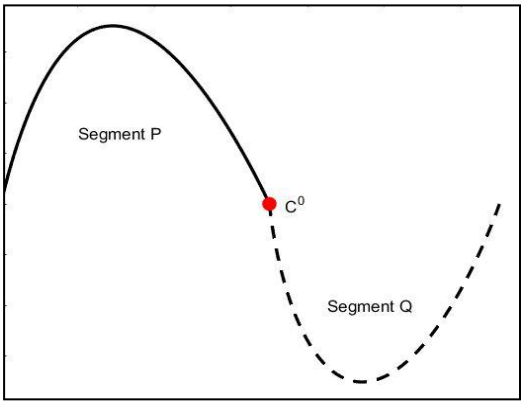


Figure 4.1 Piecewise Curve for C^0 Continuity

ii) First-order parametric continuity, C^1. A curve is considered to have C^1 continuity if it is C^0 continuous as well as when if the first-order derivative of successive segments at the common points are identical. The two curve segments possess identical tangent vectors, characterised by equivalent magnitude and direction at the intersection point. In other words, the geometric slopes at the joint must be similar. The example of C^1 continuity is illustrated in Figure 4.2.

$$P'(1) = Q'(0)$$

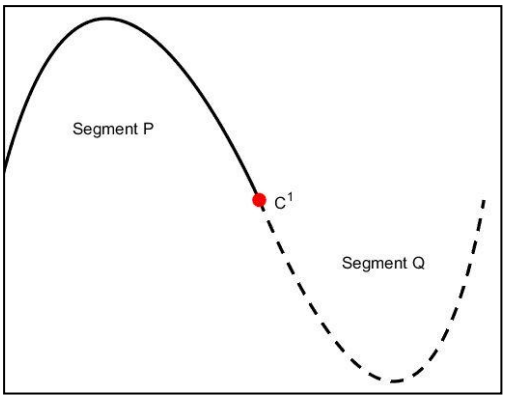


Figure 4.2 Piecewise Curve for C^1 Continuity

iii) Second-order parametric continuity, C^2. For C^2 continuity, the curve requires C^0 and C^1 continuities and the second-order derivative of two curve segments are the same at the intersection. C^2 continuity means continuity of the acceleration vectors. The curvatures for both curves have the same magnitude and direction as displayed in example in Figure 4.3.

$$P^{"}(1) = Q^{"}(0)$$

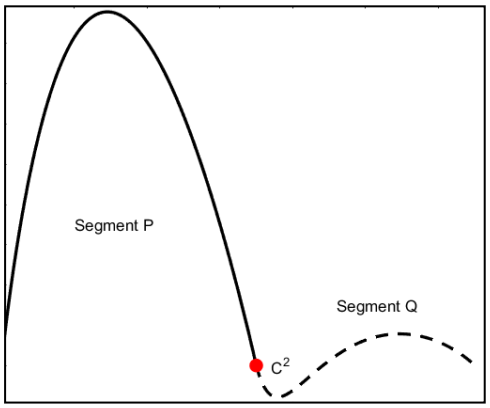


Figure 4.3 Piecewise Curve for C^2 continuity

4.1.2 Geometric Continuity

Geometric continuity is written in G^n and identified as visual continuity. Geometric continuity serves as an alternative technique for connecting two consecutive curve segments. The parametric derivatives of the two segments must be proportional at the identical boundary, rather than be equal with one another.

Definition 4.2 A curve $P(u)$ is said to be G^n continuous if the derivatives up to order $n, i.e., \frac{d^k P}{du^k}$ for $k = 0,1, \ldots, n$.

There are three types of geometric continuities follow standard formulations found in the literature (Farin, 2002; Piegl & Tiller, 1997; Hoschek & Lasser,1993):

i) Zero-order geometric continuity, G^0. G^0 continuity is equivalent to C^0 parametric continuity. The two curve segments must use the same coordinate position at the boundary point. G^0 continuity is known as positional continuity.

$$P(1) = Q(0)$$

ii) First-order geometric continuity, G^1. In Figure 4.4 shows that the examples of two consecutive curves intersect where the parametric first derivatives are directly proportional at the point. The direction of the tangent vector, yet regardless of its magnitude will be both equivalent and continuous at the joint. G^1 continuity is called tangent and directional continuity.

$$P'(1) = \lambda\, Q'(0), \qquad \lambda > 0$$

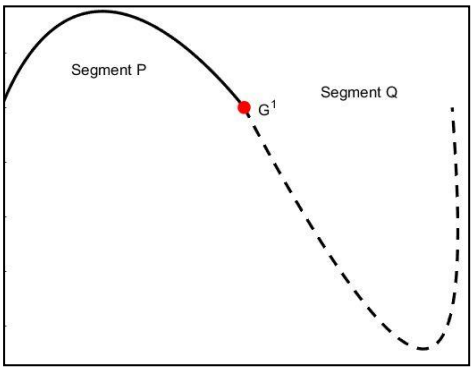


Figure 4.4 Piecewise Curve for G^1 Continuity

iii) Second-order geometric continuity, G^2. This continuity is also known as curvature continuity. The parametric first and second derivatives are proportional at their boundary. The curvatures of two adjacent curves are matched at the joining position.

$$P''(1) = \mu\, Q''(0) + \gamma\, Q'(0), \quad \mu > 0$$

The parameter μ governs the scaling of curvature, while γ is a reparameterization adjustment term.

A summary of the continuity properties of curve segments is listed below:

i) C^0 and G^0 is called positional or point continuity. The two curves are merely intersected and joined at their respective endpoints.

ii) C^1 and G^1 known as tangent or gradient continuity. It has the same velocity vector where the geometric slopes at the joint must be the same. It represents the concept of maintaining a smooth transition between two curves by ensuring that their tangents align. The first derivatives of two adjoining curve functions are equal. In order for two curves to be C^1 and G^1, they have to meet the requirements to be C^0 and G^0. Tangency continuity refers to the condition where the tangent line at the starting point of one curve is parallel to the tangent line of another curve, and both curves share the same origin point. Furthermore, C^1necessitates that the tangents possess equal lengths.

iii) C^2 and G^2 is known as curvature continuity. It has the same velocity and acceleration vector. The gradients also at the centre of curvature must be the same. Both the first and the second derivatives of two adjoining curve functions are equal.

In comparison with parametric continuity, geometric continuity gives more freedom in parameterization, while still ensuring the geometric smoothness of the final curve (Barsky & DeRose, 1989). The presence of scalar factor for geometric continuity allows for performing and designing complex shapes in less confining. Therefore, it is more competent compared to parametric continuity.

4.2 Cubic Trigonometric Spline Interpolation with C^1, C^2, G^1 and G^2.

This section focuses on the construction of a piecewise cubic trigonometric spline generated by connecting two segments of the cubic trigonometric curve. To ensure smooth transition between the segments, both first and second levels are imposed. The continuity analysis naturally leads to two conditions which are, Condition 1, that applies the same values of shape parameter, m for the connected segments, while

Condition 2 governs the case when the two segments are defined using different values of m. These conditions ensure that the resulting spline maintains the desired level of smoothness at the boundary.

4.2.1 Cubic Trigonometric Spline Interpolation with C^1 Continuity Conditions

Theorem 4.1: Let $Q_1(u)$ and $Q_2(u)$ be two cubic trigonometric curves. C^1 continuity connects the two curves at the connected point.

The definition of cubic trigonometric curve with a single shape parameter m is:

$$Q(u) = b_0P_{i-1} + b_1P_i + b_2P_{i+1} + b_3P_{i+2}, u \in [0,1]. \tag{4. 1}$$

Proof. Two curves can be expressed as:

$$\begin{aligned} Q_1(u) &= b_0(u)P_0 + b_1(u)P_1 + b_2(u)P_2 + b_3(u)P_3 \\ Q_2(u) &= b_0(u)P_4 + b_1(u)P_5 + b_2(u)P_6 + b_3(u)P_7 \end{aligned} \tag{4. 2}$$

with m as the shape parameter. The endpoints of the two curves are computed as:

Condition 1: Same value of m

$$\begin{aligned} Q_2(0) &= P_4 \\ Q_1(1) &= P_3 \\ Q_2'(0) &= (m+2)P_5 - (m+2)P_4 \\ Q_1'(1) &= (m+2)P_3 - (m+2)P_2 \\ Q_2^{"}(0) &= 2P_6 - (4m+4)P_5 + (4m+2)P_4 \\ Q_1^{"}(1) &= 2P_1 - (4m+4)P_2 + (4m+2)P_3. \end{aligned}$$

Condition 2: Different values of m

$$\begin{aligned} Q_2(0) &= P_4 \\ Q_1(1) &= P_3 \\ Q_2'(0) &= (m_1+2)P_5 - (m_1+2)P_4 \\ Q_1'(1) &= (m_2+2)P_3 - (m_2+2)P_2 \\ Q_2^{"}(0) &= 2P_6 - (4m_1+4)P_5 + (4m_1+2)P_4 \\ Q_1^{"}(1) &= 2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3. \end{aligned}$$

For C^0 Continuity by keeping the initial curve that is $Q_1(u;m)$ and the second curve $Q_2(u;m)$ equal, the results are as shown below:

$$Q_2(0) = Q_1(1) \tag{4. 3}$$

$$P_4 = P_3. \tag{4. 4}$$

In order to achieve C^1 Continuity, it is necessary for the tangent vectors of both the first and second curve segments to be equal at their corresponding first and last points. Next, the designers establish an additional control point for the second curve segment, denoted as P_5, using Equation (4.6) as illustrated below. Then, the remaining control points are determined.

Since $$Q_2'(0) = Q_1'(1), \tag{4. 5}$$

Condition 1: Same value of m

$$(m+2)P_5 - (m+2)P_4 = (m+2)P_3 - (m+2)P_2$$
$$(m+2)P_5 - (m+2)P_3 = (m+2)P_3 - (m+2)P_2$$
$$P_5 - P_3 = P_3 - P_2$$

Hence,

$$P_5 = 2P_3 - P_2. \tag{4.6}$$

Condition 2: Different values of m

$$(m_1+2)P_5 - (m_1+2)P_4 = (m_2+2)P_3 - (m_2+2)P_2$$
$$(m_1+2)P_5 - (m_1+2)P_3 = (m_2+2)P_3 - (m_2+2)P_2$$

Hence,

$$P_5 = \frac{(m_2+2)P_3 - (m_1+2)P_3 - (m_2+2)P_2}{(m_1+2)} \tag{4.7}$$

From Condition 1 and Condition 2, it is identified that the shape parameter cannot be set to $m = -2$, as this value results in breakdown of segment connectivity. Therefore, to ensure that the curve segments remain properly connected in subsequent constructions, the admissible interval for the shape parameter is revised to $m \in (-2,1]$.

Figure 4.5 demonstrates the C^1 continuity of cubic trigonometric curves with different values of shape parameter m in the range $m = (-2,1]$. Meanwhile, in Figure 4.6, cubic trigonometric spline is tested by generate curve with shape parameter m which is out of the given range to see the behaviour of the curve. The specified values of control points $P_0 = (0,0), P_1 = (2,6), P_2 = (6,6), P_3 = (8,0), P_6 = (14,0)$ and $P_7 = (16,0)$. The solid lines have been generated by the control points of the curve $Q_1(u)$ while for the curve $Q_2(u)$ referring to the dotted lines were obtained by using continuity conditions with the new control points acquired. This approach is applied consistently in all subsequent continuity examples.

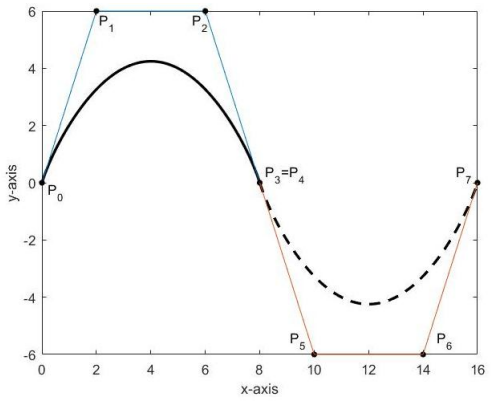
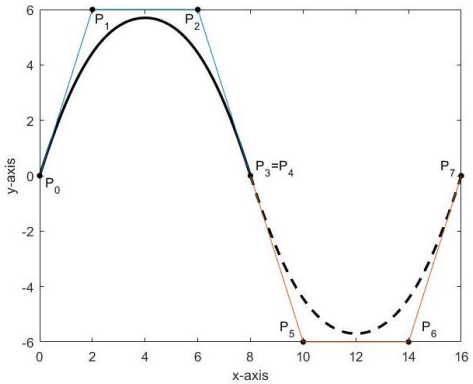
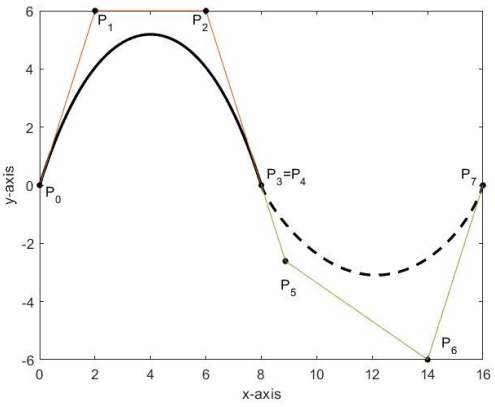
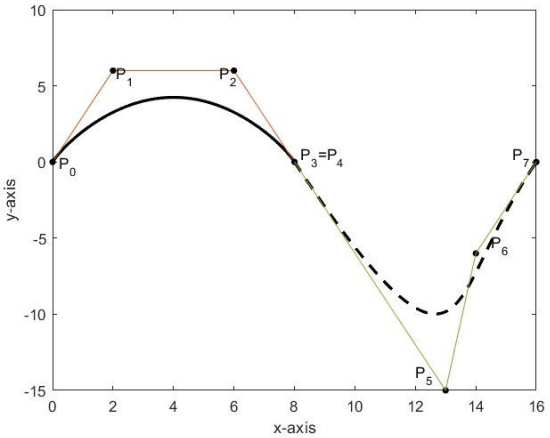
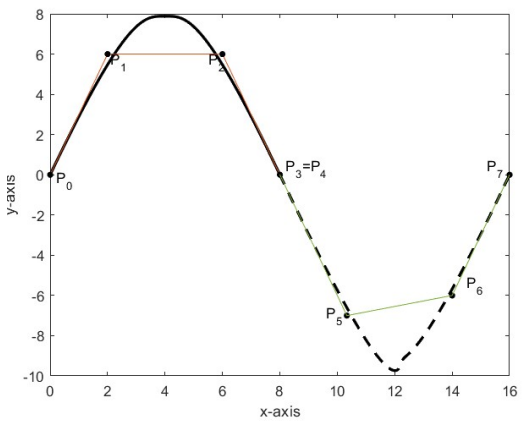
<table>
<tr><th colspan="2">Condition 1: Same value of m</th></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>a) $m = -1$</td><td>b) $m = 1$</td></tr>
<tr><th colspan="2">Condition 2: Different values of m</th></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>c) $m_1 = 0.3, m_2 = -1$</td><td>d) $m_1 = -1, m_2 = 0.5$</td></tr>
</table>

Figure 4.5 C^1 Continuity of Cubic Trigonometric Curve with Various Shape Parameter, m.

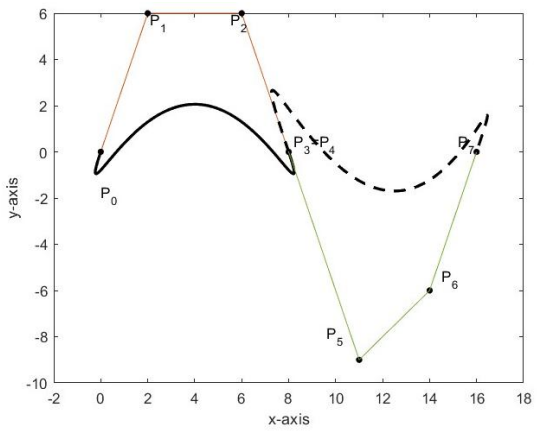
Two piecewise curves that are connected satisfy the C^0 and C^1 continuity conditions as display in Figure 4.5. This figure compares the C^1 continuity of two adjacent cubic trigonometric curve segments under two conditions. In Condition 1, both segments are connected with the same value of the shape parameter, m. The curves meet smoothly at the joining point $P_3 = P_4$, and the transition is visually uniform, as shown in Figure 4.5(a) and Figure 4.5(b).

In Condition 2, different values of m are used for each segment. Figure 4.5(c) and Figure 4.5(d) demonstrate that, despite the different values of shape parameter used

on each side, the condition ensures that the segments join with C^1 smoothness. The curvature may change across the boundary, but the tangent direction remains continuous. The results show that the cubic trigonometric formulation maintains C^1 continuity in both conditions.



a) $m_1 = 4, m_2 = 5$



b) $m_1 = -4, m_2 = -5$

Figure 4.6 C^1 Continuity of Cubic Trigonometric Curve with an Out-of-Range Shape Parameter.

From Figure 4.6 when the shape parameter value tested is out of the given range, the cubic trigonometric curve generates out of the convex hull for Figure 4.6(a). It deviated significantly from the control polygon. Meanwhile, in Figure 4.6(b), the oscillations appear on the curve and the shape becomes unstable and visually inaccurate. These examples clearly show that, when the shape parameters exceed their valid range, the trigonometric curve fails to produce meaningful or stable shapes.

4.2.2 Cubic Trigonometric Spline Interpolation with C^2 Continuity Conditions

The earlier discussion has centred on C^1 continuity of the curves, the subsequent section now addresses the C^2 continuity in detail. C^2 continuity constraints, the second derivative of both the first and second segments is kept equal along with C^1 continuity conditions. The new control point P_6 will be achieved, and the remaining control points are set freely.

Theorem 4.2: By adjusting the values of the shape parameter, m, cubic trigonometric curves $Q_1(u)$ and $Q_2(u)$ are C^2 continuous if they are satisfying the C^0, C^1continuity conditions.

Proof. The two curves are connected by C^2 continuity if

$$Q_2''(0) = Q_1''(1) \tag{4. 8}$$

Condition 1: Same value of m

$$(4m+2)P_4 - (4m+4)P_5 + 2P_6 = (4m+2)P_3 - (4m+4)P_2 + 2P_1$$
$$(4m+2)P_3 - (4m+4)P_5 + 2P_6 = (4m+2)P_3 - (4m+4)P_2 + 2P_1$$
$$-(4m+4)(2P_3 - P_2) + 2P_6 = -(4m+4)P_2 + 2P_1$$
$$-2(4m+4)P_3 + 2(4m+4)P_2 + 2P_6 = 2P_1$$

Since

$$-(4m+4)P_3 + (4m+4)P_2 + P_6 = P_1$$

Hence

$$P_6 = P_1 - (4m+4)P_2 + (4m+4)P_3 \tag{4. 9}$$

Condition 2: Different values of m

$$(4m_1+2)P_4 - (4m_1+4)P_5 + 2P_6 = (4m_2+2)P_3 - (4m_2+4)P_2 + 2P_1$$
$$(4m_1+2)P_3 - (4m_1+4)P_5 + 2P_6 = (4m_2+2)P_3 - (4m_2+4)P_2 + 2P_1$$
$$-(4m+4)(2P_3 - P_2) + 2P_6 = -(4m_1+4)P_2 + 2P_1$$
$$2P_6 = 2P_1 + (4m_2+2)P_3 - (4m_2+4)P_2 - (4m_1+2)P_3 - (4m_1+4)P_5$$

Hence

$$P_6 = P_1 + \frac{(4m_2+2)P_3 - (4m_2+4)P_2 - (4m_1+2)P_3 - (4m_1+4)P_5}{2} \tag{4. 10}$$

Figure 4.7 illustrates the behaviour of two adjacent cubic trigonometric curve segments that are joined under conditions required for achieving C^2 continuity. In Figure 4.7(a) and Figure 4.7(b), a cubic trigonometric spline made up of $Q_1(u)$ and $Q_2(u)$ curves given in Equation (4.2) are developed using control points $P_0 = (5,0), P_1 = (0,4), P_2 = (5,7)$, $P_3 = (10,7)$ and $P_7 = (15,0)$. Meanwhile, curves are generated using control points $P_0 = (0,0), P_1 = (2,6), P_2 = (6,6)$, $P_3 = (8,0)$, and $P_7 = (16,0)$ for Figure 4.7(c) and Figure 4.7(d).

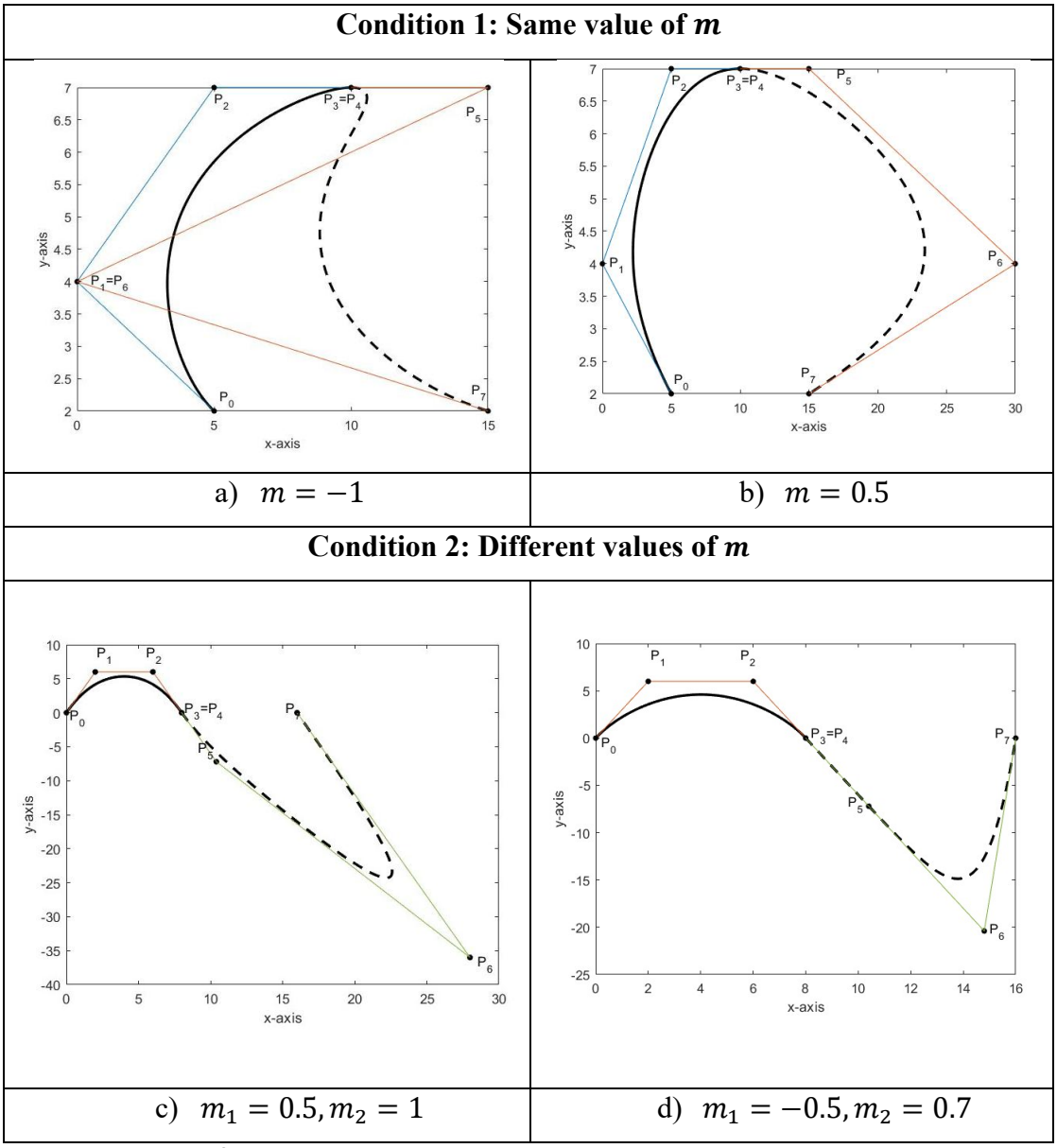


Figure 4.7 C^2 Continuity of Cubic Trigonometric Curve with Various Shape Parameter, m.

In Condition 1, both connective segments apply the same values of shape parameter, m. Figure 4.7(a) and Figure 4.7(b) show that the curves meet at the joining point and having the same slope and curvature at the join.

In Condition 2, the two segments employ different values of m. Although the generated curves differ away from the boundary, the continuity enforce equality of the second derivatives at the joining point. This guarantees that curvature remains continuous even when the shape parameter values of each connected curves vary.

The Figure 4.7 demonstrates that cubic trigonometric spline satisfied C^2 continuity for both equal and unequal values of the shape parameter yet generating smooth curves.

4.2.3 Cubic Trigonometric Spline Interpolation with G^1 Continuity Conditions

Next, cubic trigonometric curves are linked by geometric continuity conditions. Geometric continuity helps the designers construct and handle complex figures and surfaces better than parametric continuity because it gives more flexibility due to the presence of a scalar factor. First, cubic trigonometric splines are generated using G^1 continuity condition.

Theorem 4.3: By varying the values of the shape parameter, m, and scalar factor, α, cubic trigonometric curves $Q_1(u)$ and $Q_2(u)$ are G^1 continuity at the linked point.

Proof. Curves defined in (4.2) are joined together with G^1 continuity if,

$$Q_2(0) = Q_1(1)$$

where the G^0 continuity conditions are satisfied, and obtained the first control point of the second curve is,

$$P_4 = P_{3.} \tag{4. 11}$$

For G^1 continuity constraints, both curves are kept equal, and also the first derivative of both curves involves a scale factor such as,

$$Q_2(0) = Q_1(1)$$

$$Q_2'(0) = \alpha Q_1{}'(1), \quad \alpha > 0 \tag{4.12}$$

By solving the above equation, the next control points of the second curve are achieved,

Condition 1: Same value of m

$$(m+2)P_5 - (m+2)P_4 = \alpha[(m+2)P_3 - (m+2)P_2]$$
$$(m+2)P_5 - (m+2)P_3 = \alpha(m+2)P_3 - \alpha(m+2)P_2$$
$$P_5 - P_3 = \alpha P_3 - \alpha P_2$$

Hence, the condition for G^1 continuity is

$$P_5 = \alpha P_3 - \alpha P_2 + P_3$$
$$P_5 = \alpha(P_3 - P_2) + P_3 \tag{4. 13}$$

Condition 2: Different values of m

$$(m_1+2)P_5 - (m_1+2)P_4 = \alpha[(m_2+2)P_3 - (m_2+2)P_2]$$
$$(m_1+2)P_5 - (m_1+2)P_3 = \alpha(m_2+2)P_3 - \alpha(m_2+2)P_2$$

Hence, the condition for G^1 continuity is

$$P_5 = \frac{\alpha(m_2+2)P_3 - \alpha(m_2+2)P_2 + (m_1+2)P_3}{(m_1+2)}. \tag{4. 14}$$

Figure 4.8 demonstrates the behaviour of two adjacent cubic trigonometric curve segments in achieving G^1 continuity. The control points used are $P_0 = (0,0), P_1 = (2,6), P_2 = (6,6), \quad P_3 = (8,0), P_6 = (14,-6)$ and $P_7 = (16,0)$. The remaining two control points are determined by using the continuity constraints conditions described in (4.11), (4.13), and (4.14). The constructed spline is adjusted using the different values of the shape parameter, m, and the scalar factor, α that are shown in the figures. The presence of both variables had great worth in modifying the shape of the curve.

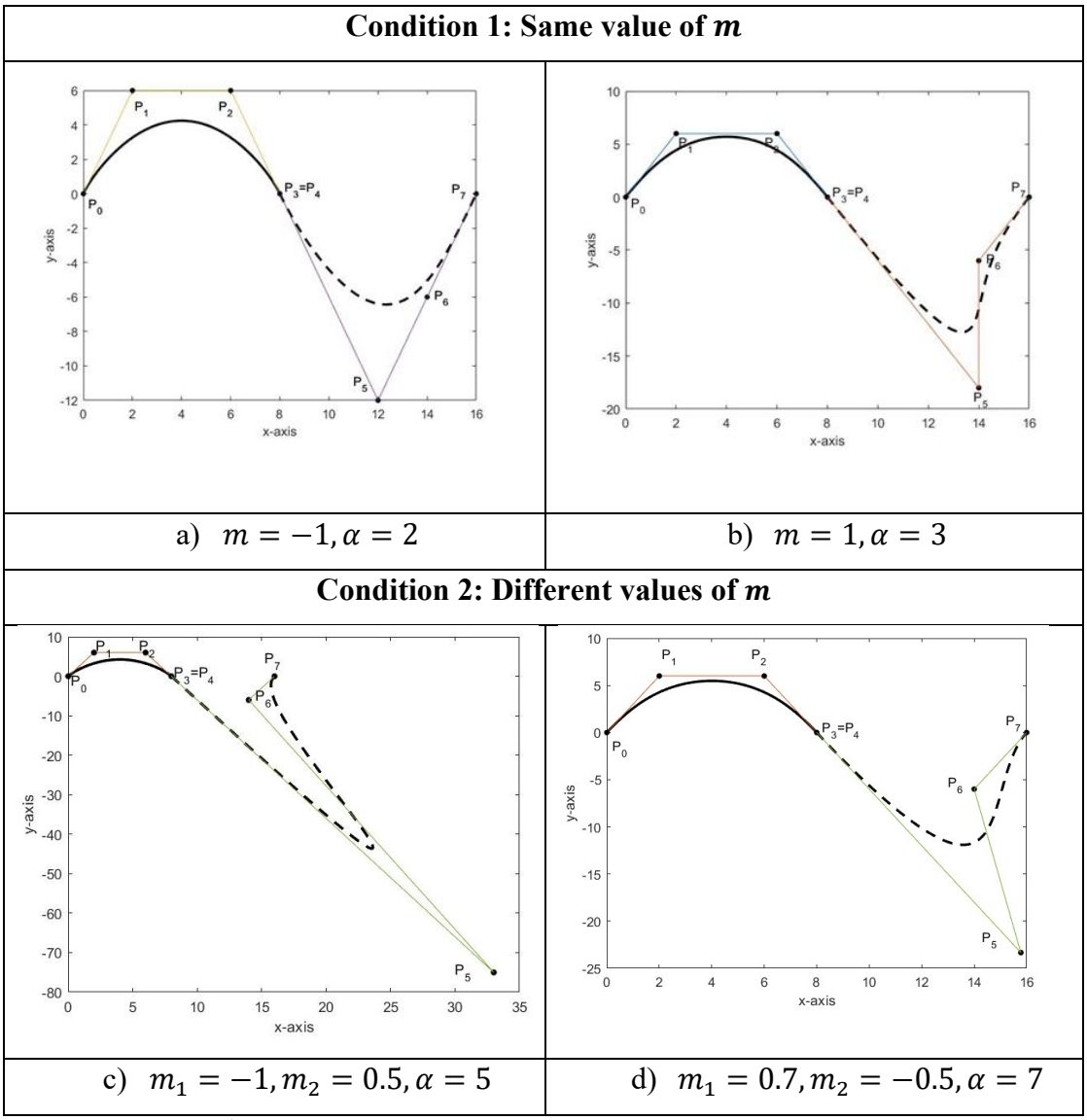


Figure 4.8 G^1 Continuity of Cubic Trigonometric Curve with Various Shape Parameter, m and Scalar Factor, α.

In Figure 4.8, all the generated cubic trigonometric curves are presented in the convex hull. Figures in Condition 1 show that the tangent directions and curvature on both sides of the join are consistent, resulting in a smooth and coherent transition. In Condition 2, the curves generated maintains tangent proportionality at the join. Although the curvature on each side may differ, the connection remains geometrically smooth. For both conditions, the generated cubic trigonometric spline satisfied the G^1 continuity condition.

4.2.4 Cubic Trigonometric Spline Interpolation with G^2 Continuity Conditions

Next, G^2 continuity conditions are presented. The piecewise cubic trigonometric splines are joined and constructed using G^2 continuity. The splines are tested with some conditions by adjusting the values of the shape parameter and scalar factor.

Theorem 4.4: Let α and β be the scalar factor, and m is the shape parameter. The cubic trigonometric curves $Q_1(u)$ and $Q_2(u)$ are joined by G^2 continuity if they are satisfying the G^0 and G^1 continuity conditions.

Proof. The two curves are joined by G^2 continuity if

$$Q_2(0) = Q_1(1)$$

$$Q_2'(0) = \alpha Q_1'(1),$$

$$Q_2''(0) = \alpha^2 Q_1''\left(\frac{\pi}{2}\right) + \beta Q_1'(1) \tag{4. 15}$$

whereupon the subsequent requirements need to be met.

$$\alpha > 0, \beta \in \mathbb{R}. \tag{4.16}$$

By substituting the endpoints into (4.12), the equation becomes,

Condition 1: Same value of m

$$2P_6 - (4m+4)P_5 + (4m+2)P_4 = \alpha^2[2P_1 - (4m+4)P_2 + (4m+2)P_3] + \beta[(m+2)P_3 - (m+2)P_2]$$

Hence, the condition for G^2 continuity is when

$$P_6 = \left(2\alpha^2 m + \alpha^2 + \frac{\beta m}{2} + \beta + 1 + 2m\alpha + 2\alpha\right)P_3 - \left(2m\alpha + 2\alpha + 2\alpha^2 m + 2\alpha^2 + \frac{\beta m}{2} + \beta\right)P_2 + \alpha^2 P_1 \tag{4. 17}$$

Condition 2: Different values of m

$$2P_6 - (4m_1 + 4)P_5 + (4m_1 + 2)P_4 = \alpha^2[2P_1 - (4m_2 + 4)P_2 + (4m_2 + 2)P_3] + \beta[(m_2 + 2)P_3 - (m_2 + 2)P_2]$$

$$2P_6 = (4m_1 + 4)P_3 - (4m_1 + 2)P_3 + \alpha^2[2P_1 - (4m_2 + 4)P_2 + (4m_2 + 2)P_3 + \beta(m_2 + 2)P_3 - (m_2 + 2)P_2]$$

Hence, the condition for G^2 continuity is when

$$P_6 = \frac{(4m_1 + 4)P_3 - (4m_1 + 2)P_3 + \alpha^2[2P_1 - (4m_2 + 4)P_2 + (4m_2 + 2)P_3 + \beta(m_2 + 2)P_3 - (m_2 + 2)P_2]}{2}. \quad (4.\ 18)$$

Figure 4.9 represents the G^2 smooth continuity between two cubic trigonometric curves. The cubic trigonometric curves are generated using the chosen control points $P_0 = (0,0), P_1 = (2,6), P_2 = (6,6)$ and $P_3 = (8,0)$. Multiple values of shape parameters and scalar factors are involved in the construction of the various curves illustrated in the figures below. The change in the shape of the G^2 cubic trigonometric spline is influenced by the values of α, β, and m.

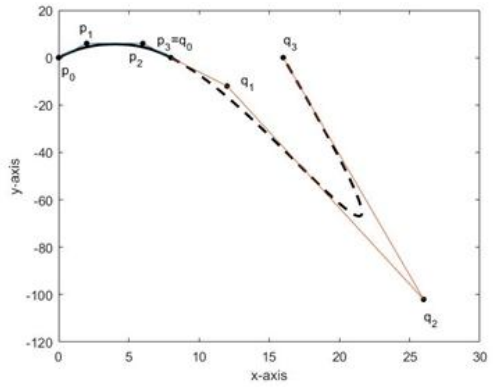
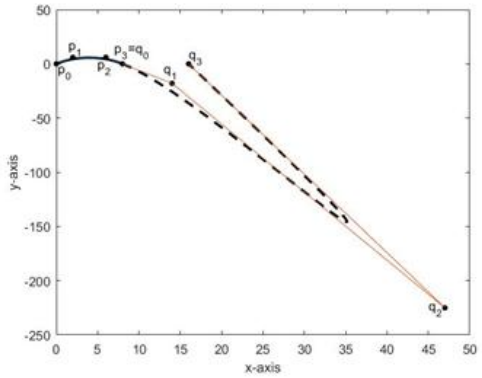
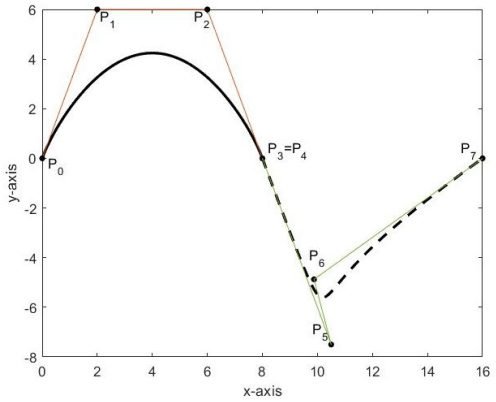
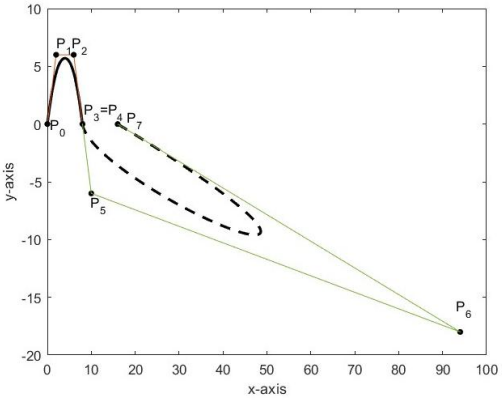
Condition 1: Same value of m	
	
a) $m = 1, \alpha = 2,\ \beta = -2$	b) $m = 1, \alpha = 3,\ \beta = -3$
Condition 2: Different values of m	
	
c) $m_1 = -1, m_2 = 0.5,$ $\alpha = 0.5, \beta = 1.5$	d) $m_1 = 1, m_2 = -0.5,$ $\alpha = 2\ , \beta = 5$

Figure 4. 9 G^2 Continuity of Cubic Trigonometric Curve when $m = 1$, with Varies Values of α and β

In Figure 4.9, for Condition 1, when using same value m with various values of α and β, the join is visually smooth. The curves meet with matching direction and curvature at the boundary. Meanwhile, when using different values of m, the generated curves move away from the control polygon. However, the G^2 continuity conditions ensure that the derivatives still match at the joining point. As a result, the two segments

meet smoothly at the shared point without any break or change in curvature. Overall, these continuity conditions guarantee a smooth transition in position direction, and curvature between the connected segments for both conditions.

4.3 Application of Cubic Trigonometric Spline Interpolation in Preserving Positivity of Real Data

In CAGD, another obstacle researchers found is in maintaining and retaining factual data information. The interpolation of the generated curve should preserve the properties and authenticity of the data, which are commonly obtainable in positivity, monotonicity, or convexity manners. This research would only focus on preserving the positivity of the positive data set.

Positivity-preserving data refers to a dataset in which all values are non-negative. A spline is said to preserve positivity if it guarantees that the resulting entire curve remains non-negative between the data points. In general, the cubic trigonometric spline does not always preserve the positivity of the given data sets. So, determination of suitable values of shape parameter and scalar factor is important to solve the problems. Thus, the schemes continuity of the interpolation of cubic trigonometric are presented for preserving the shape of the positive data and also achieve the parametric and geometric continuity of curves.

For positive data spline interpolation, let the data set be positive; that is $x_0 < x_1 \ldots < x_n$ for $f_0 > 0, f_1 > 0 \ldots f_n > 0$. The shape parameters in each scheme will benefit in achieving the desired shape-preserving splines, including ensuring the spline's smoothness and continuity.

4.3.1 Cubic Trigonometric Spline Interpolation with C^1 Continuity Conditions

Let $u = \frac{x-x_i}{h_i}$ be the normalized local parameter on the subinterval $[x_i, x_{i+1}]$ with $u \in [0,1]$ and $h_i = x_{i+1} - x_i$. The segment is expressed as a linear combination of cubic trigonometric basis functions with one shape parameter m, where $m = (-2,1]$:

$$P_i(x) = b_0(u) \cdot V_i + b_1(u) \cdot W_i + b_2(u) \cdot X_i + b_3(u) \cdot Y_i, \quad 0 \le u \le 1 \tag{4.19}$$

where b_0, b_1, b_2 and b_3 are the cubic trigonometric blending functions in (3.7) and V_i, W_i, X_i, Y_i are the four control points to be determined. Let $(x_i, f_i), i = 0,1, \dots, n,$ be a given set of data points, where $x_0 < x_1 < \dots < x_n$. f_i. Meanwhile, d_i and d_{i+1} represent the values of derivatives by the Arithmetic Mean Method from Hussain & Sarfraz (2009) at the knots x_i and x_{i+1}.

To ensure that the resulting spline is globally C^1, the interpolant on each subinterval must satisfy the following endpoint conditions:

$$\begin{aligned} P(x_i) &= f_i, \qquad P(x_{i+1}) = f_{i+1} \\ P'(x_i) &= d_i, \qquad P'(x_{i+1}) = d_{i+1}. \end{aligned} \tag{4. 20}$$

where f_i and f_{i+1} are the data values and d_i and d_{i+1} are the specified or computed derivative values at the knots.

Below are the steps control points in achieving C^1 continuity:

Step 1 – Imposing the function value conditions

Evaluate P_i at the endpoints $u = 0$ and $u = 1$,

The basis functions satisfy:

$$\begin{aligned} b_0(0) = 1\,, b_1(0) = b_2(0) = \ b_3(0) = 0, \\ b_3(1) = 0\,, b_0(1) = \ b_1(1) = \ b_2(1) = 1. \end{aligned}$$

Thus,

$$P_i(0) = V_i = f_i, \qquad P_i(1) = Y_i = f_{i+1}.$$

So, the first and control points are fixed directly by interpolation:

$$P_i V_i = f_i, \qquad Y_i = f_{i+1}.$$

Step 2 – Imposing the first-derivative conditions

Differentiate the spline with respect to u:

$$P_i'(x) = b_0'(u) \cdot V_i + b_1'(u) \cdot W_i + b_2'(u) \cdot X_i + b_3'(u) \cdot Y_i,$$

Evaluating at $u = 0$ and $u = 1$ gives,

$$P_i'(0) = d_i, \qquad P_i'(1) = d_{i+1}.$$

The derivatives of the basis functions at the endpoints form two linear equations involving the unknown control points W_i and X_i. When these equations are solved, with V_i and Y_i already known, the resulting formulas for the inner control points are:

$$W_i = \frac{2h_i d_i}{\pi(m_i + 2)} + f_i, \qquad X_i = \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}.$$

The control points defining the C^1 cubic trigonometric interpolant on the interval $[x_i, x_{i+1}]$ is:

$$\begin{aligned} V_i &= f_i; \\ W_i &= \frac{2h_i d_i}{\pi(m_i + 2)} + f_i; \\ X_i &= \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}; \\ Y_i &= f_{i+1}. \end{aligned} \tag{4. 21}$$

By using these control points, each segment satisfies the required interpolation and first-derivative conditions, ensuring C^1 smoothness across all subintervals.

A real positive data set from Dube & Singh (2014) is use in Table 4.1, Table 4.2, Table 4.3, and, Table 4.4. The shape parameter m_i, n_i, α_i and β_i is used to control the shape of the curve segment between the data points x_i and x_{i+1}. For example, each value m_i influences the shape of the i-th curve segment.

Table 4. 1
Positive Data Set (I)

i	x_i	f_i	m_i
1	1	3	0.5
2	2	6	1
3	3	5	-0.5
4	4	8	-1
5	5	1	-

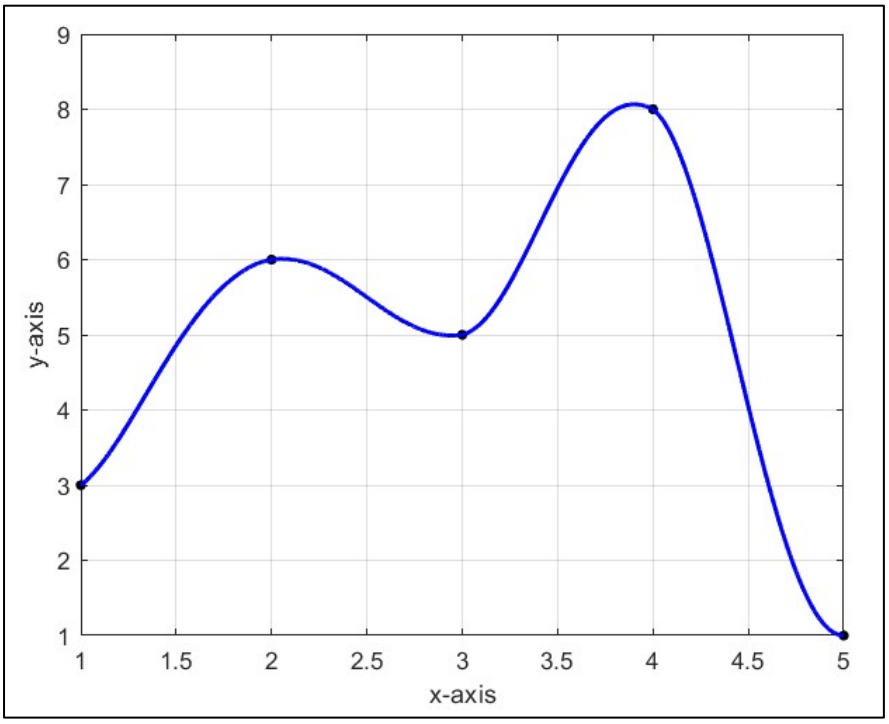


Figure 4.10 Positivity-Preserving Cubic Trigonometric Spline Curve with C^1 Continuity

Figure 4.10 illustrates the cubic trigonometric spline generated using the C^1 continuity condition using different selections of the free shape parameter m. For each adjoining segment, the value of m is adjusted to ensure that the resulting curve remains entirely above the horizontal axis, thereby preserving the positivity of the given data

set. Overall, the spline is able to preserve the shape of the positive data sets while maintaining continuity and smoothness.

4.3.2 Cubic Trigonometric Spline Interpolation with C^2 Continuity Conditions

The interpolant is obtained by expressing the cubic trigonometric spline on each subinterval $[x_i, x_{i+1}]$ using equation in (4.19).

The required C^2 continuity constraints are:

$$\begin{aligned} P(x_i) &= f_i, \qquad P(x_{i+1}) = f_{i+1}, \\ P'(x_i) &= d_i, \qquad P'(x_{i+1}) = d_{i+1}, \\ P''(x_i) &= D_i, \qquad P''(x_{i+1}) = D_{i+1}. \end{aligned} \tag{4.22}$$

Here, f_i and f_{i+1} are the function values, d_i and d_{i+1} are the first derivative values, and D_i and D_{i+1} denote the second derivative values at the boundaries of the segment.

To obtain the equation of the interpolant that satisfies C^2continuity, the following steps are used:

1. Impose the function-value conditions

$$P_i(0) = f_i, \qquad P_i(1) = f_{i+1}.$$

These two conditions define the first and the last control points:

$$V_i = f_i, \qquad Y_i = f_{i+1}.$$

2. Impose the first-derivative conditions

$$P_i'(0) = d_i, \qquad P_i'(1) = d_{i+1}.$$

Differentiating the spline with respect to u and apply the above equation, yields two linear equations in the remaining control points W_i and X_i. Solving these gives

$$W_i = \frac{2h_i d_i}{\pi(m_i + 2)} + f_i; \qquad X_i = \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1};$$

3. Impose the second-derivative conditions

$$P_i''(0) = D_i, \qquad P_i''(1) = D_{i+1}.$$

To enforce C^2 continuity, the actual curvature of each cubic trigonometric segment must be evaluated at the endpoints. By differentiating the spline twice and substituting the control points obtained above, the expressions for the second derivatives become

$$\begin{aligned} D_i = {} & \left(\frac{\pi^2 + 2m_i\pi^2}{2}\right) f_i + \pi^2(-1 - m_i)\left(\frac{2h_i d_i}{\pi(m_i + 2)} + f_i\right) \\ & + \frac{\pi^2}{2}\left(\frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}\right) \end{aligned} \tag{4.23}$$

$$\begin{aligned} D_{i+1} = {} & \frac{\pi^2}{2}\left(\frac{2h_i d_i}{m_i\pi + 2\pi} + f_i\right) + \pi^2(-1 - m)\left(\frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}\right) \\ & + \left(\frac{\pi^2}{2} + m_i\pi^2\right) f_{i+1}. \end{aligned} \tag{4.24}$$

The C^2 cubic trigonometric interpolant on the interval $[x_i, x_{i+1}]$ is specified by the following set of control points:

$$\begin{aligned} V_i &= f_i; \\ W_i &= \frac{2h_i d_i}{\pi(m_i + 2)} + f_i; \\ X_i &= \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}; \\ Y_i &= f_{i+1}. \end{aligned} \tag{4.25}$$

Table 4. 2
Positive Data Set (II)

i	x_i	f_i	m_i
1	1	3	0.5
2	2	6	-1
3	3	5	0.1
4	4	8	-0.5
5	5	1	-

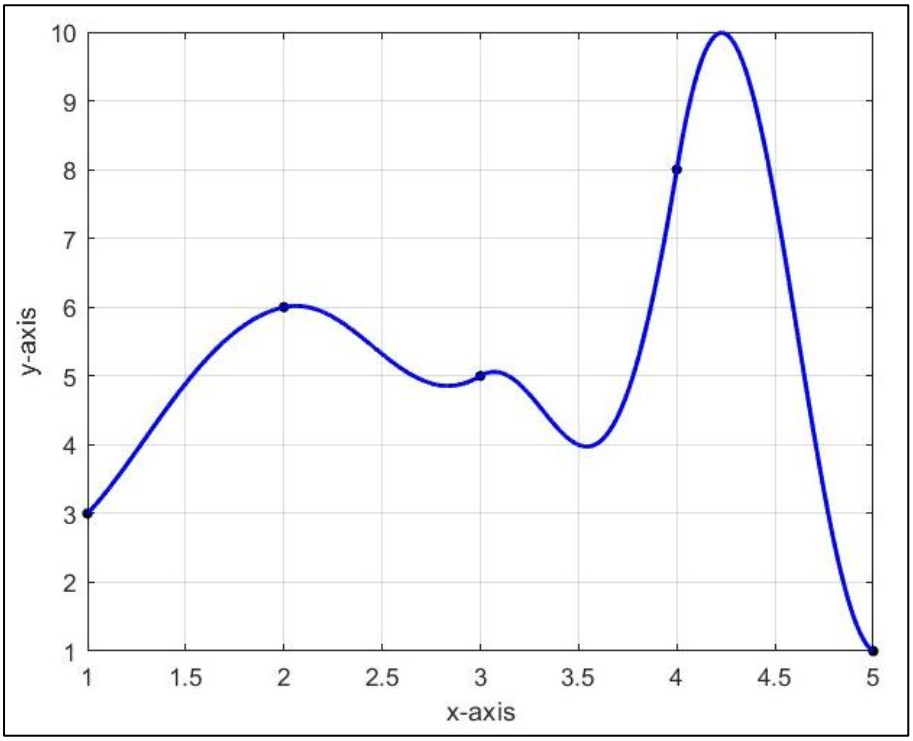


Figure 4.11 Positivity-Preserving Cubic Trigonometric Spline Curve with C^2 Continuity

Figure 4.11 displays the cubic trigonometric spline constructed through the given data points. The curve interpolates each point exactly while maintaining a high level of smoothness across the intervals. By using appropriate selection of the shape parameter, m, the cubic trigonometric spline inherited the features of the data. All piecewise segments satisfy the C^2 continuity requirements.

4.3.3 Cubic Trigonometric Spline Interpolation with G^1 Continuity Conditions

The cubic trigonometric interpolant is obtained on each subinterval $[x_i, x_{i+1}]$ is formulated through the spline representation defined in equation (4.19).

Cubic trigonometric spline achieved G^1 continuity by satisfying the below conditions,

$$\begin{aligned} P(x_i) &= f_i, \quad & P(x_{i+1}) &= f_{i+1}, \\ P'(x_i) &= n_i d_i, \quad & P'(x_{i+1}) &= d_{i+1}, \end{aligned} \tag{4. 26}$$

where

- f_i and f_{i+1} are the function values,
- d_i and d_{i+1} represent the specified endpoint derivatives,
- $n_i > 0$ is a scaling factor that adjust the magnitude of the derivative at x_i.

The steps required to formulate the G^1- continuous interpolant are as follows:

1. **Determination the endpoint control points**

   As in the C^1 and C^2 cases, the first and last control points are fixed by the interpolation conditions:

   $$V_i = f_i, \qquad Y_i = f_{i+1}.$$

   These guaranteed that the spline interpolates the given data.

2. **Imposing the geometric derivative conditions**

   Differentiating the spline gives

   $$P_i'(x) = b_0'(u) \cdot V_i + b_1'(u) \cdot W_i + b_2'(u) \cdot X_i + b_3'(u) \cdot Y_i,$$

   and evaluating at the endpoints yields two linear equations involving W_i and X_i:

   $$P_i'(0) = n_i d_i, \qquad P_i'(1) = d_{i+1}.$$

The equations can be solved explicitly, giving

$$W_i = \frac{2h_i d_i}{\pi(m_i + 2)} n_i + f_i, \qquad X_i = \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}.$$

The G^1 cubic trigonometric interpolant over the interval $[x_i, x_{i+1}]$ is determined by the following control points:

$$\begin{aligned} V_i &= f_i; \\ W_i &= \frac{2h_i d_i}{\pi(m_i + 2)} n_i + f_i; \\ X_i &= \frac{-2h_i d_{i+1}}{\pi(m_i + 2)} + f_{i+1}; \\ Y_i &= f_{i+1}; \end{aligned} \tag{4. 27}$$

Table 4. 3
Positive Data Set (III)

$\boldsymbol{i}$	$\boldsymbol{x_i}$	$\boldsymbol{f_i}$	$\boldsymbol{m_i}$	$\boldsymbol{n_i}$
1	1	3	1	0.1
2	2	6	0.6	2
3	3	5	0.5	3
4	4	8	-1	1.5
5	5	1	-	-

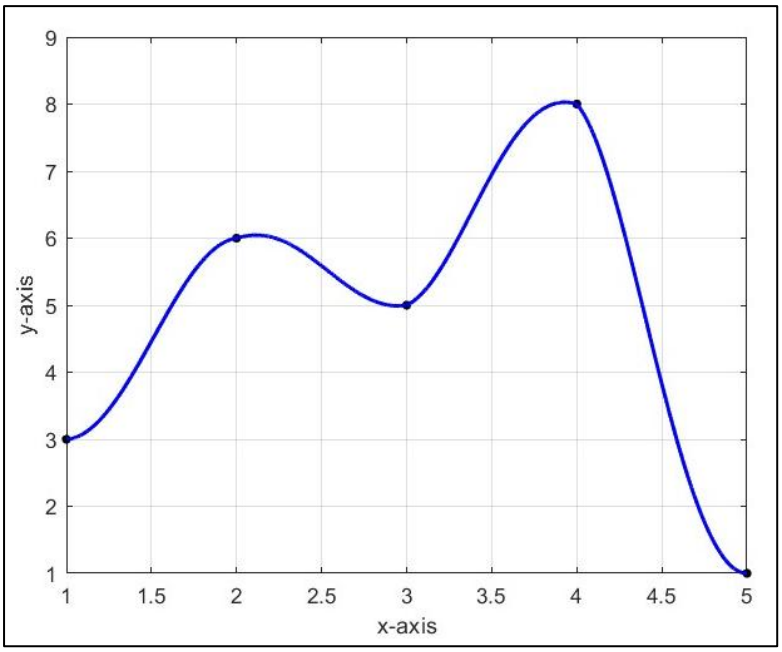


Figure 4.12 Positivity-Preserving Cubic Trigonometric Spline Curve with G^1 Continuity

Visualization of the positive data is shown in Figure 4.12 by using the G^1 continuity conditions to preserve the shape of positive data. The generated curves pass smoothly through all data points. The values of the shape parameter, m and scalar factor, n are selected through empirical adjustment to obtain the desired curve behaviour while maintaining the features of the positive data.

4.3.4 Cubic Trigonometric Spline Interpolation with G^2 Continuity Conditions

The cubic trigonometric interpolant is obtained on each subinterval $[x_i, x_{i+1}]$ is formulated through the spline representation defined in equation (4.19).

1. Geometric continuity conditions

For two consecutive segments, the G^2 compatibility constraints are

$$\begin{aligned} P_{i+1}(0) &= P_i(1) \\ P'_{i+1}(0) &= \alpha P'_i(1) \\ P''_{i+1}(0) &= \alpha^2 P''_i(1) + \beta P'_i(1) \end{aligned} \tag{4. 28}$$

where

- $\alpha > 0$ controls the scale of the tangent vector
- β introduces an additional degree of freedom that governs curvature matching.

Cubic trigonometric spline achieved G^1 continuity by satisfying the below conditions,

$$\begin{aligned} P(x_i) = f_i, \quad & P(x_{i+1}) = f_{i+1}, \\ P'(x_i) = \alpha d_i, \quad & P'(x_{i+1}) = d_{i+1}, \\ P'(x_i) = \alpha^2 D_{i+1} + \beta D_i, \quad & P^{"}(x_{i+1}) = D_{i+1}. \end{aligned} \tag{4. 29}$$

2. Endpoint evaluation using basis function derivatives

Next, evaluate the derivative expressions at $u = 0$ and $u = 1$.

At $= 0$:

$$b_0'(0) = -\pi - \frac{m\pi}{2}, \quad b_1'(0) = \pi + \frac{m\pi}{2}, \quad b_2'(0) = 0, \quad b_3'(0) = 0,$$
$$b_0''(0) = \frac{\pi^2}{2} + m\pi^2, \quad b_1''(0) = -\pi^2 - m\pi^2, \quad b_2''(0) = \frac{\pi^2}{2}, \quad b_3''(0) = 0.$$

At $t = 1$:

$$b_0'(1) = 0, \quad b_1'(1) = 0, \quad b_2'(1) = -\pi - \frac{m\pi}{2}, \quad b_3'(1) = \pi + \frac{m\pi}{2},$$
$$b_0''(1) = 0, \quad b_1''(1) = \frac{\pi^2}{2}, \quad b_2''(1) = -\pi^2 - m\pi^2, \quad b_3''(1) = \frac{\pi^2}{2} + m\pi^2.$$

Now substitute the endpoint values into the expressions for $P'$and P'':

At $u = 0$:

$$P(0) = V_0$$

$$\begin{aligned} P'(0) &= V_0 b_0'(0) + V_1 b_1'(0) + V_2 b_2'(0) + V_3 b_3'(0) \\ &= V_0\left(-\pi - \frac{m\pi}{2}\right) + V_1\left(\pi + \frac{m\pi}{2}\right) + V_2(0) + V_3(0) \\ &= \left(-\pi - \frac{m\pi}{2}\right)V_0 + \left(\pi + \frac{m\pi}{2}\right)V_1, \end{aligned}$$

$$
\begin{aligned}
P''(0) &= V_0 b_0''(0) + V_1 b_1''(0) + V_2 b_2''(0) + V_3 b_3''(0) \\
&= V_0 \left(\frac{\pi^2}{2} + m\pi^2\right) + V_1(-\pi^2 - m\pi^2) + V_2\left(\frac{\pi^2}{2}\right) + V_3(0) \\
&= \left(\frac{\pi^2}{2} + m\pi^2\right) V_0 + (-\pi^2 - m\pi^2) V_1 + \left(\frac{\pi^2}{2}\right) V_2.
\end{aligned}
$$

At $u = 1$ the same procedure gives

$$
\begin{aligned}
P(1) &= V_3 \\
P'(1) &= \left(-\pi - \frac{m\pi}{2}\right) V_2 + \left(\pi + \frac{m\pi}{2}\right) V_3 \\
P''(0) &= \left(\frac{\pi^2}{2} + m\pi^2\right) V_0 + (-\pi^2 - m\pi^2) V_1 + \frac{\pi^2}{2} V_2 \\
P^{"}(1) &= \frac{\pi^2}{2} V_1 + (-\pi^2 - m\pi^2) V_2 + \left(\frac{\pi^2}{2} + m\pi^2\right) V_3.
\end{aligned}
$$

So, the endpoint expressions are in equation (4.28).

Endpoint property

$$
\begin{aligned}
P(0) &= V_0 \\
P(1) &= V_3 \\
P'(0) &= \left(-\pi - \frac{m\pi}{2}\right) V_0 + \left(\pi + \frac{m\pi}{2}\right) V_1 \\
P'(1) &= \left(-\pi - \frac{m\pi}{2}\right) V_2 + \left(\pi + \frac{m\pi}{2}\right) V_3 \\
P''(0) &= \left(\frac{\pi^2}{2} + m\pi^2\right) V_0 + (-\pi^2 - m\pi^2) V_1 + \frac{\pi^2}{2} V_2 \\
P^{"}(1) &= \frac{\pi^2}{2} V_1 + (-\pi^2 - m\pi^2) V_2 + \left(\frac{\pi^2}{2} + m\pi^2\right) V_3.
\end{aligned}
\qquad (4.30)
$$

3. **Applying G^2 constraints to adjacent segments**

   Consider two consecutive segments:

$$P_1(u) = \sum_{i=0}^{3} b_i(u) V_i^1, \tag{4. 31}$$

$$P_2(u) = \sum_{i=0}^{3} b_i(u) V_i^2 \tag{4. 32}$$

The G^2 continuity requirements at their joining point give:

$$\begin{aligned} P_2(0) &= P_1(1) \\ P_2'(0) &= \alpha P_1'(1) \\ P_2''(0) &= \alpha^2 P_1''(1) + \beta P_1'(1) \end{aligned} \tag{4. 33}$$

4. Resulting control points for the G^2 spline

$$\begin{aligned} V_0^2 &= V_3^1 \\ V_1^2 &= (V_3^1 - V_2^1)\alpha + V_3^1 \\ V_2^2 &= \alpha^2 V_1^1 + 2(-1-m)(\alpha^2+\alpha)V_2^1 + \frac{2}{\pi}\left(-1-\frac{m}{2}\right)\beta V_2^1 \\ &\quad + 2(1+m)(\alpha+1)V_3^1 + 2\left(\frac{1}{2}+m\right)\alpha^2 V_3^1 \\ &\quad + \frac{2}{\pi}\left(1+\frac{m}{2}\right)\beta V_3^1. \end{aligned} \tag{4. 34}$$

Table 4. 4
Positive Data Set (IV)

i	x_i	f_i	m_i	α_i	β_i
1	1	3	0.5	0.5	4
2	2	6	-1	1	5
3	3	5	-1	2	0.15
4	4	8	0.2	2	3
5	5	1	-	-	-

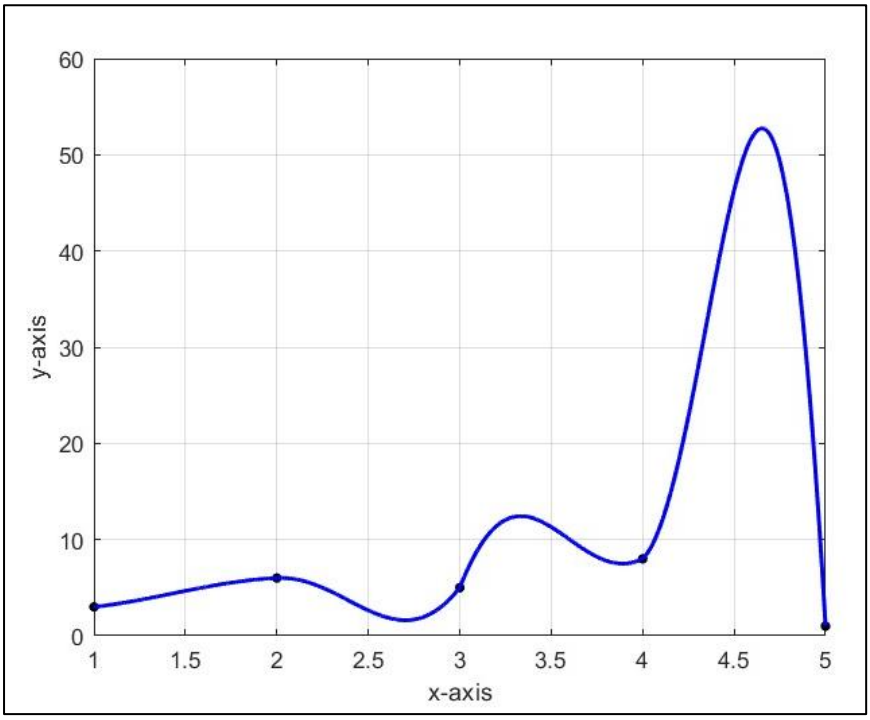


Figure 4.13 Positivity-Preserving Cubic Trigonometric Spline Curve with G^2 Continuity (I)

Lastly, in Figure 4.12, positive interpolation curves are illustrated using G^2 continuity conditions. The spline interpolates all data points exactly while producing a highly smooth curve with visually consistent bending behaviour. As shown in the plot, the spline accurately follows the mild oscillations between $x = 1$ and $x = 4$, capturing the small rise without generating artificial oscillations. The method also accommodates the steep increase near $x = 4.5$, where the values increase rapidly to approximately 55, before dropping sharply towards the endpoint. Despite the dramatic change, the G^2 formulation maintains the smooth curvature transitions across the entire curves. Shape

parameter, m and scalar factors α and β are assigned with different values throughout the spline to maintain the characteristics of the positive shape data. This flexibility allows the interpolant to adapt locally while preserving global smoothness and positivity.

Figure 4.10 - Figure 4.13 show the shape-preserving interpolation results obtained using the proposed cubic trigonometric spline for datasets in Table 4.1 - Table 4.4. The suggested default positive curve is sufficient for a smooth data display. Nevertheless, the presence of the shape parameter m and scalar factor, n, α and β in geometric continuity conditions enables the user more flexibility to adjust the curve according to specific shape requirements.

From Figure 4.10 to Figure 4.13, it is clear that the parametric and geometric continuity of the cubic trigonometric spline that has been developed is capable of maintaining the geometric properties of the provided data sets. Additionally, the resulting adequate conditions for cubic trigonometric spline positivity are simple to apply and can be modified appropriately.

Next, the cubic trigonometric spline is tested with different datasets with containing near-zero value.

4.3.5 Cubic Trigonometric Spline Interpolation with G^1 and G^2 Continuity Conditions

Table 4.5
Positive Data Set (V)

i	x_i	f_i	m_i	α_i	β_i
1	0	12	0.8	0.5	5
2	1	0.1	-1	0.04	0.5
3	2	1	0.5	2	0.15
4	3	13	-	-	-

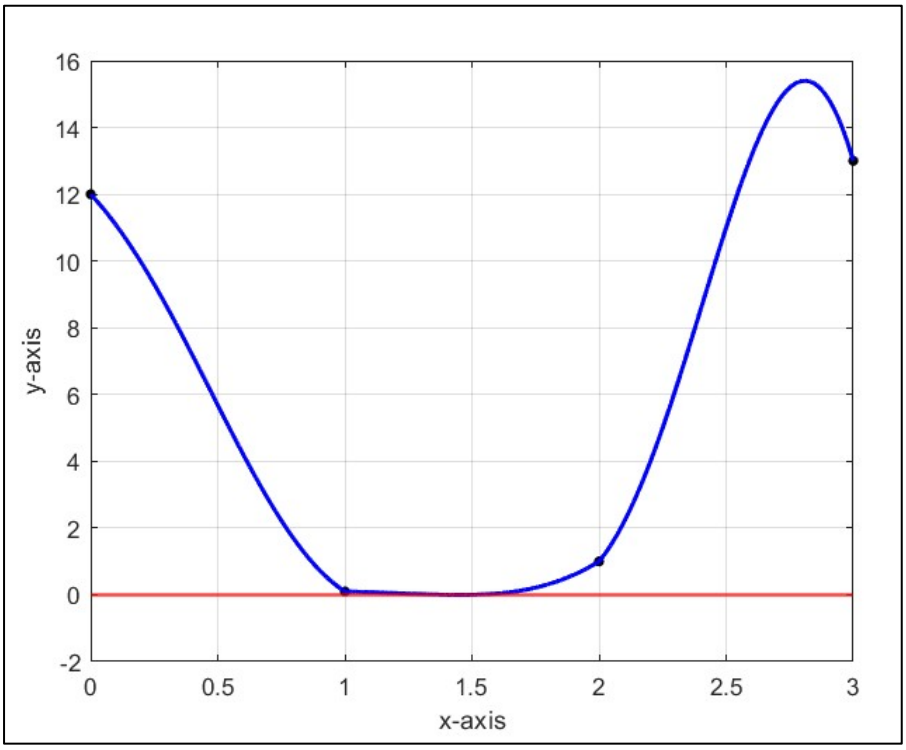


Figure 4.14 Positivity-Preserving Cubic Trigonometric Spline Curve with G^2 Continuity (II)

Figure 4.14 illustrates the performance of the proposed spline when applied to a dataset containing near-zero values in Table 4.5. The interpolating segments are connected under G^2 continuity, and the resulting curve remains entirely non-negative across the whole interval, even though several data points lie exactly on or extremely close to zero axis. This demonstrates the capability of the method to preserve positivity in regions where the data approach zero.

Table 4. 6.
Positive Data Set (V1)

i	x_i	f_i	m_i	n_i
1	0	12	1	0.1
2	1	0.1	0.9	0.001
3	2	1	1	1.5
4	3	13	-	-

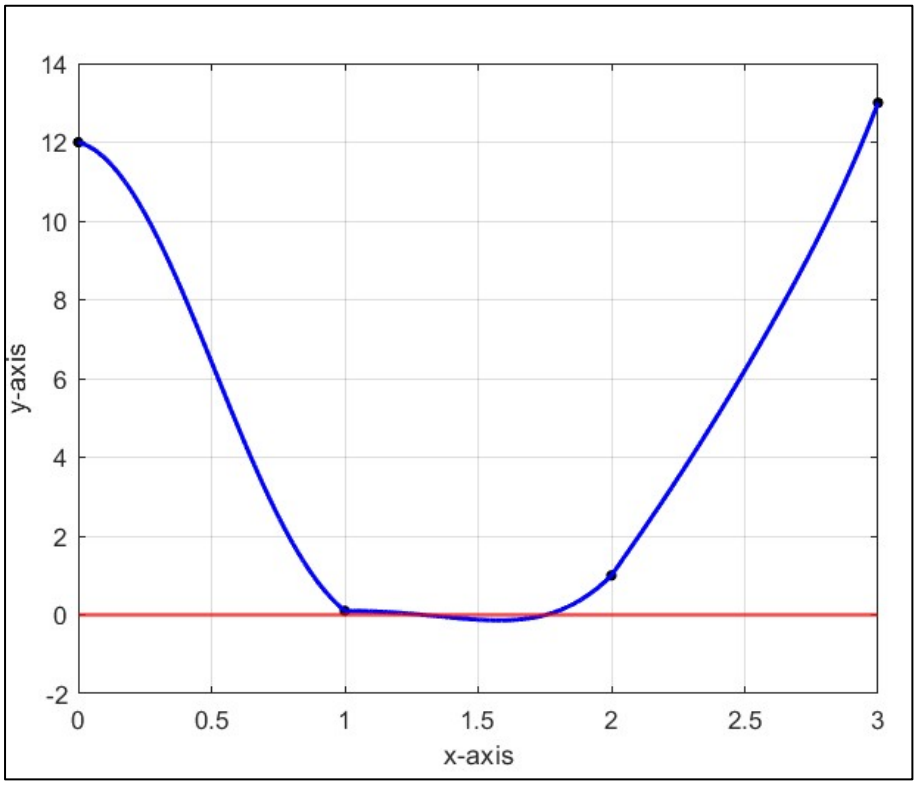


Figure 4.15 Positivity-Preserving Cubic Trigonometric Spline Curve with G^1 Continuity (II)

Similar behaviour is also observed when the segments are connected using G^1 continuity as shown in Figure 4.15. The curves passed through a point lying on the zero axis at $x = 1$ and a near-zero point at $x = 2$, yet it remains entirely non-negative across the interval. In both cases, the spline flattens naturally near the bottom region without producing any negative undershoot. The presence of shape parameter and scalar factor in G^1 and G^2 formulations, provide additional control over the curve and help maintain the non-negativity of the reconstructed shape even in sensitive areas.

In contrast, when the curves are generated using C^1, C^2 continuity conditions, the resulting curves exhibit a high tendency to oscillate below the zero level. The continuity constraints do not provide enough freedom to control the steep variation in slope near the sensitive values. This comparison highlights the advantages of geometric continuity in preserving positivity and controlling shape behaviour in datasets with near-zero points.

4.4 Summary

This chapter has presented the parametric continuity conditions ($C^1$and C^2) and geometric continuity conditions ($G^1$and G^2), together with underlying mathematical formulations. An illustrate example demonstrating how two spline segments are connected under these continuity constraints was also included. A smooth transition from one curve segment to the next can be achieved by applying a choice of continuity criteria at the piecewise parametric curve's connection point. By using parametric and geometric continuity, pleasing continuous curves are formed.

The findings show that although the default positive spline formulation already produces smooth interpolants, geometric continuity offers a higher degree of flexibility. The presence of the shape parameter and the geometric scaling factors in the G^1 and G^2 schemes allow localized control over tension, slope, and curvature. This enhances flexibility enables the spline to follow intricate features of the data more faithfully while preventing unwanted oscillations.

The positivity scheme incorporating both parametric and geometric continuity has been applied to ensure positivity-preserving interpolation using the proposed cubic trigonometric spline. Across Figure 4.10 to 4.15, the spline successfully preserves the essential geometric characteristics of each dataset. By assigning appropriate values to the shape parameter and the geometric scalar factors, the spline consistently retains the non-negative nature of the data specifically referring to Figure 4.14 and Figure 4.15. The ultimate form of the positive interpolating curve is therefore governed by these parameters, which offer the flexibility needed to control the curve's behaviour. In particular, the G^1 and G^2 formulations reliably maintain positivity even when the data contain points that lie exactly on or close to the zero axis. The curve naturally flattens

in these sensitive regions, avoiding negative undershoot and faithfully reflecting the intended shape.

In summary, the cubic trigonometric spline method proposed forms a solid foundation for further development in advanced interpolation and geometric modelling. The capability in preserving positivity, managing complex data patterns, and incorporate various continuity conditions making it suitable for future work such as surface modelling, bigger datasets, and real-time applications requiring smooth, reliable curve generation.

CHAPTER 5
CUBIC TRIGONOMETRIC SURFACES WITH PARAMETRIC AND GEOMETRIC CONTINUITY CONDITIONS

5.1 Introduction

The ability in joining several adjacent curves and surfaces smoothly is said to be significant since designers frequently encounter complicated curves and surfaces that are not defined by a single curve or surface. This section starts with formulating $C^0, C^1, C^2, G^0, G^1, G^2$ cubic trigonometric surface equation. Then, the equation is developed into 3-D objects, and the value of the shape parameter and scaling factor are tested to observe the behaviour of the surface generated. The performance of the equation formulated is evaluated in the last part.

In this chapter, surface equations are formulated using the proposed basis functions of the cubic trigonometric spline. The 3D surfaces are fitted to satisfy the parametric and geometric continuity C^0, C^1, C^2, G^0, G^1 and G^2 condition. The general surface equation is defined as follows:

$$S(u,v) = \sum_{i=0}^{p}\sum_{j=0}^{q} P_{i,j} B_i^p(u) B_j^q(v), \qquad u, u_1 \in [0,1], \tag{5.1}$$

where for $i \in 0,1,\dots,p, j \in 0,1,\dots,q$.

$P_{i,j}$ are the control points.

$B_i^p(u)$ is the general basis function in the direction of u.

$B_j^q(v)$ is the general basis function in the direction of v.

This chapter used the basis function of the cubic trigonometric spline as stated below

$$\begin{cases} B_0(u) = \left(1 - \sin\frac{\pi}{2}u\right)^2\left(1 - m\sin\frac{\pi}{2}u\right) \\ B_1(u) = m\sin\frac{\pi}{2}u\left(1 - \sin\frac{\pi}{2}u\right)^2 - 2\sin\frac{\pi}{2}u\left(\sin\frac{\pi}{2}u - 1\right) \\ B_2(u) = m\cos\frac{\pi}{2}u\left(1 - \cos\frac{\pi}{2}u\right)^2 - 2\cos\frac{\pi}{2}u\left(\cos\frac{\pi}{2}u - 1\right) \\ B_3(u) = \left(1 - \cos\frac{\pi}{2}u\right)^2\left(1 - m\cos\frac{\pi}{2}u\right) \end{cases} \quad (5.2)$$

where the shape parameter $m \in (-2,1]$ is defined for $u \in [0,1]$ in the direction of u and v.

Formulation of C^0, C^1, C^2, G^0, G^1 and G^2 smooth continuity is developed to ensure that the surface patches merge smoothly. There are three alternative techniques to produce smooth continuity at the joint since smooth continuity of surfaces inherits directionality, as mentioned earlier. Perhaps one of them consists of the continuity of two surfaces in the u direction, the continuity of two surfaces in the v direction, and finally, the continuity of one surface in the u direction and another surface in the v direction. The following conditions for each will be covered in this chapter.

Then, the formulas were developed to fit into designing data to form 3D objects, such as combining two surfaces and cylinders. The formations of modelling examples of the cubic trigonometric surface using multiple values of shape parameters and different scalar factor values are presented to clarify the influence of parameters in designing features of developable cubic trigonometric surfaces.

5.2 Development of Cubic Trigonometric Surface Equation

The cubic trigonometric surface function given in (5.2) can be expressed as follows in the u and v direction

$$S(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{i,j}B_i(u)B_j(u). \quad (5.3)$$

Equation (5.3) is represented in matrix form as

$$F(u,v) = [T][A][V][A]^T[U], \tag{5.4}$$

where

$[T]$ is the polynomial matrix in u direction,

$$[T] = \left[1 \quad \sin\left(\frac{\pi}{2}u\right) \quad \sin^2\left(\frac{\pi}{2}u\right) \quad \sin^3\left(p\frac{\pi}{2}u\right) \quad \cos\left(\frac{\pi}{2}u\right) \quad \cos^2\left(\frac{\pi}{2}u\right) \quad \cos^3\left(\frac{\pi}{2}u\right)\right]. \tag{5.5}$$

Since

$$s = \sin\frac{\pi}{2}u, \qquad c = \cos\frac{\pi}{2}u,$$

Equation (5.2) and $[T]$ can be expressed as

$$\begin{aligned} B_0(u) &= (1-s)^2(1-ms), \\ B_1(u) &= m\,s(1-s)^2 - 2\,s(s-1), \\ B_2(u) &= m\,c(1-c)^2 - 2\,c(c-1), \\ B_0(u) &= (1-c)^2(1-mc). \end{aligned}$$

$$[T] = [1, s, s^2, s^3, c, c^2, c^3].$$

Expand $B_0(u)$,

$$\begin{aligned} B_0(u) &= (1-2s+s^2)(1-ms) \\ &= 1 - ms - 2s + 2ms^2 + s^2 - ms^3 \\ &= 1 - (m+2)s + (1+2m)s^2 - ms^3. \end{aligned}$$

So, the coefficients (in order $[1, s, s^2, s^3, c, c^2, c^3]$) are

$$\text{col}\,(B_0) = \begin{bmatrix} 1 \\ -(m+2) \\ 1+2m \\ -m \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Expand $B_1(u)$

$$\begin{aligned} B_1(u) &= m\,s(1-s)^2 - 2\,s(s-1) \\ &= ms(1-s)^2 + 2s(1-s) \\ &= s(1-s)(m(1-s)+2) \\ B_1(u) &= s[(m+2) - (2m+2)s + ms^2] \\ &= (m+2)s - 2(m+1)s^2 + ms^3 \end{aligned}$$

So, the coefficient column is

$$\text{col}\,(B_1) = \begin{bmatrix} 0 \\ (m+2) \\ -2(m+1) \\ m \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Expand $B_2(u)$

$$\begin{aligned} B_2(u) &= m\,c(1-c)^2 - 2\,c(c-1) \\ &= mc(1-c)^2 + 2c(1-c) \\ &= c(1-c)(m(1-c)+2) \\ B_1(u) &= c[(m+2) - (2m+2)c + mc^2] \\ &= (m+2)c - 2(m+1)c^2 + mc^3 \end{aligned}$$

So, the coefficient column is

$$\text{col}\,(B_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m+2 \\ -2(m+1) \\ m \end{bmatrix}.$$

Expand $B_3(u)$

$$\begin{aligned} B_3(u) &= (1-2c+c^2)(1-mc) \\ &= 1 - mc - 2c + 2mc^2 + c^2 - mc^3 \\ &= 1 - (m+2)c + (1+2m)c^2 - mc^3 \end{aligned}$$

So, the coefficients (in order $[1, s, s^2, s^3, c, c^2, c^3]$) are

$$\text{col}\,(B_0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -(m+2) \\ 1+2m \\ -m \end{bmatrix}.$$

Assemble $[A]$

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ -(m+2) & m+2 & 0 & 0 \\ 1+2m & -2(m+1) & 0 & 0 \\ -m & m & 0 & 0 \\ 0 & 0 & m+2 & -(m+2) \\ 0 & 0 & -2(m+1) & 1+2m \\ 0 & 0 & m & -m \end{bmatrix} \tag{5.6}$$

$[V]$ is the control point matrix, $[V] = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 \\ V_5 & V_6 & V_7 & V_8 \\ V_9 & V_{10} & V_{11} & V_{12} \\ V_{13} & V_{14} & V_{15} & V_{16} \end{bmatrix}$ (5.7)

$[U]$ is the polynomial matrix in v direction,

$$[U] = \left[1 \quad \sin\left(\frac{\pi}{2}v\right) \quad \sin^2\left(\frac{\pi}{2}v\right) \quad \sin^3\left(\frac{\pi}{2}v\right) \quad \cos\left(\frac{\pi}{2}v\right) \quad \cos^2\left(\frac{\pi}{2}v\right) \quad \cos^3\left(\frac{\pi}{2}v\right)\right]^T \tag{5.8}$$

The surface equation is manually expanded and calculated to demonstrate that it was successfully obtained, for *i*=0,1,2,3 and *j*=0,1,2,3. However, only expansion for *i*=0 and *j*=0 will be shown below as the proof for other pairs will be lengthy and repetitive.

Basis function in u- direction:

$$
\begin{aligned}
B_0(u) &= \left(1-\sin\frac{\pi}{2}u\right)^2\left(1-m\sin\frac{\pi}{2}u\right);\\
B_1(u) &= m\sin\frac{\pi}{2}u\left(1-\sin\frac{\pi}{2}u\right)^2-2\sin\frac{\pi}{2}u\left(\sin\frac{\pi}{2}u-1\right);\\
B_2(u) &= m\cos\frac{\pi}{2}u\left(1-\cos\frac{\pi}{2}u\right)^2-2\cos\frac{\pi}{2}u\left(\cos\frac{\pi}{2}u-1\right);\\
B_3(u) &= \left(1-\cos\tfrac{\pi}{2}u\right)^2\left(1-m\cos\tfrac{\pi}{2}u\right);
\end{aligned}
\tag{5.9}
$$

Basis function in v- direction:

$$
\begin{aligned}
B_0(v) &= \left(1-\sin\frac{\pi}{2}v\right)^2\left(1-m\sin\frac{\pi}{2}v\right);\\
B_1(v) &= m\sin\frac{\pi}{2}v\left(1-\sin\frac{\pi}{2}v\right)^2-2\sin\frac{\pi}{2}v\left(\sin\frac{\pi}{2}v-1\right);\\
B_2(v) &= m\cos\frac{\pi}{2}v\left(1-\cos\frac{\pi}{2}v\right)^2-2\cos\frac{\pi}{2}v\left(\cos\frac{\pi}{2}v-1\right);\\
B_3(v) &= \left(1-\cos\tfrac{\pi}{2}v\right)^2\left(1-m\cos\tfrac{\pi}{2}v\right);
\end{aligned}
\tag{5.10}
$$

A single surface is generated by running the code in MATLAB. The underlying surface is defined by a 4×4 control net containing the following control points:

$$
\begin{gathered}
(0,0,0),(1,0,1),(2,0,2),(3,0,0),\\
(0,1,1),(1,1,3),(2,1,4),(3,1,2),\\
(0,2,2),(1,2,4),(2,2,3),(3,2,1),\\
(0,3,0),(1,3,2),(2,3,1),(3,3,0).
\end{gathered}
$$

These points form the control structure from which the surface is constructed. The subsequent parts of this chapter illustrate further developments of additional surfaces.

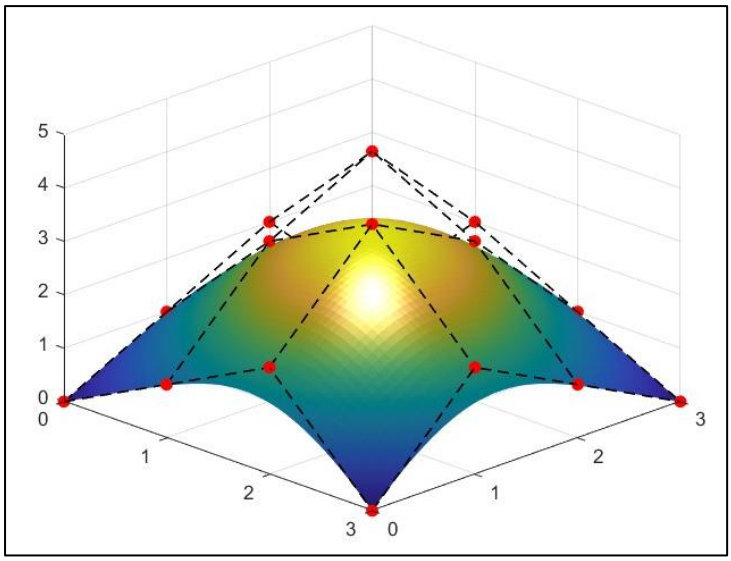


Figure 5. 1 Single Surface of Cubic Trigonometric Bezier

5.3 Continuity Condition

Given that there are two cubic trigonometric surfaces required to achieve the parametric and geometric continuity conditions:

$$\begin{cases} S_1(u,v\,;\,m_1,m_2) = \sum_{i=0}^{p_1}\sum_{j=0}^{q_1} B_{i,p_1}(u;m_1)B_{i,q_1}(v;m_2)P_{i,j}^1 \\ S_2(u,v\,;\,m_1^*,m_2^*,) = \sum_{i=0}^{p_2}\sum_{j=0}^{q_2} B_{i,p_2}(u;m_1^*)B_{i,q_2}(v;m_2^*)\,P_{i,j}^2 \end{cases} \tag{5.11}$$

where $-2 < m_1, m_2, m_1^*, m_2^* \leq 1$, $P_{i,j}^1 (i = 0,1,\dots,k_1 = 3; j = 0,1,\dots,l_1 = 3)$ and $P_{i,j}^2 (i = 0,1,\dots,k_2 = 3; j = 0,1,\dots,l_2 = 3)$ are control points of the surfaces $S_1(u,v\,;\,m_1,m_2)$ and $S_2(u,v\,;\,m_1^*,m_2^*,)$, respectively. Since surfaces have directivity, which distinguishes them from curves, so there exists continuity in three kinds of directions namely u-direction, u- and v- direction, and v-direction.

The cubic trigonometric function has the following terminal properties. Terminal properties can be described as the boundary between the first and second planes, where the end line of the first plane is equal to the initial line of the second plane. The first and second derivatives with respect to u are represented by the single and double primes, respectively.

Terminal Properties

$$\begin{aligned} &S(0;m) = P_4, \qquad S(1;m) = P_3, \\ &S'(0;m) = (m+2)P_5 - (m+2)P_4, \\ &S'(1;m) = (m+2)P_3 - (m+2)P_2, \\ &S''(0;m) = 2P_6 - (4m+4)P_5 + (4m+2)P_4, \\ &S''(1;m) = 2P_1 - (4m+4)P_2 + (4m+2)P_3 \end{aligned} \tag{5.12}$$

5.3.1 Continuity in the u-Direction and v −Direction

5.3.1.1 C^1 Continuity in the u-Direction

Theorem 5.1: Let $S_1(u,v\,;\,m_1,m_2)$ and $S_2(u,v\,;\,m_1^*,m_2^*,)$ be two adjacent cubic trigonometric surface patches defined in (5.11).

Condition 1: Equal Shape Parameter, *m*

$$\begin{cases} m_1 = m_1^* \\ P_4 = P_3 \\ P_5 = 2P_3 - P_2 \end{cases} \tag{5.13}$$

Condition 2: Distinct Shape Parameters, *m*

$$\begin{cases} m_1 = m_1^* \neq\ m_2 = m_2^* \\ P_4 = P_3 \\ P_5 = \dfrac{(m_2+2)P_3 - (m_1+2)P_3 - (m_2+2)P_2}{(m_1+2)} \end{cases} \tag{5. 14}$$

Surfaces S_1 and S_2 achieve C^1 smooth continuity in the u direction at the common boundary.

Proof. The two surfaces, $S_1(u,v\,;\,m_1,m_2)$ and $S_2(u,v\,;\,m_1^*,m_2^*,)$ will achieve C^1 continuity if and only if they satisfy the C^0 continuity condition. The surfaces will achieve C^0 when

$$S_2(u,0;m_1^*,m_2^*) = S_1(u,1;m_1,m_2). \tag{5.15}$$

Based on the terminal properties of the cubic trigonometric surface in (5.12), the following equation can be obtained

$$\sum_{i=0}^{p_2} B_{i,p_2}(u; m_1^*) P_4 = \sum_{i=0}^{p_1} B_{i,p_1}(u; m_1) P_3. \tag{5.16}$$

Comparison of the coefficients allows for the simplification of the equation, $m_1 = m_2$, as follows

$$P_4 = P_3. \tag{5.17}$$

Thus, the two patches' posses C^0 continuity.

Based on the concept of C^1 continuity, the two surfaces are must possess a derivative that is continuous across the boundary. Thus, it is necessary for them to meet the following requirements:

$$\begin{aligned} &\frac{\partial}{\partial v} S_2(u, 0; m_1^*, m_2^*) \times \frac{\partial}{\partial u} S_2(u, 0; m_1^*, m_2^*) \\ &\qquad = \frac{\partial}{\partial v} S_1(u, 1; m_1, m_2) \times \frac{\partial}{\partial u} S_1(u, 1; m_1, m_2). \end{aligned} \tag{5.18}$$

The equation (5.17) can be expressed in a simplified form as:

$$\frac{\partial}{\partial v} S_2(u, 0; m_1^*, m_2^*) = \frac{\partial}{\partial v} S_1(u, 1; m_1, m_2). \tag{5.19}$$

Using endpoint terminal property derived in (5.12),

$$\frac{\partial}{\partial v} S_2(u, 0; m_1^*, m_2^*) = \sum_{i=0}^{p_2} B_{i,p_2}(u; m_1^*)(m_1^* + 2)(P_5 - P_4) \tag{5.20}$$

$$\frac{\partial}{\partial v} S_1(u, 1; m_1, m_2) = \sum_{i=0}^{p_1} B_{i,p_1}(u; m_2)(m_2 + 2)(P_3 - P_2). \tag{5.21}$$

Substituting equations (5.20) and (5.21) into (5.19), we get

$$\sum_{i=0}^{p_2} B_{i,p_2}(u; m_1^*)(m_1^* + 2)(P_5 - P_4) = \sum_{i=0}^{p_1} B_{i,p_1}(u; m_2)(m_2 + 2)(P_3 - P_2). \tag{5.22}$$

Condition 1: Equal Shape Parameters, m

If the adjacent patches satisfy

$$m_1 = m_1^* = m_2 = m, \qquad p_1 = p_2,$$

then (5.21) simplifies to

$$(m + 2)(P_5 - P_4) = (m + 2)(P_3 - P_2)$$

Using $P_4 = P_3$, we obtain

$$P_5 - P_3 = P_3 - P_2,$$

and consequently,

$$P_5 = 2P_3 - P_2. \tag{5.23}$$

Condition 2: Distinct Shape Parameters, m

If the patches do not share the same parameter, then (5.22) becomes

$$(m_1 + 2)P_5 - (m_1 + 2)P_4 = (m_2 + 2)P_3 - (m_2 + 2)P_2$$
$$(m_1 + 2)P_5 - (m_1 + 2)P_3 = (m_2 + 2)P_3 - (m_2 + 2)P_2$$

Solving for P_5 yields

$$P_5 = \frac{(m_2 + 2)P_3 - (m_1 + 2)P_3 - (m_2 + 2)P_2}{(m_1 + 2)} \tag{5.24}$$

Both positional and first-order derivative constraints must be satisfied for the adjacent cubic trigonometric patches to meet with C^1 continuity in the u -direction. The proof for C^1 continuity in the v-direction and uv -direction proceeds analogously.

5.3.1.2 G^1 continuity in the u-Direction

Theorem 5.2: When two adjacent cubic trigonometric surface patches (5.11) fulfil all the following conditions,

Condition 1: Equal Shape Parameter, *m*

$$\begin{cases} m_1 = m_1^* \\ P_4 = P_3 \\ P_5 = \alpha(P_3 - P_2) + P_3 \end{cases} \tag{5.25}$$

Condition 2: Distinct Shape Parameters, *m*

$$\begin{cases} m_1 = m_1^* \neq m_2 = m_2^* \\ P_4 = P_3 \\ P_5 = \dfrac{\alpha(m_2+2)P_3 - \alpha(m_2+2)P_2 + (m_1+2)P_3}{(m_1+2)} \end{cases} \tag{5.26}$$

surfaces S_1 and S_2 achieve G^1 smooth continuity in the u direction at the common boundary, where $\alpha > 0$ is a real constant.

Proof. In order for the two neighbouring generalized cubic trigonometric surfaces, $S_1(u, v\,;\, m_1, m_2)$ and $S_2(u, v\,;\, m_1^*, m_2^*,)$, attain smooth continuity in the u and v directions, it is necessary to initially meet G^0 continuity in (5.15). Next, the two developable surfaces have to fulfil the requirements listed below for G^1 continuity to be guaranteed at the joint:

$$\begin{cases} S_2(0) = S_1(1), \\ S_2'(0) = \alpha S_1'(1). \end{cases} \tag{5.27}$$

Equation (5.27) is implemented based on the cubic trigonometric surface's terminal properties in (5.12). The next criterion to be taken into account is that the two cubic trigonometric surface patches must meet normal vector direction continuity at the common boundary.

$$\frac{\partial}{\partial v}S_2(u,0;m_1^*,m_2^*)\times\frac{\partial}{\partial u}S_2(u,0;m_1^*,\beta_1^*,m_2^*)$$
$$=\alpha(u)\frac{\partial}{\partial v}S_1(u,1;m_1,m_2)\times\frac{\partial}{\partial u}S_1(u,1;m_1,m_2) \tag{5.28}$$

where $\alpha(u)$ is a scaling vector between the normal vector such that $\alpha(u)>0$. The simplified equation above can be written as:

$$\frac{\partial}{\partial v}S_2(u,0;m_1^*,m_2^*)=\alpha(u)\frac{\partial}{\partial v}S_1(u,1;m_1,m_2). \tag{5.29}$$

Using endpoint terminal property derived from the equation (5.12)

$$\frac{\partial}{\partial v}S_2(u,0;m_1^*,m_2^*)=\sum_{i=0}^{p_2}B_{i,p_2}(u;m_1^*)(m_1^*+2)(P_5-P_4) \tag{5.30}$$

$$\frac{\partial}{\partial v}S_1(u,1;m_1,m_2)=\sum_{i=0}^{p_1}B_{i,p_1}(u;m_2)(m_2+2)(P_3-P_2). \tag{5.31}$$

Substituting equations (5.30) and (5.31) into (5.29), we get

$$\sum_{i=0}^{p_2}B_{i,p_2}(u;m_1^*)(m_1^*+2)(P_5-P_4)$$
$$=\alpha\sum_{i=0}^{p_1}B_{i,p_1}(u;m_2)(m_2+2)(P_3-P_2). \tag{5.32}$$

Condition 1: Equal Shape Parameter, *m*

If

$$m_1=m_1^*=m_2=m,\qquad p_1=p_2,$$

Then (5.32) reduces to

$$(m+2)(P_5-P_4)=\alpha(m+2)(P_3-P_2)$$
$$P_5=\alpha(P_3-P_2)+P_3. \tag{5.33}$$

Condition 2: Distinct Shape Parameters, *m*

If the two patches use different parameters,

$$m_1 = m_1^* \neq m_2 = m_2^*,$$

Then the continuity condition becomes

$$(m_1 + 2)(P_5 - P_4) = \alpha(m_2 + 2)(P_3 - P_2)$$

Hence, the condition for G^1 continuity is

$$P_5 = \frac{\alpha(m_2 + 2)P_3 - \alpha(m_2 + 2)P_2 + (m_1 + 2)P_3}{(m_1 + 2)}. \tag{5.34}$$

Therefore, the equation represents the G^1 smooth continuity conditions of two surfaces, $S_1(u, v\,;\, m_1, m_2)$ and $S_2(u, v\,;\, m_1^*, m_2^*,)$ in the u direction. The theorem has been proven.

Similarly, the G^1continuity condition in the v-direction is obtained by applying the u-direction and uv -direction geometric continuity argument from Theorem 5.2.

5.3.1.3 C^2 continuity in the u-Direction

Theorem 5.3: When two cubic trigonometric Bezier surface patches (5.11) fulfil the following conditions,

Condition 1: Equal Shape Parameter, *m*

$$\begin{cases} m_1 = m_1^* \\ P_4 = P_3 \\ P_5 = 2P_3 - P_2 \\ P_6 = P_1 - (4m + 4)P_2 + (4m + 4)P_3 \end{cases} \tag{5.35}$$

Condition 2: Distinct Shape Parameters, *m*

$$\begin{cases} m_1 = m_1^* \neq m_2 = m_2^* \\ P_4 = P_3 \\ P_5 = \dfrac{(m_2 + 2)P_3 - (m_1 + 2)P_3 - (m_2 + 2)P_2}{(m_1 + 2)} \\ P_6 = P_1 + \dfrac{(4m_2 + 2)P_3 - (4m_2 + 4)P_2 - (4m_1 + 2)P_3 - (4m_1 + 4)P_5}{2} \end{cases} \tag{5.36}$$

at the common boundary, the u directions of S_1 and S_2 both attain C^2 smooth continuity.

Proof. C^2 continuity could be achieved by the two surfaces, $S_1(u, v\,;\, m_1, m_2)$ and $S_2(u, v\,;\, m_1^*, m_2^*,)$ in the u direction at the joint by satisfying C^0 and C^1 continuity conditions as stated in (5.15) and (5.19). For smooth C^1 continuity, it is necessary for the two surfaces to possess identical normal curvature at every point along their shared boundary. Next, both surfaces can achieve C^2 continuity if

$$\frac{\partial^2}{\partial v^2} S_2(u, 0; m_1^*, m_2^*) = \frac{\partial^2}{\partial v^2} S_1(u, 1; m_1, m_2). \tag{5.37}$$

Using endpoint terminal property derived from the equation in (5.12)

$$\frac{\partial^2}{\partial v^2} S_2(u, 0; m_1^*, m_2^*) = \sum_{i=0}^{p_2} B_{i,p_2}(u; m_1^*)[2P_6 - (4m_1^* + 4)P_5 + (4m_1^* + 2)P_4] \tag{5.38}$$

$$\frac{\partial^2}{\partial v^2} S_1\left(u, \frac{\pi}{2}; m_1, m_2\right) = \sum_{i=0}^{p_1} B_{i,p_1}(u; m_2)[2P_1 - (4m_2 + 4)P_2 + (4m_2 + 2)P_3] \tag{5.39}$$

Condition 1: Equal Shape Parameter, *m*

If

$$m_1 = m_1^* = m_2 = m, \qquad p_1 = p_2,$$

Then

$$\begin{aligned} &(4m + 2)P_4 - (4m + 4)P_5 + 2P_6 \\ &= (4m + 2)P_3 - (4m + 4)P_2 + 2P_1 \\ &-2(4m + 4)P_3 + 2(4m + 4)P_2 + 2P_6 = 2P_1 \end{aligned}$$

Thus,

$$P_6 = P_1 - (4m + 4)P_2 + (4m + 4)P_3. \tag{5.40}$$

Condition 2: Distinct Shape Parameters, *m*

If the two patches use different parameters,

$$m_1 = m_1^* \neq m_2 = m_2^*,$$

Then the continuity condition becomes

$$\begin{aligned}
&(4m_1+2)P_4 - (4m_1+4)P_5 + 2P_6 \\
&= (4m_2+2)P_3 - (4m_2+4)P_2 + 2P_1 \\
2P_6 &= 2P_1 + (4m_2+2)P_3 - (4m_2+4)P_2 - (4m_1+2)P_3 \\
&\quad - (4m_1+4)P_5
\end{aligned}$$

Hence,

$$P_6 = P_1 + \frac{(4m_2+2)P_3 - (4m_2+4)P_2 - (4m_1+2)P_3 - (4m_1+4)P_5}{2} \tag{5.41}$$

Therefore, the equation represents the C^2 smooth continuity conditions of two surfaces, $S_1(u, v\,;\, m_1, m_2)$ and $S_2(u, v\,;\, m_1^*, m_2^*,)$ in the u direction. The theorem has been proven.

The C^2continuity in v-direction and uv -direction of surfaces is demonstrated using a similar approach as the continuity as described in Theorem 5.3. By considering the second partial derivatives with respect to v, the required smoothness conditions across the boundary are satisfied.

5.3.1.4 G^2 continuity in the u-Direction

Theorem 5.4: When two cubic trigonometric Bezier surface patches (5.11) fulfil the following conditions

Condition 1: Equal Shape Parameter, *m*

$$\begin{gathered} m_1 = m_1^* \\ P_4 = P_3 \\ P_5 = \alpha(P_3 - P_2) + P_3 \\ P_6 = \left(2\alpha^2 m + \alpha^2 + \frac{\beta m}{2} + \beta + 1 + 2m\alpha + 2\alpha\right) P_3 - \\ \left(2m\alpha + 2\alpha + 2\alpha^2 m + 2\alpha^2 + \frac{\beta m}{2} + \beta\right) P_2 + \alpha^2 P_1 \end{gathered} \tag{5.42}$$

Condition 2: Distinct Shape Parameters, *m*

$$\begin{cases} m_1 = m_1^* \neq\ m_2 = m_2^* \\ P_4 = P_3 \\ P_5 = \dfrac{\alpha(m_2+2)P_3 - \alpha(m_2+2)P_2 + (m_1+2)P_3}{(m_1+2)} \\ (4m_1+4)P_3 - (4m_1+2)P_3 + \\ P_6 = \dfrac{\alpha^2[2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3 + \beta(m_2+2)P_3 - (m_2+2)P_2]}{2} \end{cases} \tag{5.43}$$

surfaces S_1 and S_2 achieves G^2 continuity in the u directions at the common boundary where $\alpha > 0$ and $\beta \in \mathbb{R}$.

Proof. G^2 continuity could be achieved by the two surfaces, $S_1(u, v\ ;\ m_1, m_2)$ and $S_2(u, v\ ;\ m_1^*, m_2^*,)$ in the u direction at the joint by satisfying G^0 and G^1 continuity conditions first. The conditions are stated in (5.15) and (5.19). Further, both surfaces can meet G^2 continuity when

$$\frac{\partial^2}{\partial v^2} S_2(u, 0; m_1^*, m_2^*) = \alpha^2 \frac{\partial^2}{\partial v^2} S_1(u, 1; m_1, m_2) + \beta \frac{\partial}{\partial v} S_1(u, 1; m_1, m_2). \tag{5.44}$$

Using endpoint terminal property derived from the equation (5.12)

$$\frac{\partial^2}{\partial v^2} S_2(u, 0; m_1^*)$$

$$= \sum_{i=0}^{p_2} B_{i,P_2}(u; m_1^*)[2P_6 - (4m_1^* + 4)P_5 + (4m_1^* + 2)P_4]$$

$$\alpha^2 \frac{\partial^2}{\partial v^2} S_1(u,1;m_2) + \beta \frac{\partial}{\partial v} S_1(u,1;m_2) = \alpha^2 \sum_{i=0}^{p_2} B_{i,p_2}(u;m_2)[2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3] + \beta \sum_{i=0}^{p_2} B_{i,p_2}(u;m_2)[(m_2+2)P_3 - (m_2+2)P_2]. \quad (5.45)$$

Condition 1: Equal Shape Parameter, *m*

If

$$m_1 = m_1^* = m_2 = m, \qquad p_1 = p_2,$$

Then

$$P_6 = \left(2\alpha^2 m + \alpha^2 + \frac{\beta m}{2} + \beta + 1 + 2m\alpha + 2\alpha\right) P_3 - \left(2m\alpha + 2\alpha + 2\alpha^2 m + 2\alpha^2 + \frac{\beta m}{2} + \beta\right) P_2 + \alpha^2 P_1. \quad (5.46)$$

Condition 2: Distinct Shape Parameters, *m*

If the two patches use different parameters,

$$m_1 = m_1^* \neq m_2 = m_2^*,$$

Then the continuity condition becomes

$$\begin{aligned} 2P_6 - (4m_1+4)P_5 + (4m_1+2)P_4 &= \alpha^2[2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3] \\ &\quad + \beta[(m_2+2)P_3 - (m_2+2)P_2] \\ 2P_6 = (4m_1+4)P_3 - (4m_1+2)P_3 &+ \alpha^2[2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3 + \beta(m_2+2)P_3 \\ &\quad - (m_2+2)P_2] \end{aligned}$$

Hence,

$$P_6 = \frac{(4m_1+4)P_3 - (4m_1+2)P_3 + \alpha^2[2P_1 - (4m_2+4)P_2 + (4m_2+2)P_3 + \beta(m_2+2)P_3 - (m_2+2)P_2]}{2} \quad (5.47)$$

Therefore, the equation represents the G^2 smooth continuity conditions of two surfaces, $S_1(u,v\,;\,m_1,m_2)$ and $S_2(u,v\,;\,m_1^*,m_2^*,)$ in the u direction. The theorem has been proven.

The G^2 continuity in the v-direction of surfaces is demonstrated using a similar approach as the continuity in the u -direction and uv -direction, as described in Theorem 5.4.

5.4 Designing Examples of Cubic Trigonometric Surfaces

In order to demonstrate the efficacy of the approach developed in Section 5.3; by using the surface equation of cubic trigonometric spline, many kinds of surface have been constructed. The developable single surface, two surfaces, and cylinder surfaces are formed and described. The impacts of shape parameters and scalar parameters on the developable surface also displayed.

5.4.1 Single Surfaces Generated by Cubic Trigonometric Surface Equation

The surface equation derived in (5.11) is built into the code as a matrix, resulting in the single surfaces shown below. The formation of single surfaces is evaluated with different values of the shape parameter, m, within the range $(-2,1]$ and also out of the range. The control points are depicted as red circles on the polylines. The surfaces' behaviour is assessed either inside or outside the control polygon, which is defined by the red polylines. The surfaces can be viewed from various viewpoints, and without having to redetermine the control planes, the designer can manually modify the surfaces' shapes by altering the values of their shape parameter.

Figure 5.3 and Figure 5.4 below present the behaviour of a single surface tested with different values of the shape parameter.

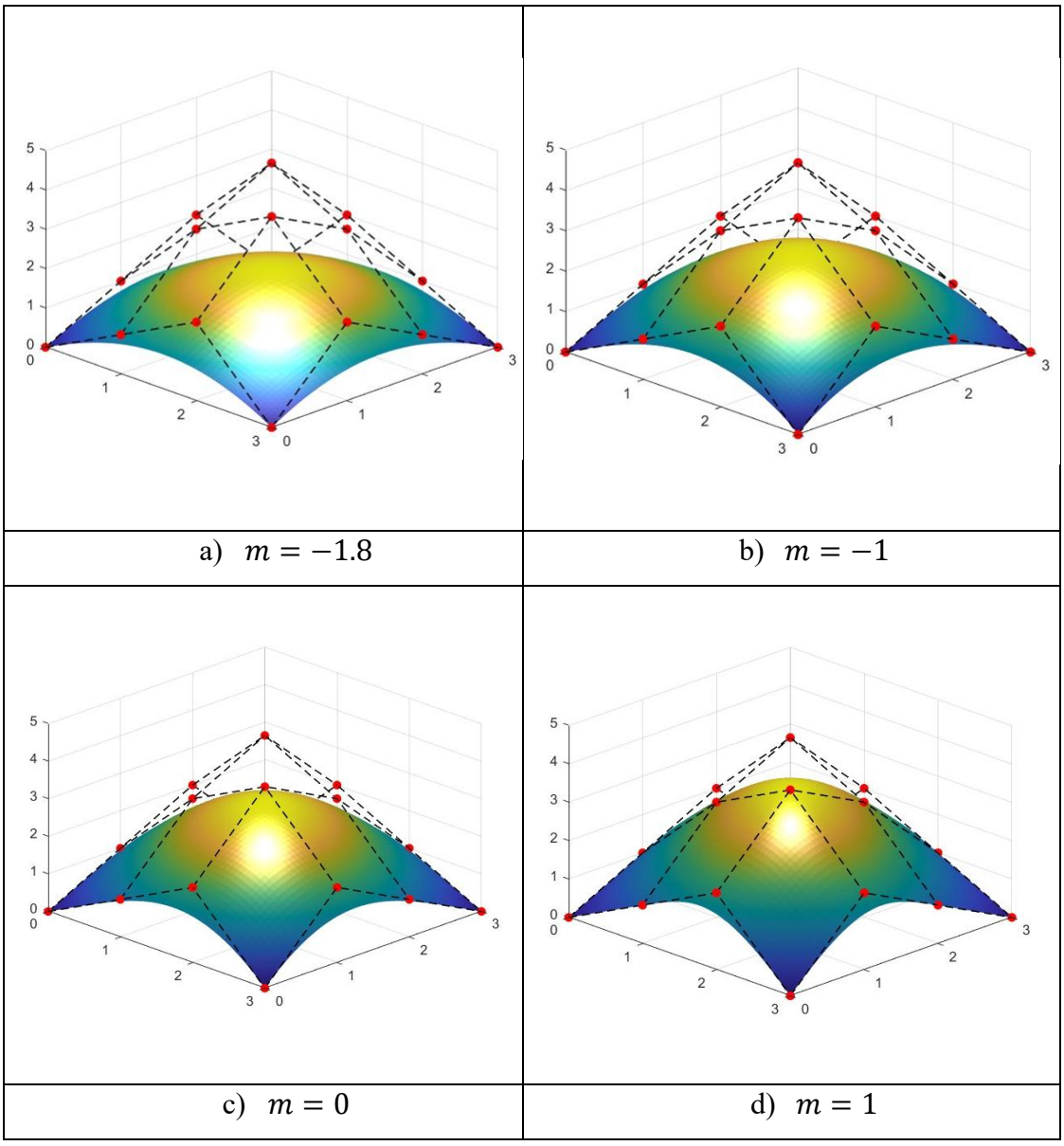


Figure 5. 2 Single Surface Formed by a Cubic Trigonometric Surface Equation with Various Shape Parameters within the Range

Figure 5.2 shows that all the single surfaces that developed with the shape parameter $m = -1.8, m = -1, m = 0$, and $m = 1$, lie inside the control net. When the values of $m < 0$, the surfaces are formed away from the control net, as the negative values of m introduces stronger oscillation because it reverses the influence of trigonometric terms. A neutral or relatively smooth surface are formed near to the control net when using $m = 0$. The surface generated becomes tighter and smoother as the shape parameter value increases because of the trigonometric blending functions increasing the curvature around the control points. Increasing values of m makes the surface "pulled in" toward the control net.

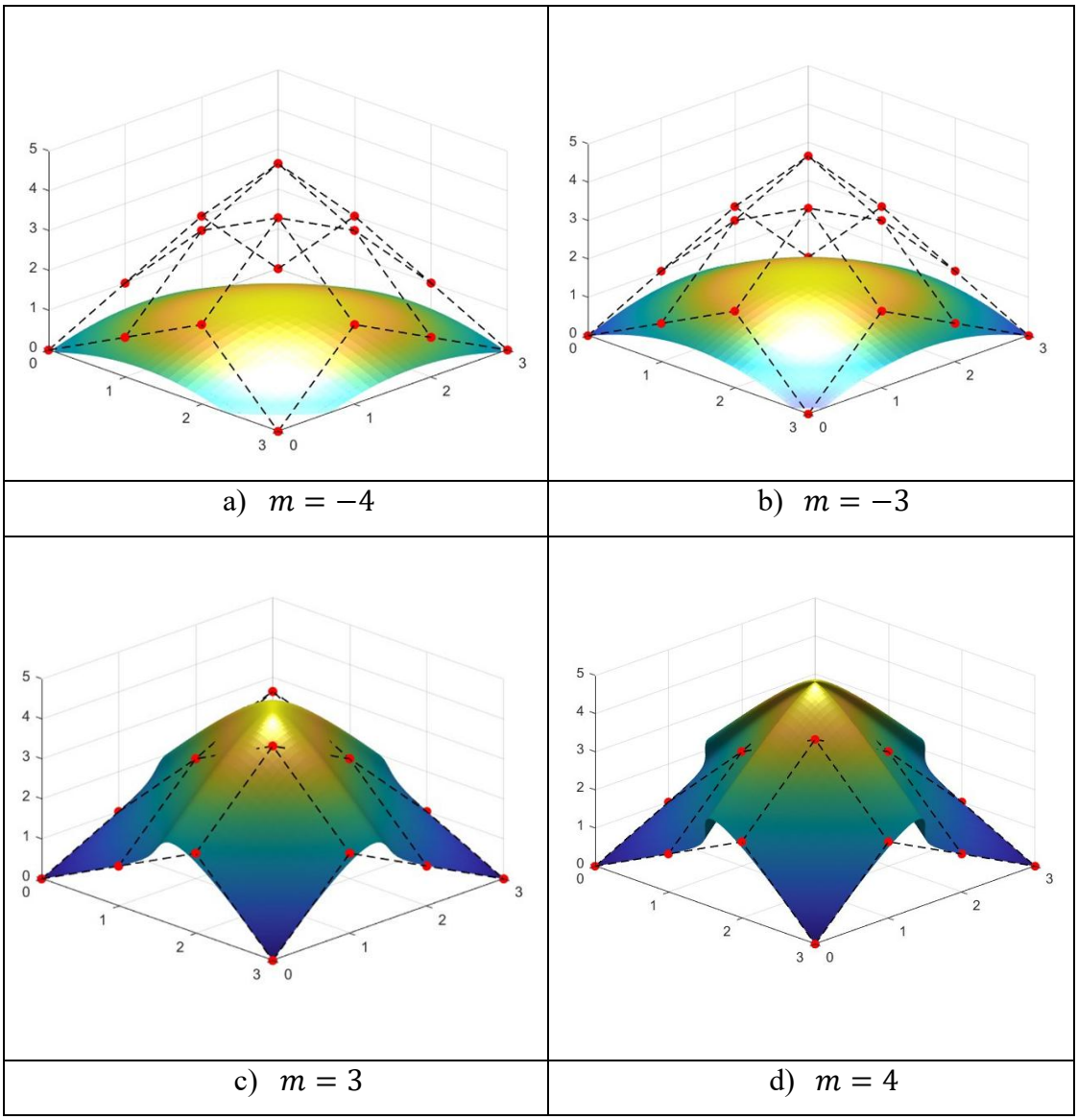


Figure 5. 3 Single Surface Formed by a Cubic Trigonometric Surface Equation with Various Shape Parameters Out of the Range

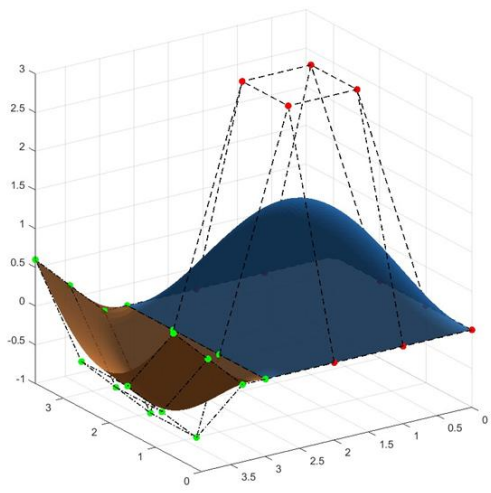
Surfaces with the shape parameters $m = -4, m = -3, m = 3$, and $m = 4$, as generated in Figure 5.3, are slightly generated out of the range provided surfaces that lie outside the control polygon. The prerequisites for generating cubic trigonometric curves and surfaces are satisfied in the range $(-2,1]$, which is stated. Meanwhile, when shape parameters beyond the range $(-2,1]$ are selected the curves and surfaces obtained frequently originate from the control mesh polygon.

At $m = -4$ and $m = -3$, the surface flips and stretches and some parts are folding and dipping below the control net. The shape becomes unstable and self-intersecting. Meanwhile, when the values of $= 3$ and $m = 4$ are used, the surfaces inverted and fold in on itself and the surface becomes very unstable. This happens

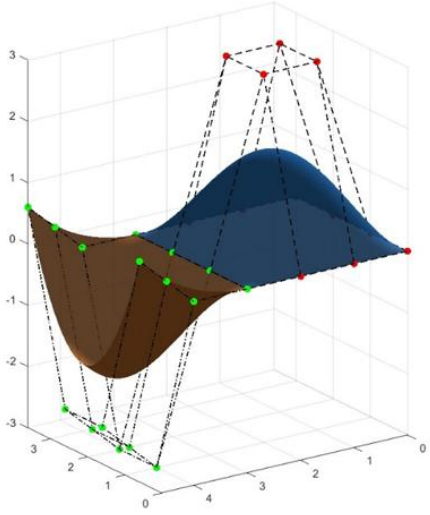
because the basis functions are not normalized like Bezier basis functions (which always sum to 1 and remain positive. Surfaces which generated with larger m will lose the smoothness and boundness, and no longer form a stable interpolation surface.

5.4.2 Two Surfaces Generated from Cubic Trigonometric Bezier Surface Equation in u and v Direction

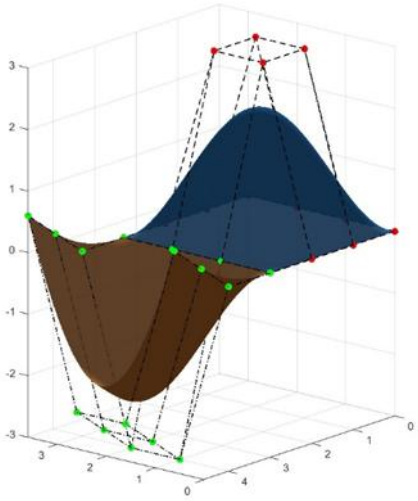
In Figure 5.4- Figure 5.7, two surfaces were generated by combining two patches of surfaces with continuity in C^1, C^2, G^1 and G^2 continuity. The shape parameter, m is appropriately selected within the range $(-2,1]$ provided and for scalar factor $\alpha > 0$, β is any real number.



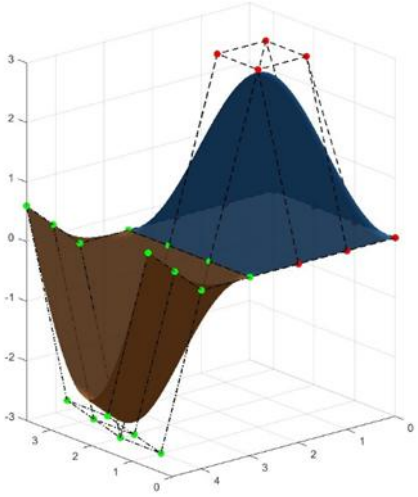
a) $m_1 = -1, m_2 = 1$



b) $m_1 = m_2 = -1$

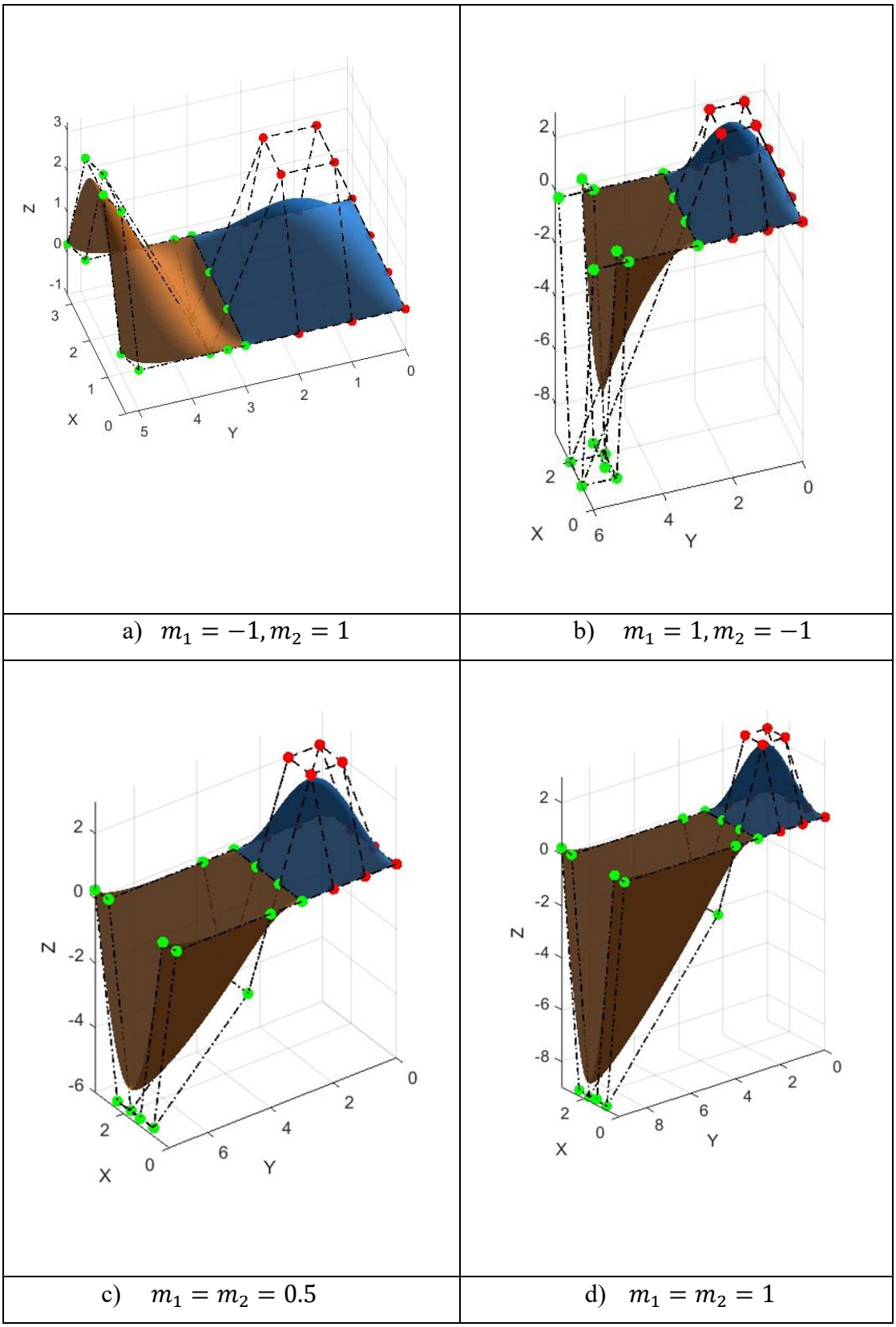


c) $m_1 = m_2 = 0$



d) $m_1 = m_2 = 1$

Figure 5. 4 Two Surfaces Formed by C^1 Cubic Trigonometric Surface Equation with Various Shape Parameters



a) $m_1 = -1, m_2 = 1$ b) $m_1 = 1, m_2 = -1$

c) $m_1 = m_2 = 0.5$ d) $m_1 = m_2 = 1$

Figure 5. 5 Two Surfaces Formed by C^2 Cubic Trigonometric Surface Equation with Various Shape Parameters

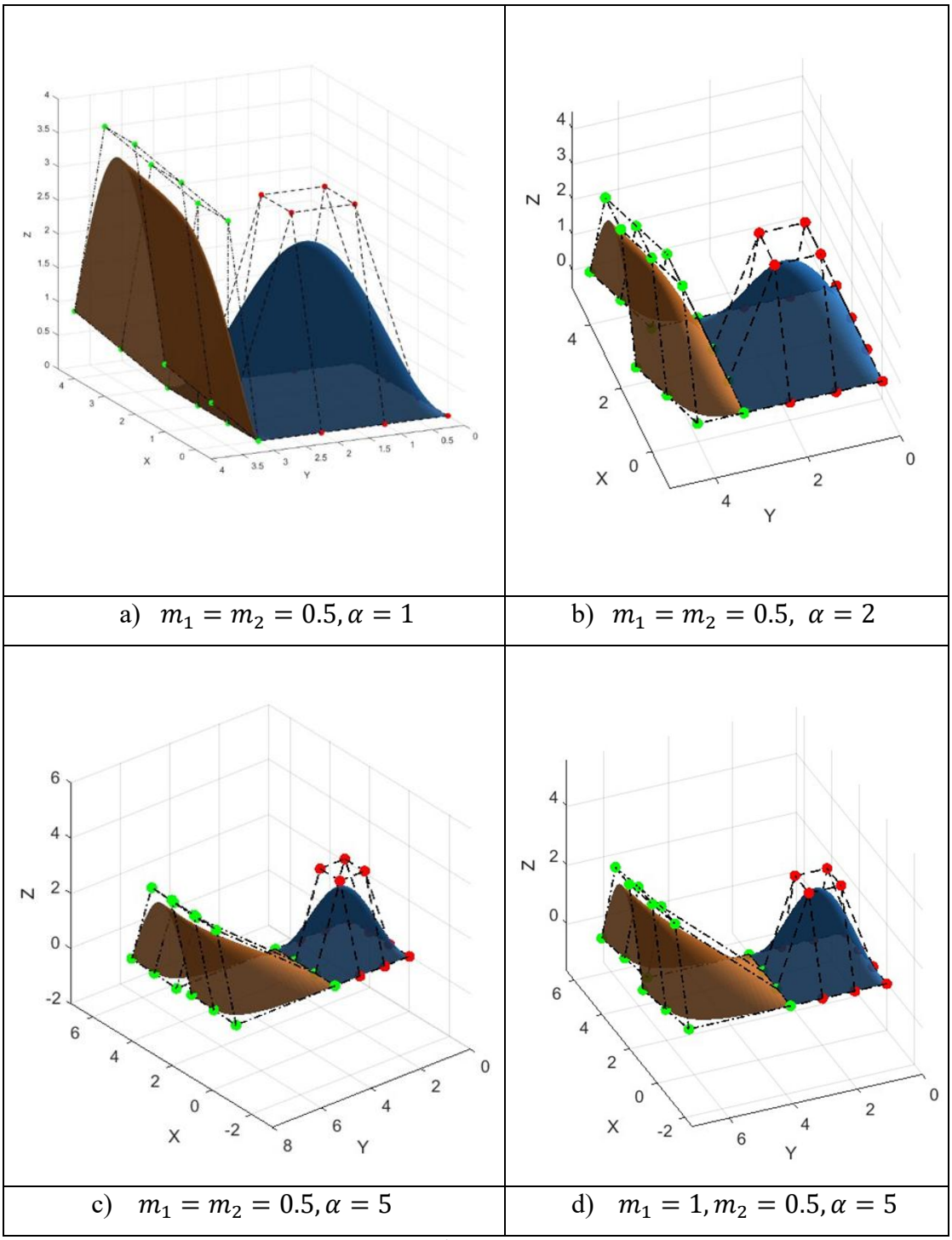
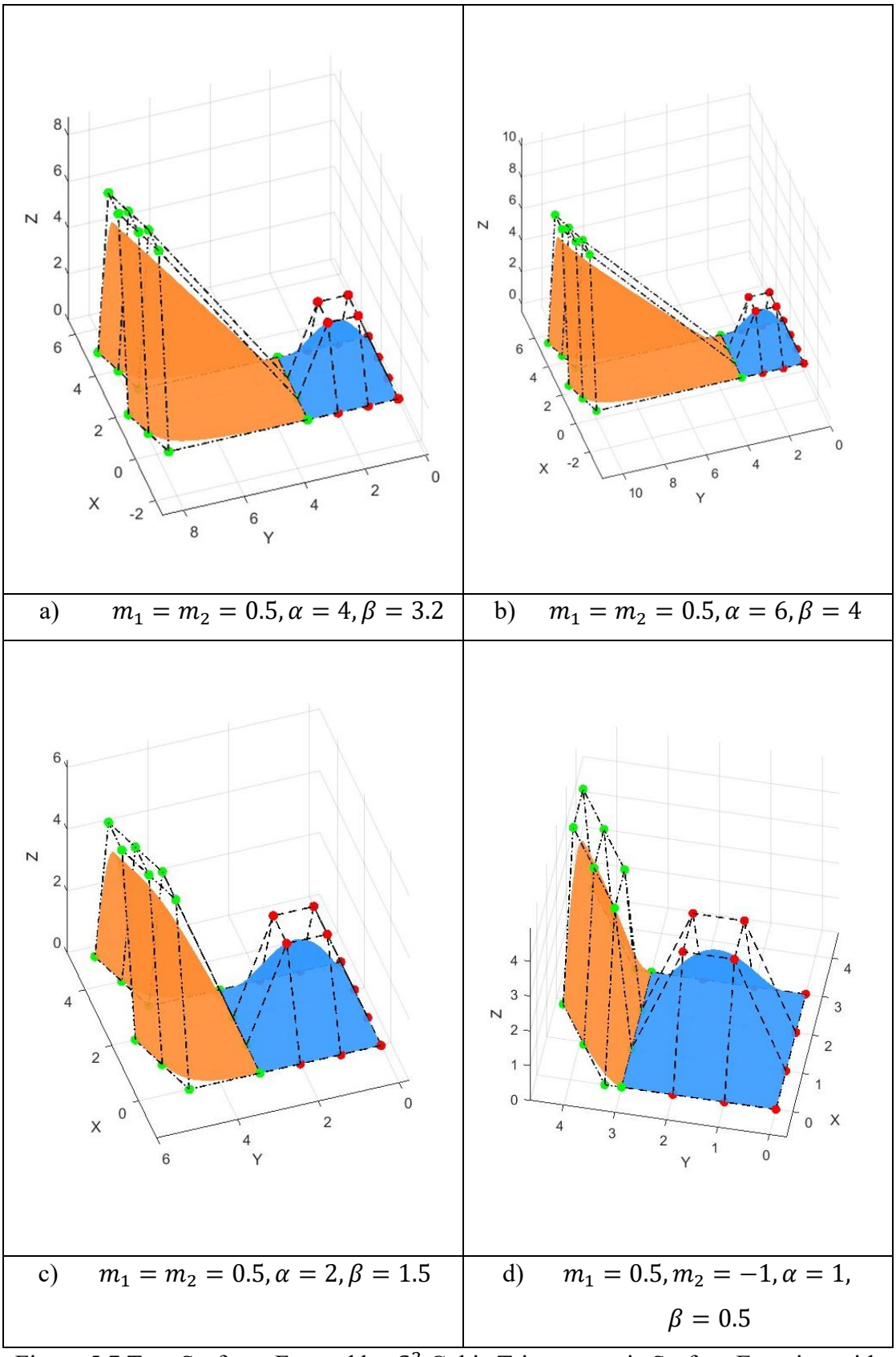


Figure 5. 6 Two Surfaces Formed by G^1 Cubic Trigonometric Surface Equation with Various Shape Parameters and Scalar Factor, α



a) $m_1 = m_2 = 0.5, \alpha = 4, \beta = 3.2$

b) $m_1 = m_2 = 0.5, \alpha = 6, \beta = 4$

c) $m_1 = m_2 = 0.5, \alpha = 2, \beta = 1.5$

d) $m_1 = 0.5, m_2 = -1, \alpha = 1, \beta = 0.5$

Figure 5.7 Two Surfaces Formed by G^2 Cubic Trigonometric Surface Equation with Various Shape Parameters and Scalar Factor, α, β

In Figure 5.4, two adjacent patches connected with C^1 continuity under different combinations of the shape parameters m_1 and m_2. Although each parameter chosen produces different curvature on the individual patches, the surfaces remain smoothly joined because both the boundary positions and the cross-boundary first derivatives are matched. This ensures that the tangent plane is continuous across the sharp edge, preventing any visual break in the surface. The results demonstrate that the proposed formulation consistently preserves C^1 smoothness while allowing flexible control over the boundary shape through the parameters m_1 and m_2.

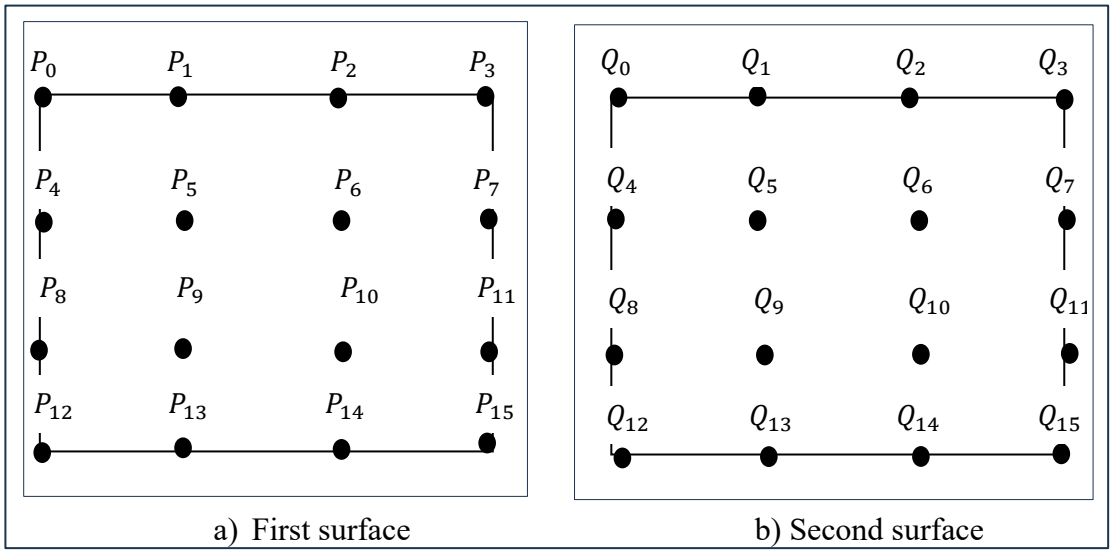
Figure 5.5 demonstrates the neighbouring surface patches connected with C^2 continuity under various selections of the shape parameters m_1 and m_2. All the connected patches remain smooth as the tangent direction, and its curvature are all matched across the shared edge. The results show that the method able to enforces C^2 continuity while still allowing the designer to adjust the parameters m_1 and m_2.

Figure 5.6 illustrates the two adjacent patches with C^1 continuity for different choices of the shape parameters m_1 and m_2, and the tangent-scaling factor α. In each case, the tangent directions on both sides are kept aligned, ensuring a smooth geometric transition between them. As α increases, the tangent is altered, producing noticeable changes in the local shape near the join, yet the surfaces remain visually smooth and free from sharp breaks.

Lastly, Figure 5.7 demonstrates two connected surface patches achieving G^2 continuity using various value of m_1 and m_2, and the geometric continuity α and β. In the figure, the orange and blue patches share the same boundary curve, while the boundary control points (shown in green and red) are arranged so that both the tangent direction and curvature direction match across the join. The parameters α and β govern the scaling of the first and second derivatives. The surfaces may appear to bend more sharply or more gently depending on the chosen values, yet the transition along the shared boundary seamless. Overall, the results give that the proposed formulation successfully maintains G^2 smoothness by ensuring consistent curvature flow between the parches, while still allowing adjusting the shape of the curves flexibly by using the parameters m_1,m_2, α and β.

5.4.3 Two Surfaces Generated (Cylinder) from Cubic Trigonometric Spline Surface Equation in u and v Direction

Figure 5.8 displays the schematic representations of the joined overlay of the two cubic trigonometric surfaces using the same value of m. The illustration for connecting in achieving parametric and geometric continuity of the two patches are illustrated using the control points. For joining the two patches in constructing cylinder, two rows of control points involved side by side. Below, the connection of the control points for each continuity conditions are presented in detail. Let $P_i, (i = 0,..,15)$ and $Q_i, (i = 0,..,15)$ be the control points for the first and second surface.



a) First surface b) Second surface

Figure 5.8 Respective Superposition of Control Points of Two Cubic Trigonometric Surfaces

C^0 and G^0 continuity

$$Q_0 = P_3 \qquad Q_3 = P_0$$
$$Q_4 = P_7 \qquad Q_7 = P_4$$
$$Q_8 = P_{11} \qquad Q_{11} = P_8$$
$$Q_{12} = P_{15} \qquad Q_{15} = P_{12}.$$

C^1 continuity

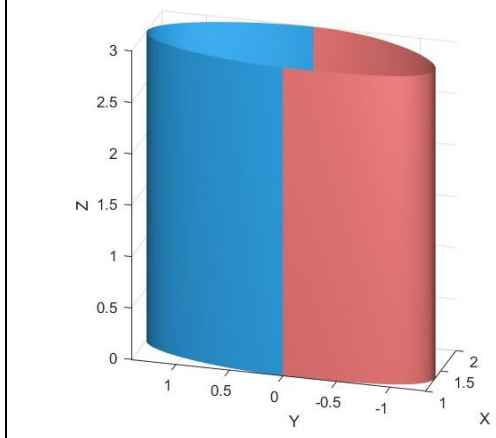
$$Q_1 = 2P_3 - P_2 \qquad P_1 = 2Q_3 - Q_2$$
$$Q_5 = 2P_7 - P_6 \qquad P_5 = 2Q_7 - Q_6$$
$$Q_9 = 2P_{11} - P_{10} \qquad P_9 = 2Q_{11} - Q_{10}$$
$$Q_{13} = 2P_{15} - P_{14} \qquad P_{13} = 2Q_{15} - Q_{14}.$$

C^2 continuity

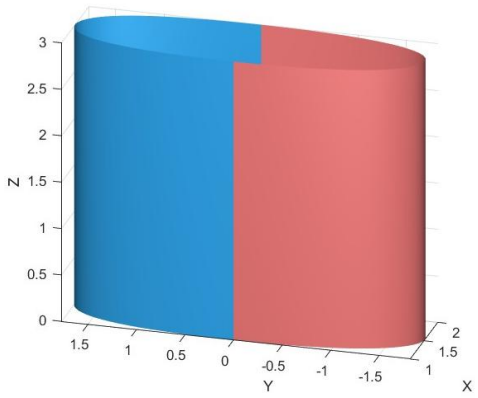
$$Q_2 = P_1 - (4m+4)P_2 + (4m+4)P_3$$
$$Q_6 = P_5 - (4m+4)P_6 + (4m+4)P_7$$
$$Q_{10} = P_9 - (4m+4)P_{10} + (4m+4)P_{11}$$
$$Q_{14} = P_{13} - (4m+4)P_{14} + (4m+4)P_{15}.$$

$$P_2 = Q_1 - (4m+4)Q_2 + (4m+4)Q_3$$
$$P_6 = Q_5 - (4m+4)Q_6 + (4m+4)Q_7$$
$$P_{10} = Q_9 - (4m+4)Q_{10} + (4m+4)Q_{11}$$
$$P_{14} = Q_{13} - (4m+4)Q_{14} + (4m+4)Q_{15}.$$

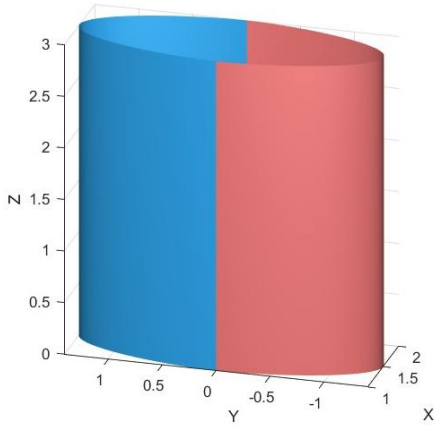
The figures below provide an example of cylinders that demonstrate the manner in which the shape of the generalized cubic trigonometric surfaces can be adjusted by using various continuity conditions and shape parameter.



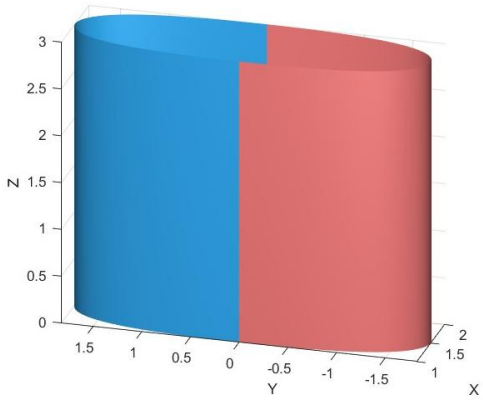
a) $m_1 = -1, m_2 = 0.5$



b) $m_1 = 0.5, m_2 = 0.2$



c) $m_1 = m_2 = -1$



d) $m_1 = m_2 = 0.7$

Figure 5.9 Cylinders Formed by C^1 Cubic Trigonometric Surface Equation with Various Values of Shape Parameters.

In Figure 5.9, cylinders are generated by two patches that connected continuously at both left and right sides with C^1 continuity. Each pair of patches shares a common boundary, ensuring that both the surface positions and the first derivatives match along the connecting curve. This guarantees a smooth transition without visible creases, even two patches are generated using different shape parameter values.

In each case, the shape parameters $m_1 = m_2$ influence the curvature and the cylindrical surface. In Figure 5.9(a), for contrasting value $m_1 = m_2$ create asymmetric cylindrical form, while in (b), smaller and more balanced parameter values yield a more uniform curvature. In (c), both negative values of shape parameter are used, produces a cylinder with a stronger inward bending effect, where else in (d), $m_1 = m_2 = 0.7$ results in smooth, outward-curving profile.

Across all four cases, the two coloured parches (blue and pink) meet seamlessly along the shared boundary because C^1 conditions ensure consistent tangent directions on both sides. This demonstrates that the proposed cubic trigonometric surface formulation can generate a range of cylindrical shapes while maintaining smooth and continuous joins under different choices of shape parameters.

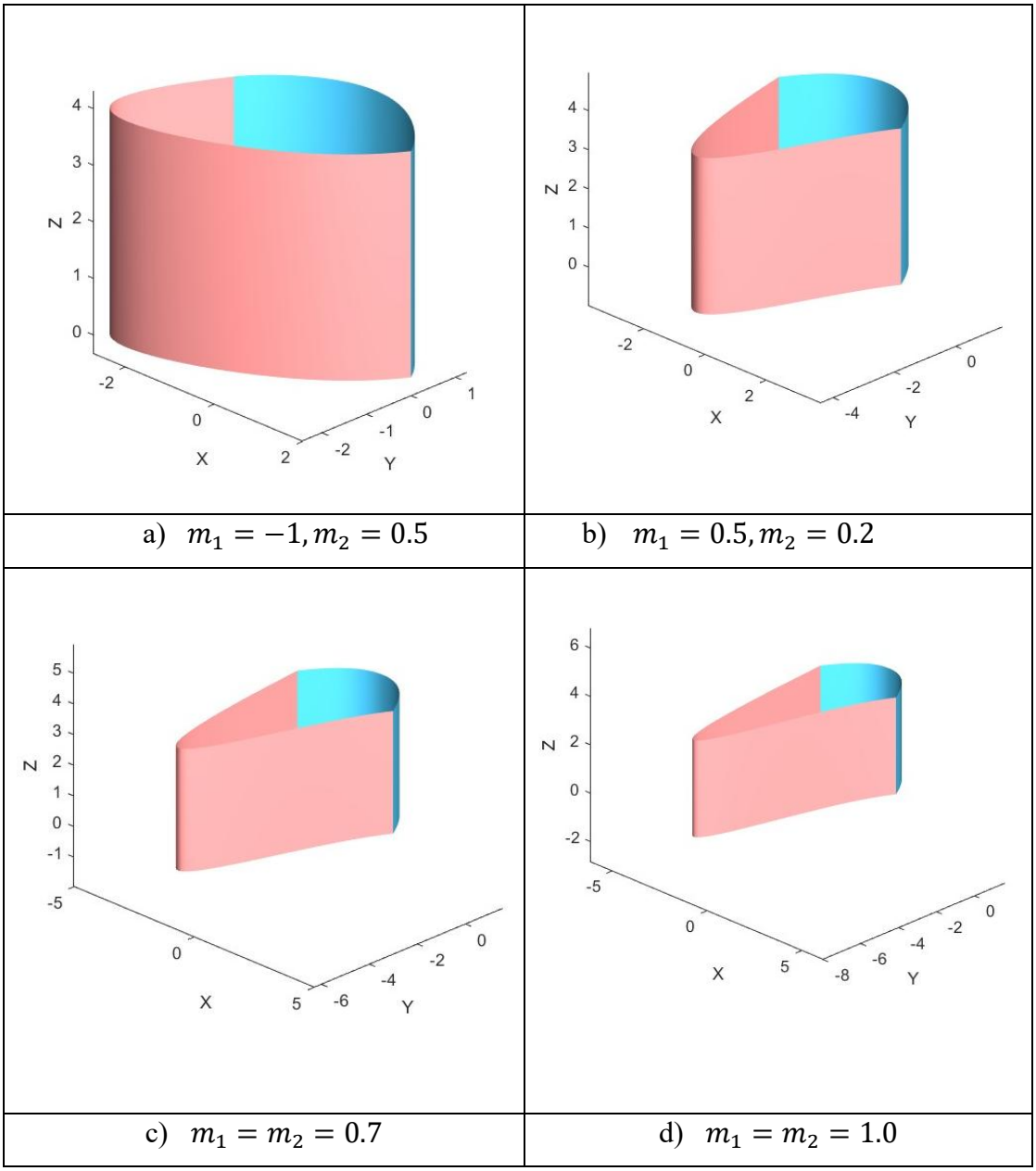
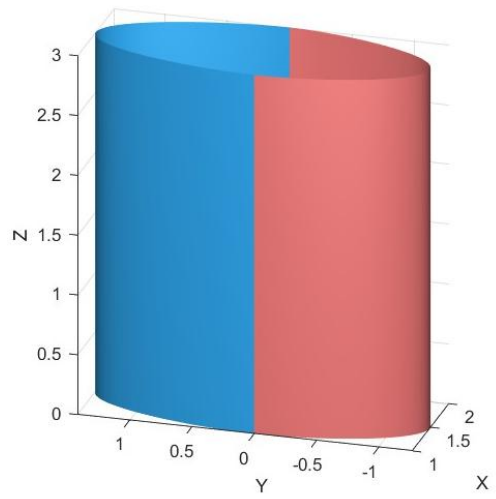


Figure 5.10 Cylinders Formed by C^2 Cubic Trigonometric Surface Equation with Various Values of Shape Parameters.

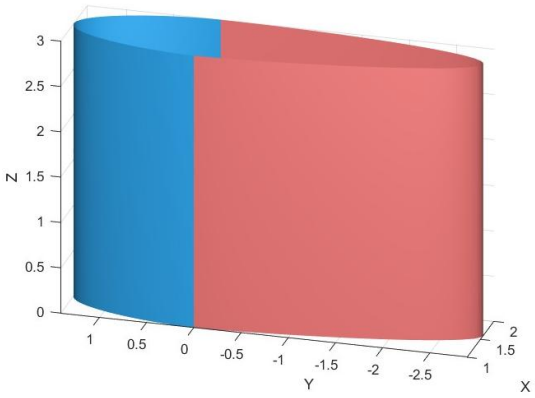
Figure 5.10 presents cylindrical surfaces generated using two adjacent cubic trigonometric patches that are joined with C^2 continuity. This level of continuity requires that the two patches match not only in position and first derivative but also in their second derivative along the shared boundary. As a result, the surfaces join with fully consistent curvature, producing a smooth transition in both slope and bending.

In Figure 5.10(a), the contrasting values $m_1 = 1$ and $m_2 = 0.5$ produce a pronounced curvature difference between the two halves of the cylinder, yet the connection remains smooth because the curvature is matched across the join. In (b), by using smaller parameter values the surface generated does not bend strongly, and the cylinder looks narrower at one end and slightly wider at the other. In (c), both parameters are equal to 0.7, produces a more balanced shape, while in (d), where both parameters are 1.0, yields the cylindrical surface bends more noticeably because the curvature is larger. The surface pushes outward more strongly, creating a rounded and appears swollen.

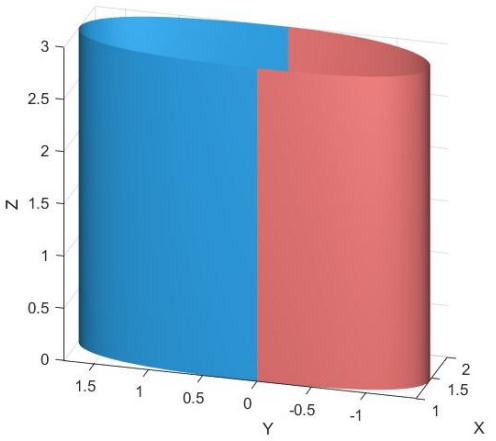
Despite the variation in shape produced by different parameter values chosen, the two-colour patches (cyan and pink) join smoothly in all cases. This shows that the proposed cubic trigonometric formulation maintains C^2 continuity by enforcing consistent curvature behaviour along the boundary.



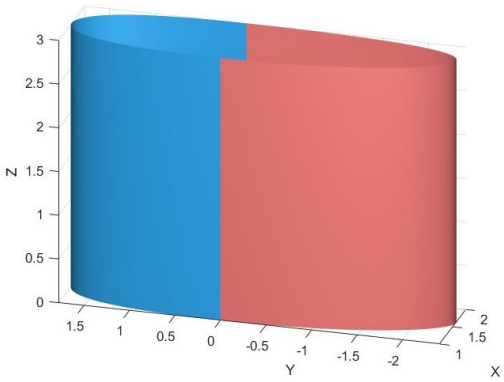
a) $m_1 = m_2 = -1.0, \alpha = 0.8$



b) $m_1 = m_2 = -1.0, \alpha = 3.0$



c) $m_1 = 0.5, m_2 = 0.2, \alpha = 0.8$



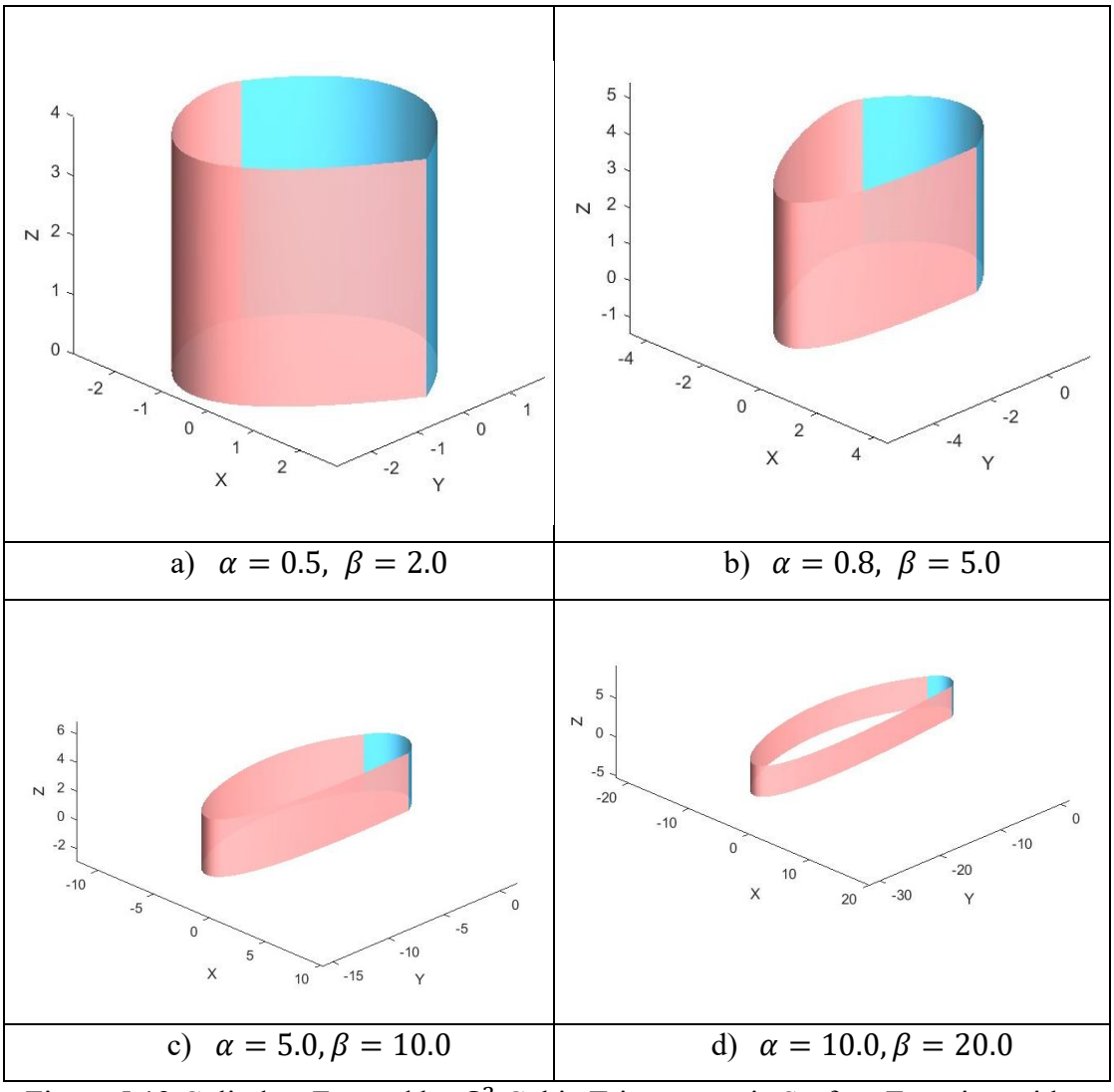
d) $m_1 = 0.5, m_2 = 0.2, \alpha = 2.0$

Figure 5.11 Cylinders Formed by G^1 Cubic Trigonometric Surface Equation with Various Values of Shape Parameters and Scalar Factor, α

Next, Figure 5.11 display cylindrical surfaces formed from two trigonometric patches that are joined using G^1 continuity. In a G^1 connection, the two patches share the same boundary curve, and their tangent directions along that boundary are aligned, although the magnitudes of the derivatives may differ.

Different values of the shape parameters $m_1, m_2,$ and the tangent-scaling factor α influence how each half of the cylinder bends. In Figure 5.11(a) and (b), where $m_1 = m_2 = 1.0$, the cylinder exhibits a stronger inward curvature, while increasing α makes the surface lean more the joining line. In Figure 5.11(c) and (d), the values of shape parameter used are more moderate and closer to zero, which results in a smoother cylindrical shape.

With all the variations in appearance, the two-coloured patches (blue and pink) always connect smoothly as the tangent directions remain consistent along their share boundary. This demonstrates that the formulation maintains G^1 geometric smoothness.



a) $\alpha = 0.5,\ \beta = 2.0$ b) $\alpha = 0.8,\ \beta = 5.0$

c) $\alpha = 5.0, \beta = 10.0$ d) $\alpha = 10.0, \beta = 20.0$

Figure 5.12 Cylinders Formed by G^2 Cubic Trigonometric Surface Equation with a Constant Value of $m_1 = 0.5$, $m_2 = 0.2$, and Varies Values of α and β

The four subfigures in Figure 5.12 show cylinders constructed with two trigonometric surface patches joined with G^2 geometric continuity, where the position, tangent direction, and curvature direction along the boundary are matched. All cylindrical shapes in the figure are generated with same shape parameter values $m_1 = 0.5$ and $m_2 = 0.2$ but different values of α and β. Since the values of m_1 and m_2 are similar for all cases, the basis cylindrical structure remains consistent, and the differences in appearance are entirely due to the values of α and β, which control the strength of the first- and second- derivative effects.

In Figure 5.12(a), by using relatively small values $\alpha = 0.5,\ \beta = 2.0$, the cylinder appears close to its natural form, with only gentle curvature adjustments. In (b). moderated increases in α and β make the cylinder narrow more clearly and bend

more. When α and β become larger, as in (c), the effect of the curvature constraints intensifies, producing a strongly bent and elongated cylindrical shape. Finally, in (d), very large values $\alpha = 10.0, \beta = 20.0$ dominate the geometry, stretching and distorting the cylinder into a highly curves, almost loop like form.

Despite these dramatic shape changes, the two-coloured patches remain smooth connected because the G^2 conditions guarantee consistent curvature behaviour along the shared boundary.

5.5 Designing 3D Image Using Real Data

Next, a large dataset was employed to assess the performance of the proposed cubic trigonometric spline under specified continuity conditions. The reconstructed human face and pelvic, generated using cubic trigonometric interpolation using (5.3) are depicted in Figure 5.13 and Figure 5.14, respectively. The human face dataset was obtained from Hadi, N. A. (2014). G^2 parametric curve and surface fitting using beta spline (Doctoral dissertation, Universiti Teknologi MARA), whereas the pelvic dataset was sourced from (Hadi & Alias, 2020). Both figures consist of thousands boundary data points to outline the reconstruction process. They are developed by numbers layer of patches, which ensuring C^0 and G^0 continuity is achieve within the boundaries. The presence of shape parameter, m are manipulated in order to get the best fitted surface. Increasing the number of related points will enhance the accuracy of the reconstructed image, but it is not efficient in terms of time consumption. The illustration demonstrates the efficacy of cubic trigonometric spline in generating a high-quality and complex reconstructed image.

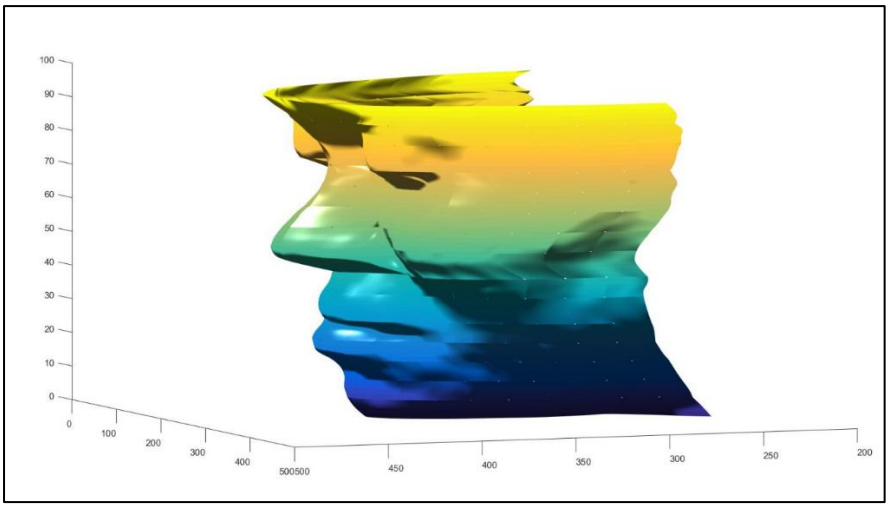


Figure 5.13 The 3D Surface Reconstruction Result of Human Face

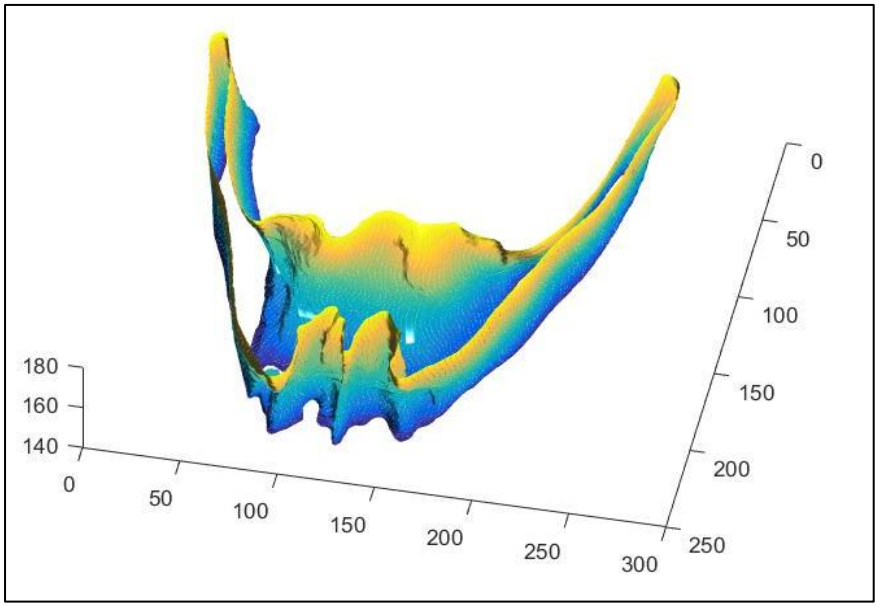


Figure 5.14 The 3D Surface Reconstruction Result of Pelvic

5.6 Summary

This chapter has presented a detailed investigation into the construction and analysis of cubic trigonometric surface patches under various continuity requirements. By examining both parametric continuity (C^1 and C^2) and geometric continuity (G^1 and G^2), the chapter has demonstrated the impact of each continuity level on the smoothness of connected surfaces and how those requirements can be satisfied by adjusting the control points, tangents, and curvature conditions along the boundary. The results

clearly show that cubic trigonometric surfaces offer significant modelling flexibility, allowing complex shapes to be created while still satisfying high smoothness level.

Numerous illustrations of a single surface, two connected surfaces, and cylindrical forms with effects of varying scaling factors and shape parameter values are included in this chapter. Under C^1 continuity, surfaces maintain a smooth tangent transition, and the examples show that by using different choices of shape parameters able to adjust the form of each patch without breaking the smooth join. For C^2 continuity, the matching of curvature produces even better in term of smoothness, ensuring that bending behaviour flows continuously across boundaries. Similarly, the G^1 and G^2 examples highlight the advantages of geometric continuity, where direction-based alignment of tangents and curvature can be achieved with greater flexibility compared to strict parametric matching. The presence of shape parameters and scaling factors is significant since they provide convenience in determining and adjusting the shape of the surface without the need to modify the control planes. This offers designers more freedom in controlling surface behaviour while maintaining visual smoothness.

The cylindrical case studies further emphasise the versatility of the method. Even with significant variations in shape parameters and scalar factor, the surfaces remain smooth and free from undesirable artefacts such as creases or curvature breaks. These examples show that the approach can generate a wide variety of surfaces while maintaining the intended continuity constraints.

The application of parametric and geometric continuity conditions makes a smooth connection between surface patches possible. The surfaces are formed in the relevant directions. This demonstrates the strong shape-control capability inherent in the cubic trigonometric formulation.

The modelling examples and reconstructed of 3D images from real data demonstrated the practicality of the suggested techniques: compared to existing Bezier-like surfaces, the generalized Bezier-like surfaces offer stronger form adjustability and approximation capability. This study's description of higher order continuity is not only comprehensible and simple to use, but it also provides more options for generating challenging surfaces in engineering design.

To sum up, the analysis carried out in this chapter confirms that cubic trigonometric surface patches act as mathematical tool for constructing smooth and adaptable surfaces. The ability in maintaining the continuity, respond predictably to parameter adjustments, and produce visually coherent shapes makes them valuable for

applications in CAGD, animation, engineering modelling, and surface reconstruction. The findings presented serve as a foundation for further exploration, including multi-patch surface modelling and potential extensions into higher-dimensional or real-time modelling frameworks.

CHAPTER 6
TRIGONOMETRIC CURVE FITTING ON 2D IMAGE OUTLINE

6.1 Introduction

Data obtained in the CAGD area is commonly derived from precise figures of an existing element by keeping an image in mathematical representation. This approach will benefit in requiring less file storage where the data is preserved in coordinate form instead of being stored in image form (Hadi, 2014). Next, the issue arises of fitting curves to data points in numerous natural industries, including manufacturing, data analysis, and image processing. It is necessary to fit a point passing through or as close to the data point while attempting to generate a smooth curve.

In the step of reconstructing a curve using a mathematical function, curve reconstruction is necessary. An essential component of curve reconstruction is curve fitting, which discovers a set of data points objectively that can precisely represent a predefined curve. Bezier, B-spline, and NURBS are among the preferred methods for the curve fitting process. Each method showed the differences in terms of various knot distributions and also the presence of parameters. Meanwhile, all the techniques used the best approximation method called LSM to obtain the control points to fit the data point. Specifically, Bezier's parametric curve fitting method is being considered in this research.

This chapter intends to reconstruct a mathematical outline of 2D boundary-based images of the Greek alphabet and Arabic letters. The letters that are fourth and fifth of the Greek alphabet letters are delta, δ, epsilon, ε, and the Arabic letter '*ى*.' The selection of these letters is based on their curved edges, which necessitate the use of piecewise curve fitting. In industries such as digital printing, logo and branding, animation, advertising, and Islamic art restoration rely on accurate geometric representations of Arabic script. By applying cubic trigonometric spline, it is expected that the spline will generate smoother and flexible curves. better visual quality and consistency. This study offers practical solutions by offering mathematical technique that improves the modelling of complex calligraphic shapes.

The images are reconstructed using four different kinds of cubic trigonometric basis functions taken from the proposed cubic trigonometric spline (3.7), Majeed & Qayyum (2020), Han et al., (2010) and Usman et al., (2020). All the splines consist of the same criteria regarding the number of shape parameters with different range values, degrees of the splines, and Bezier-like curve properties.

The splines will apply all the curve fitting processes algorithm, as illustrated in Figure 6.1, including the curve refinement process and approximation error calculation. The ultimate goal is to have an effective spline with the total minimal distance produced between the fitted curve from the corresponding data point of the real image. Some descriptive examples show the performance of each generated spline. To obtain a highly accurate curve, the LSM is employed, and Sum Square Error (SSE) is utilized to compute the inaccuracy in the approximation.

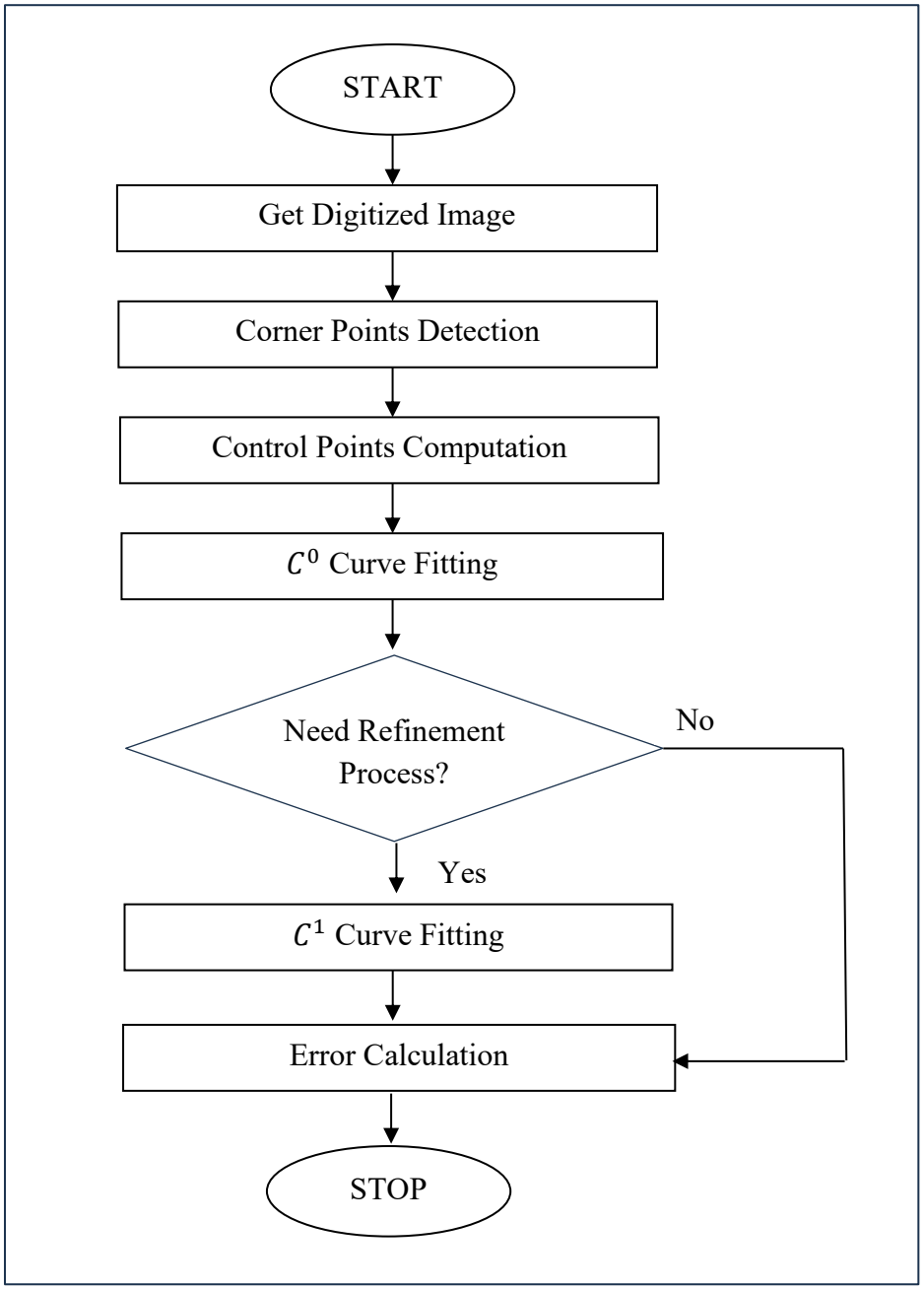
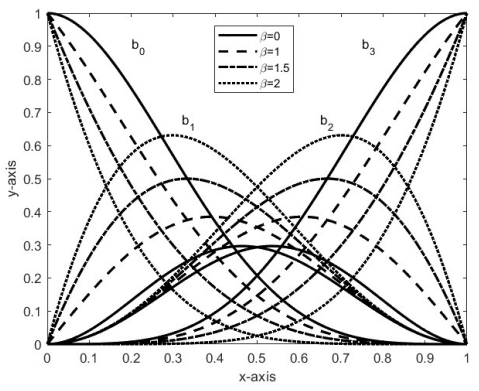
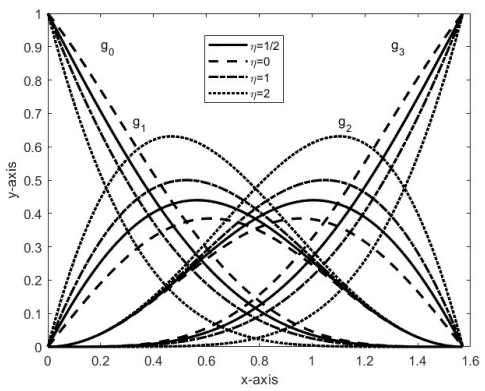


Figure 6. 1 The Pre-processing Steps of Curve Fitting Operation Algorithm

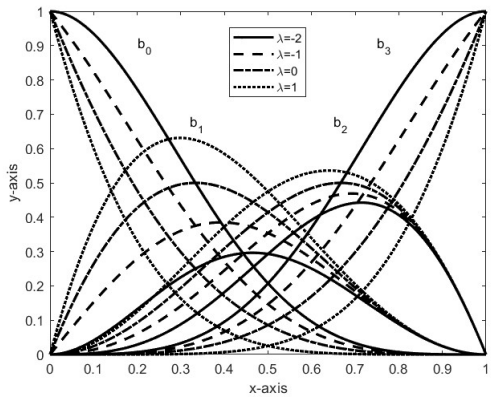
6.2 Cubic Trigonometric Curve



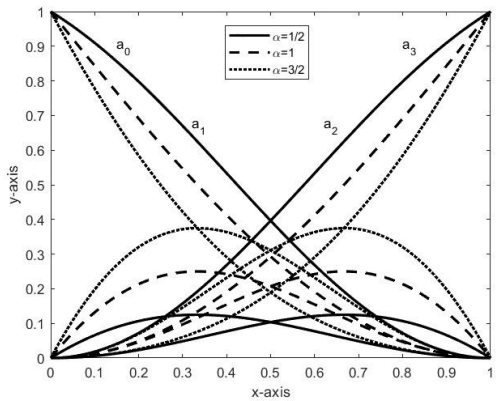
a) Proposed cubic trigonometric spline



b) Majeed & Qayyum (2020)



c) Han et al., (2010)



d) Usman et al., (2020)

Figure 6.2 The Basis Functions

All four cubic trigonometric basis functions are presented in this section. First, the basis functions of the proposed cubic trigonometric spline with shape parameter m from (3.7) are demonstrated in Figure 6.2 (a) with the interval $m = [-2,1]$.

Next, the basis functions proposed by Majeed & Qayyum, (2020) are constructed in Figure 6.2 (b) with shape parameter $\eta = (1/2,2]$, and $t \in \left[0, \frac{\pi}{2}\right]$, based on below equations (6.1).

$$
\begin{aligned}
g_0(t) &= (1-\sin t)^2\,(1+(1-\eth)\sin t),\\
g_1(t) &= \big(2-(1-\eth)(1-\sin t)\big)(\sin t)(1-\sin t),\\
g_2(t) &= \big(2-(1-\eth)(1-\cos t)\big)(\cos t)(1-\cos t),\\
g_3(t) &= (1-\cos t)^2\,(1+(1-\eth)\cos t).
\end{aligned}
\tag{6.1}
$$

Figure 6.2 (c) shows the cubic trigonometric spline with a shape parameter λ in the interval [-2,1] and $t \in [0,1]$, generated by Han et al., (2010). The curves are developed by using equation basis functions in (6.2).

$$
\begin{aligned}
b_0(t) &= \left(1-sin\frac{\pi}{2}t\right)^2\left(1-\lambda sin\frac{\pi}{2}t\right),\\
b_1(t) &= sin\frac{\pi}{2}t\,\left(1-sin\frac{\pi}{2}t\right)\left(2+\lambda-\lambda sin\frac{\pi}{2}t\right),\\
b_2(t) &= cos\frac{\pi}{2}t\,\left(1-cos\frac{\pi}{2}t\right)\left(2+\lambda-\lambda cos\frac{\pi}{2}t\right),\\
b_3(t) &= \left(1-cos\frac{\pi}{2}t\right)^2\left(1-\lambda cos\frac{\pi}{2}t\right).
\end{aligned}
\tag{6.2}
$$

Lastly, trigonometric Cubic Bernstein-like basis functions from Usman et al., (2020) are developed using the basis functions in (6.3). The basis functions are illustrated in Figure 6.3 (d).

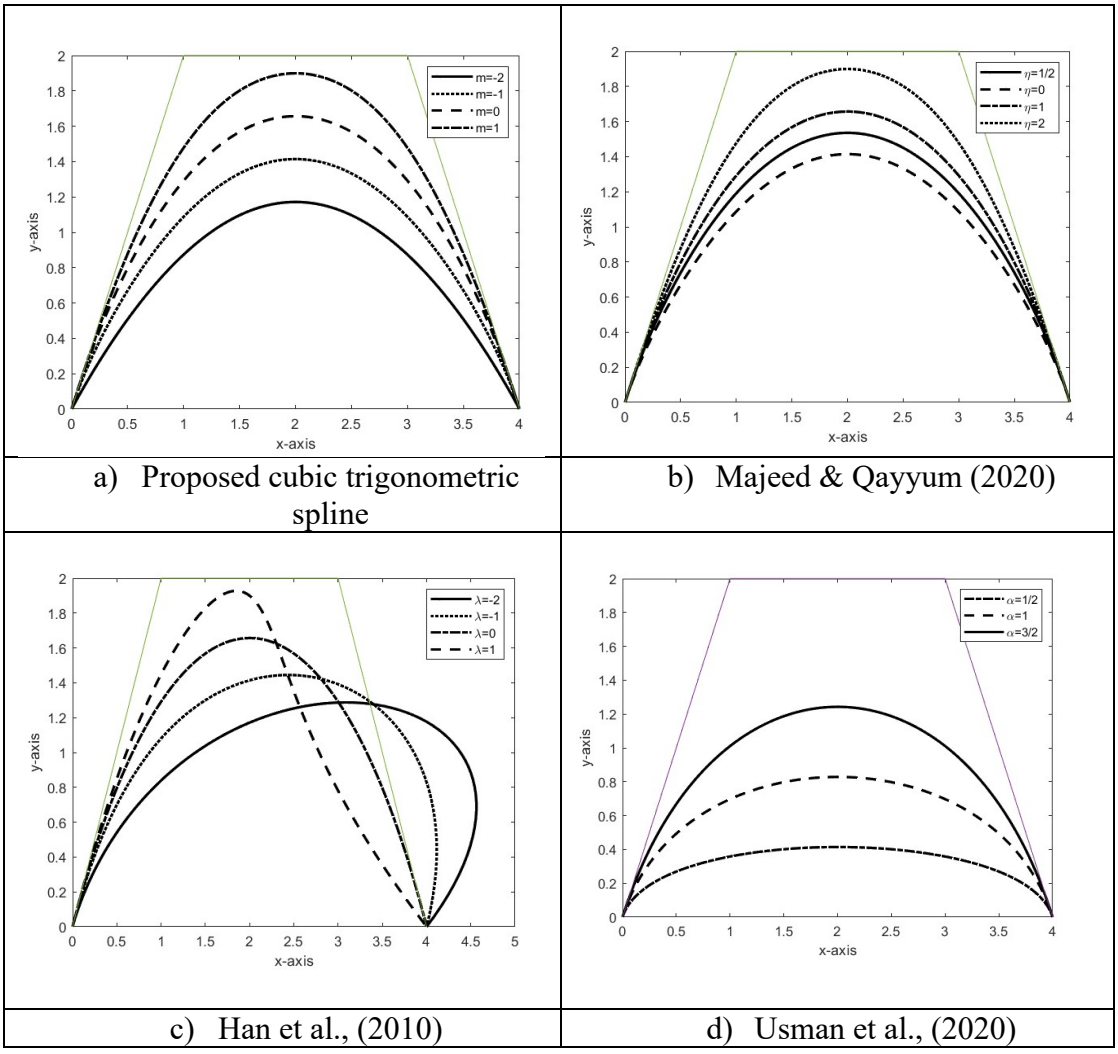
$$
\begin{aligned}
a_0(\eth) &= \alpha S^2-\alpha S+U^2,\\
a_1(\eth) &= \alpha(S-S^2),\\
a_2(\eth) &= \alpha(S^2+U-1),\\
a_3(\eth) &= (1-\alpha)S^2-\alpha U+\alpha.
\end{aligned}
\tag{6.3}
$$

where $S = sin\left(\frac{\pi\eth}{2}\right)$, $U = cos\left(\frac{\pi\eth}{2}\right)$, $\alpha \in (0,2)$ and $\eth \in [0,1]$.

Basically, the general equation for generating curves using the developed basis functions for all four comparative splines is as shown below,

$$
F(t) = \sum_{t=0}^{n} b_i(t)V_i \tag{6.4}
$$

where $b_i(t)$ is the basis functions and V_i is the i^{th} control point. Four different cubic trigonometric basis functions are presented using similar control points where $V_0 = [0,0], V_1 = [1,2], V_2 = [3,2]$ and $V_3 = [4,0]$. The cubic trigonometric spline is generated based on a specified interval chosen by each spline type.



a) Proposed cubic trigonometric spline

b) Majeed & Qayyum (2020)

c) Han et al., (2010)

d) Usman et al., (2020)

Figure 6.3 Trigonometric Curve

In Figure 6.3 (a), for the proposed cubic trigonometric spline, by increasing the value of the shape parameter, m, causes the fitted curves drawn nearer to the control polygon. All resulting curves remain entirely within the convex hull. Similarly, in Figure 6.3 (b), the fitted cubic trigonometric spline only approximates the control polygon as the values of the shape parameter ŋ increases, exhibiting the same general behaviour.

Next, in Figure 6.3 (c), when the value of shape parameters $\lambda = 0$ and $\lambda = 1$ are used, the curves are generated within the convex hull. However, the curves are

generated out of the convex hull when negative values such as $\lambda = -1$ and $\lambda = -2$ are applied even though the shape parameter values are fall within the given range.

For the cubic trigonometric spline shown in Figure 6.3 (d), all curves demonstrated for the specified intervals $\eta \in [0,1]$ fitted within the convex hull. Despite employing a larger shape parameter, the curves develop at a greater distance from the control polygon.

6.3 Data Acquisition

The initial step of the curve fitting procedure involves obtaining the digitised data points. The data set for 'epsilon', ε consists of 745 points, the data set for 'delta', δ, consists of 460 points, and the data set for 'ى' consists of 366 points. The boundary data points are spaced at a theoretical distance of three pixels, as depicted in Figure 6.4.

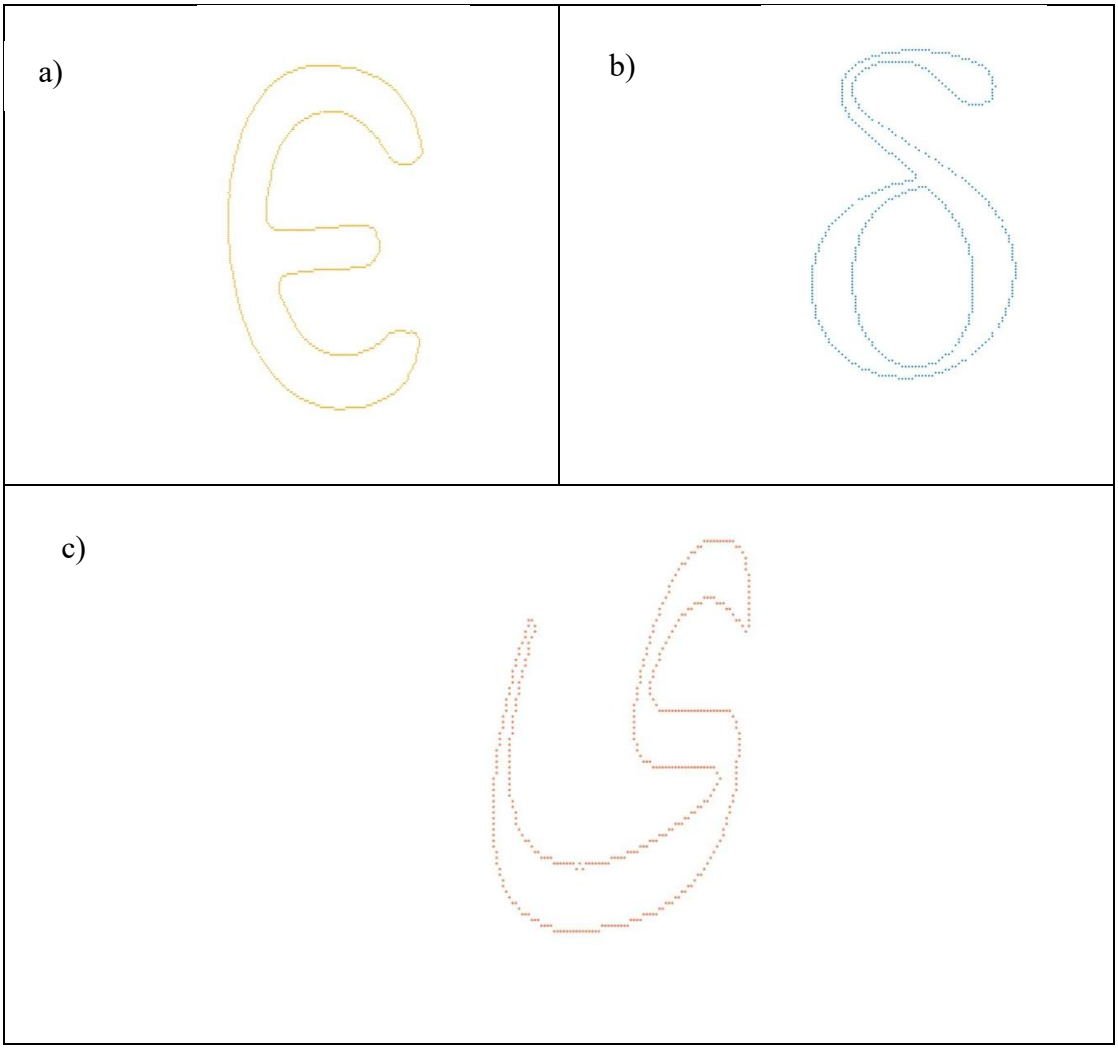


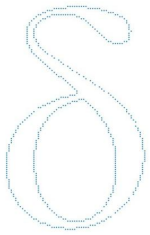
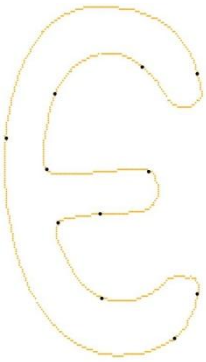
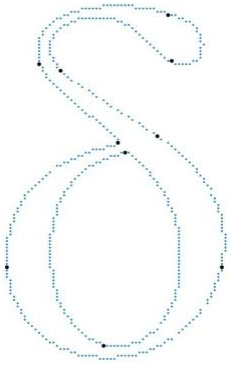
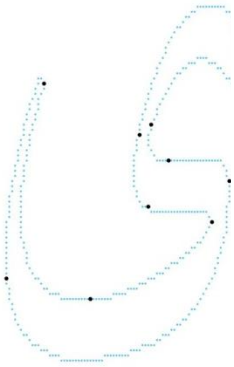
Figure 6. 4 Contour of (a) 'Epsilon', (b) 'Delta' and (c) 'ى'

6.4 Corner Point Detection

In this stage, the vertices of the outlines of the proposed shape are identified. This will result in breaking the outlines into several segments to set up the control points. Each segment will represent the end control points of the curve segments. The importance of this step is to reduce the difficulty in the fitting process. The process also needs better approximation due to the boundary's condition being separated into smaller parts (Sarfraz et al., 2013). A precise determination of the corner points is essential for a suitable fitting or portrayal of the design. The areas where two edges meet at an angle, forming angular spaces, are known as corners. A corner can be described as a point when two rigid and distinguishable edge directions are quite close to the point. The number of corners in this research is detected by manual intervention by employing inconsistency along the contour direction to identify the curves from the actual imagery boundaries. Another approach is to evaluate the visual curvature along the contour direction. These two approaches are simple and easy, neglecting the works of computation for determining the corner point. The corner detection for 'epsilon', ε, 'delta', δ, and 'ى' is presented in Table 6. 1.

The corner detection technique identifies the boundary points within a collection of numerous segments. The total quantity of segments for 'epsilon', ε is eleven. The letter 'delta', δ, is composed of 10 segments, including two inner boundaries and eight boundary segments. The letter 'ى', comprises of 10 segments. The letters 'epsilon', ε, and 'ى' do not have any internal boundaries.

Table 6. 1
Outline Images and Detection of Corner Points

Name	**'Epsilon', ε**	**'Delta', δ**	**'ى'**
Outline image			
Corner Detections			

6.5 Parameterization

In Chapter 3, the cubic trigonometric spline is interpolated at the beginning and end points and consists of two interior points that determine the shape of the curve. This cubic trigonometric spline has four control points: V_0, V_1, V_2 and V_3. The initial and final points of the curve are V_0 and V_3. The two end control points correspond to the two corner points of the curve segment. At this stage, the two intermediate points V_1 and V_2 were calculated using parameterization to minimize the distance between the boundary

and the parametric curve. The parameterization approach was effective in assisting designers in constructing a curve that interpolates the chosen data points. The least square approach is employed for this particular reason. The resulting points will then be utilized to identify the parametric curve that yields the most significant outcomes for a certain value of t.

6.5.1 Least Squares Method

Let a set of data points be approximated as P where

$$P = \{P_i |\ i = 1,2,3, \dots, N\} \tag{6.5}$$

and the fitted cubic trigonometric spline curve at the knot t_i as,

$$F(t_i) = V_0 b_0(t_i) + V_1 b_1(t_i) + V_2 b_2(t_i) + V_3 b_3(t_i) \tag{6.6}$$

with basis function, $b_i(t)$ as given in (3.7). The expression for the sum of squared distances, S_i from the boundary of the original image, P_i to parametric curve $F(t_i)$ can be represented as follows,

$$S_i = ||F(t_i) - P_i||^2 \tag{6.7}$$

Therefore, for a set of data points P, total distance S is written as,

$$\begin{aligned} S &= \sum_{i=1}^{N} S_i \\ &= \textstyle\sum_{i=1}^{N} ||V_0 b_0(t_i) + V_1 b_1(t_i) + V_2 b_2(t_i) + V_3 b_3(t_i) - P_i||^2. \end{aligned} \tag{6.8}$$

The data parameterization approach is used to determine the t_i value. This method may often be expressed as,

$$t_i = \frac{\sum_{r=1}^{i} DP_r}{\sum_{r=1}^{N} DP_r} \tag{6.9}$$

with DP_i be the difference measured by two data points, evaluated as

$$DP_i = ||P_{i+1} - P_i||. \tag{6.10}$$

The data endpoints, P_1 and P_N can be utilized to determine the end control points V_0 and V_3. Thus, V_1 and V_2 are two unidentified control points. The aim is to reduce distance, S. By letting the distance S, between the curve and data points, get closer to zero, these two points can be computed.

For instance, in V_1 case, let the change of the distance S with respect to the second control point, V_1 as zero and written as,

$$\frac{\partial S}{\partial V_1} = 0. \tag{6.11}$$

Substitution of the equation of S from (6.8) into (6.11) above produces,

$$2\sum_{i=1}^{N} b_1(t_i)[V_0 b_0(t_i) + V_1 b_1(t_i) + V_2 b_2(t_i) + V_3 b_3(t_i)] = 0$$

$$\sum_{i=1}^{N} [V_0 b_0(t_i) b_1(t_i) + V_1 b_1^2(t_i) + V_2 b_1(t_i) b_2(t_i) + V_3 b_1(t_i) b_3(t_i) - P_i b_1(t_i)] = 0$$

$$\sum_{i=1}^{N} V_0 b_0(t_i) b_1(t_i) + \sum_{i=1}^{N} V_1 b_1^2(t_i) + \sum_{i=1}^{N} V_2 b_1(t_i) b_2(t_i) + \sum_{i=1}^{N} V_3 b_1(t_i) b_3(t_i) - \sum_{i=1}^{N} P_i b_1(t_i) = 0. \tag{6.12}$$

Operations on these parameters are segregated from the unknown control points V_1 and V_2, as the two end control points V_0 and V_3, as well as the data point P_i, are already recognized.

$$\sum_{i=1}^{N} V_1 b_1^2(t_i) + \sum_{i=1}^{N} V_2 b_1(t_i) b_2(t_i) = \sum_{i=1}^{N} P_i b_1(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_1(t_i) - \sum_{i=1}^{N} V_3 b_1(t_i) b_3(t_i). \tag{6.13}$$

Steps in (6.11) and (6.13) are applied for V_2 case and produce,

$$\begin{aligned}\sum_{i=1}^{N} V_1 b_1(t_i) b_2(t_i) &\\ + \sum_{i=1}^{N} V_2 b_2^2(t_i) &\\ = \sum_{i=1}^{N} P_i b_2(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_2(t_i) - \sum_{i=1}^{N} V_3 b_2(t_i) b_3(t_i).&\end{aligned} \tag{6.14}$$

Evaluations (6.13) and (6.14) are two linear functions of two unknowns V_1 and V_2. Thus, the solution can be evaluated simultaneously in matrix operations. The matrix representation of (6.13) and (6.14) is,

$$\begin{bmatrix} \sum_{i=1}^{N} b_1^2(t_i) & \sum_{i=1}^{N} b_1(t_i) b_2(t_i) \\ \sum_{i=1}^{N} b_1(t_i) b_2(t_i) & \sum_{i=1}^{N} b_2^2(t_i) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} P_i b_i(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_1(t_i) - \sum_{i=1}^{N} V_3 b_1(t_i) b_3(t_i) \\ \sum_{i=1}^{N} P_i b_i(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_1(t_i) - \sum_{i=1}^{N} V_3 b_1(t_i) b_3(t_i) \end{bmatrix} \tag{6.15}$$

Matrix in (6.15) is simplified as

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix} \tag{6.16}$$

where

$$A_{11} = \sum_{i=1}^{N} b_1^2(t_i)$$

$$A_{12} = A_{21} = \sum_{i=1}^{N} b_1(t_i) b_2(t_i)$$

$$A_{22} = \sum_{i=1}^{N} b_2^2(t_i)$$

$$B_{11} = \sum_{i+1}^{N} P_i b_1(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_1(t_i) - \sum_{i=1}^{N} V_3 b_1(t_i) b_3(t_i)$$

$$B_{12} = \sum_{i+1}^{N} P_i b_2(t_i) - \sum_{i=1}^{N} V_0 b_0(t_i) b_2(t_i) - \sum_{i=1}^{N} V_3 b_2(t_i) b_3(t_i).$$

To simplify

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = A^{-1} \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix}$$

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \frac{1}{A_{11}A_{22} - A_{12}A_{21}} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix}$$

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \frac{1}{A_{11}A_{22} - A_{12}^2} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix}.$$

Therefore, two intermediate control points V_1 and V_2 are evaluated as

$$V_1 = \frac{A_{22}B_{11}}{A_{11}A_{22} - A_{12}^2} - \frac{A_{12}B_{12}}{A_{11}A_{22} - A_{12}^2} \tag{6.17}$$

$$V_2 = \frac{A_{11}B_{12}}{A_{11}A_{22} - A_{12}^2} - \frac{A_{21}B_{11}}{A_{11}A_{22} - A_{12}^2}. \tag{6.18}$$

Values for each of the four control points V_0, V_1, V_2 and V_3 and t are calculated.

6.6 Curve Fitting

By determining the optimal values for the middle control points V_1 and V_2, it is possible to obtain a well-suited cubic trigonometric spline. The cubic trigonometric curve is fitted to all segments during the steps of fitting the curve. The parameter was evaluated over a representative set of values sampled across its admissible range, and 0.6 provided the most favourable curve behaviour. For consistency across all methods, the parameter value was fixed at $m = 0.6$ for every generated curve. Given that this value lies within the allowable interval of all compared methods, applying it uniformly provides an equitable basis for assessing their shape characteristics. The presence of shape parameters in the cubic trigonometric serves to modify the fitted curve in such a way that all segments closely align with the outermost data points. Figure 6.5, Figure 6.6, and Figure 6.7 display the representation of fitted cubic trigonometric curves (solid line) over the boundary (dotted line). The approximation errors produced are computed.

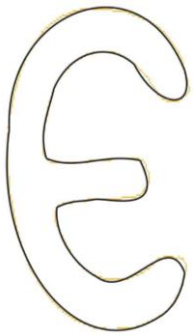
	
a) Proposed spline	b) Majeed & Qayyum (2020)
	
c) Han et al., (2010)	d) Usman et al., (2020)

Figure 6. 5 The Visualization of 'Epsilon' Result of Cubic Trigonometric Curves (solid line) Over the Boundary (dotted line)

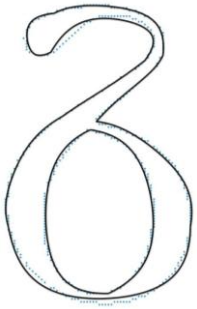
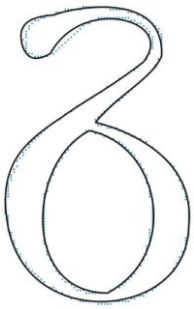
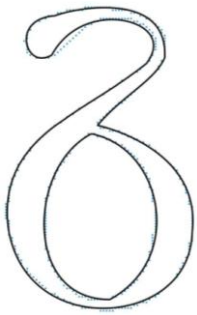
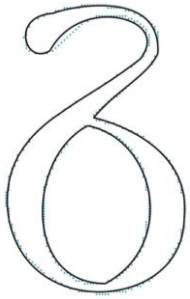
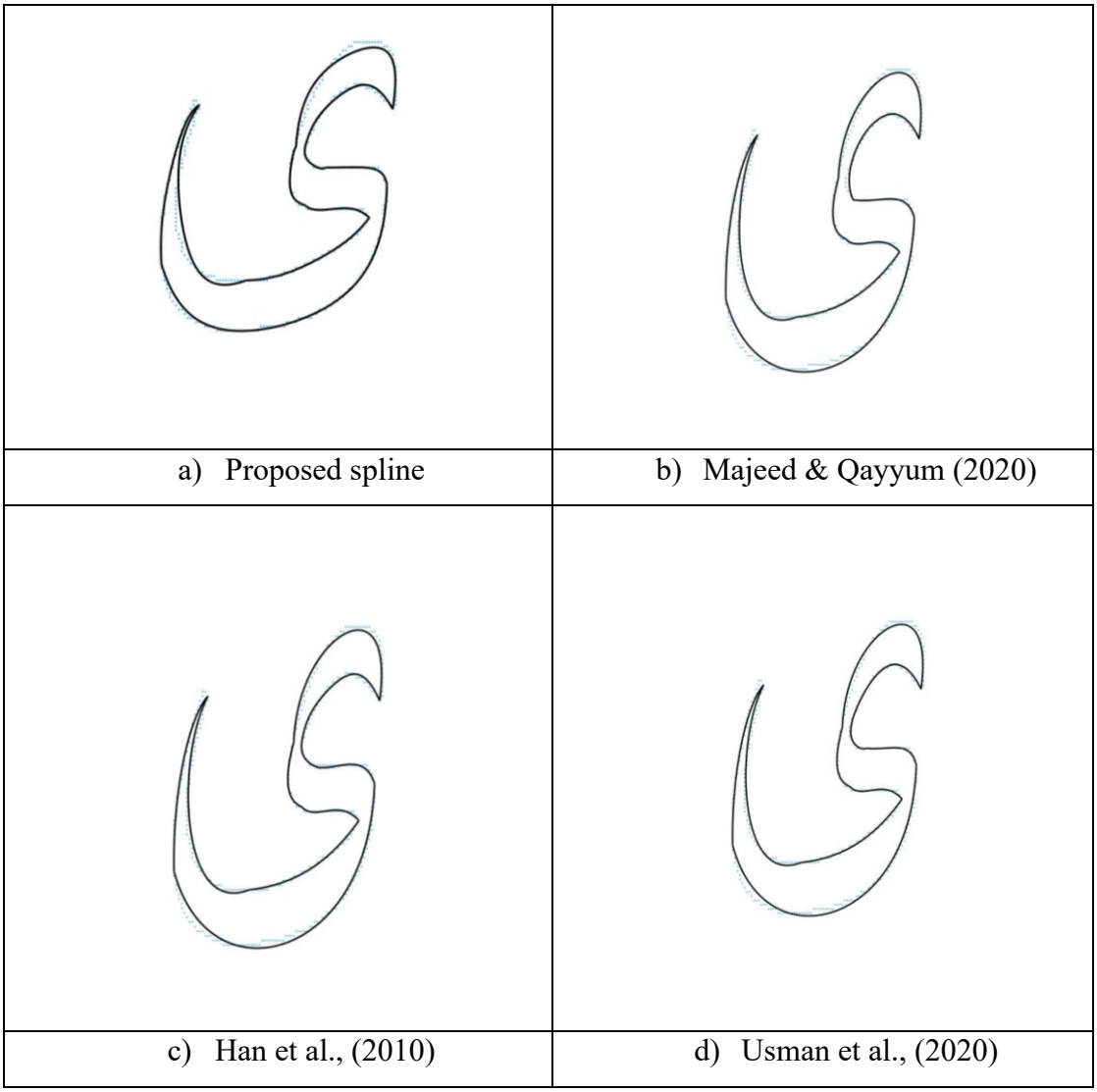
	
a) Proposed spline	b) Majeed & Qayyum (2020)
	
c) Han et al., (2010)	d) Usman et al., (2020)

Figure 6.6 The Visualization of the 'Delta' Result of Cubic Trigonometric Curves (solid line) Over the Boundary (dotted line)



a) Proposed spline | b) Majeed & Qayyum (2020)

c) Han et al., (2010) | d) Usman et al., (2020)

Figure 6.7. The Visualization of 'ى' Result of Cubic Trigonometric Curves (solid line) Over the Boundary (dotted line)

All the cubic trigonometric curves generated achieve C^0 continuity condition. The images obtained in the figures demonstrated that the segments were not linked smoothly and comprised of cusps despite being connected by C^0 continuity.

In essence, concerns like continuity and approximation error will arise when combining two or more segments. The crucial factor that cannot be overlooked in CAGD while producing curves is smoothness. Therefore, the process is continued by applying C^1 continuity condition to the linked segments to address the problem. During the smoothing process with C^1 continuity, the boundary's corner points could vary. In order to retain the corner points, the smoothing procedure is carried out segmentally in between the corners using the equation in (4.6). Consequently, the segments' endpoints

need to be adjusted to comply with the continuity requirements. That indicates that the control points adjustment process will solely affect only the control points of V_3 of the first curve and V_0 for the next curve.

The flowchart in Figure 6.8 presents the procedure for transforming each curve segment from C^0 continuity, to C^1 continuity, which also ensures a smooth transition in slope between consecutive segments. Meanwhile, the Figure 6.9 and Figure 6.10 illustrates the example changes in the two curve segments before and after applying the C^1 continuity condition.

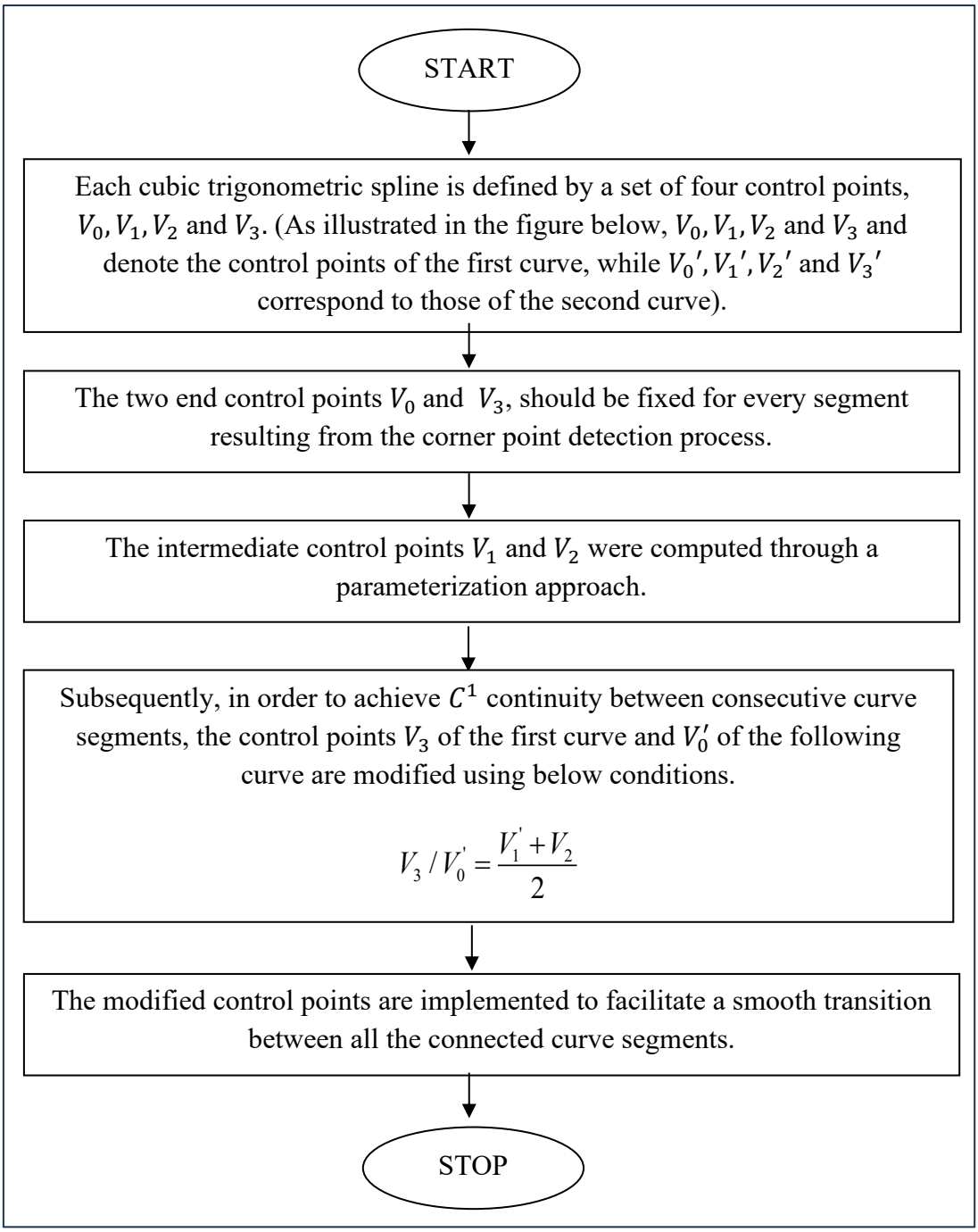


Figure 6.8 Continuity-preserving Curve-fitting Process

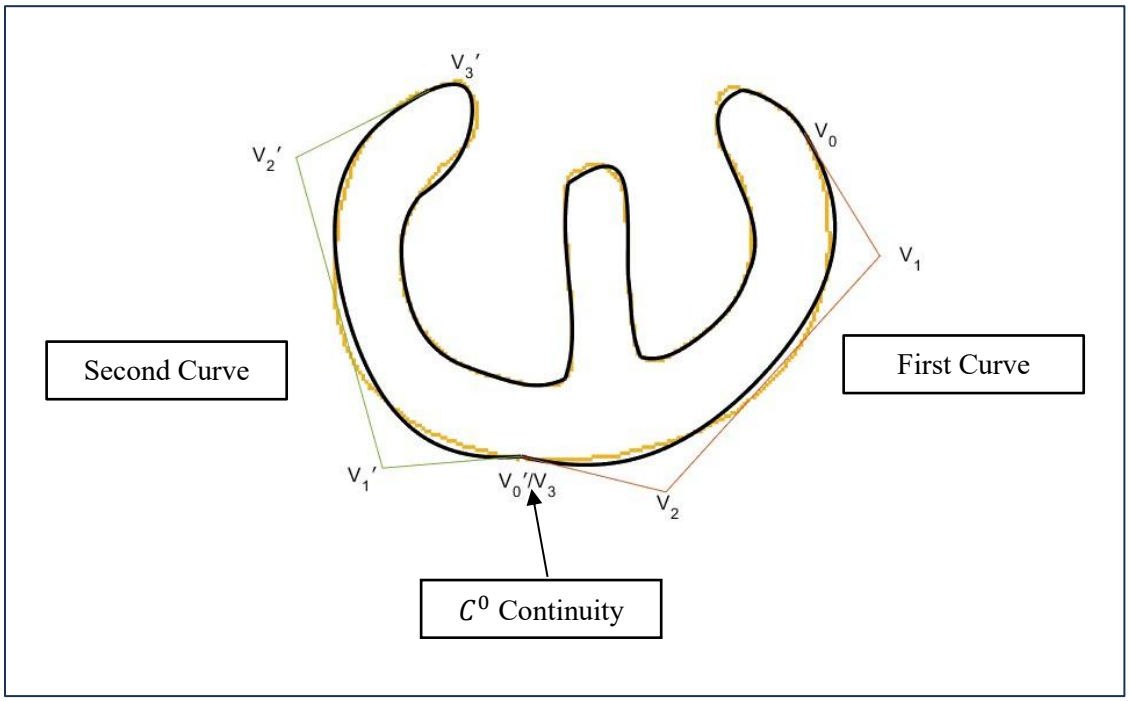


Figure 6.9 Epsilon with C^0 continuity

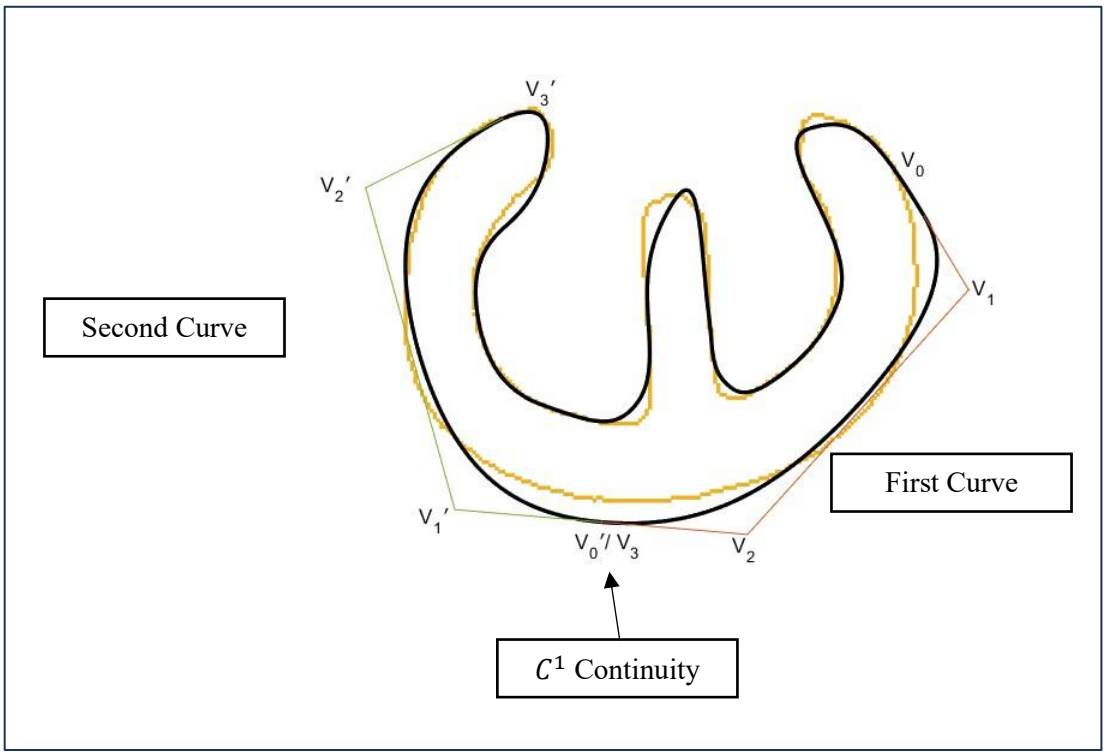


Figure 6.10 Epsilon with C^1 continuity

Next. the results before and after the curve fitting process for all segments with C^0 continuity and C^1 continuity is presented. The regenerated curves for all segments exhibit continuity and enhanced smoothness, resulting in an improved image. After the C^1 continuity curve fitting process, the approximation error is once again recalculated. By applying continuity due to maintain the smoothness of the fitted curve, the distance from the actual data points and the fitted curve could be increases and the error will be greater than before. The fitted cubic trigonometric spline before and after applying C^1 continuity conditions for the letter 'epsilon', 'delta' and 'ى' are illustrated in Figure 6.11 – 6.22.

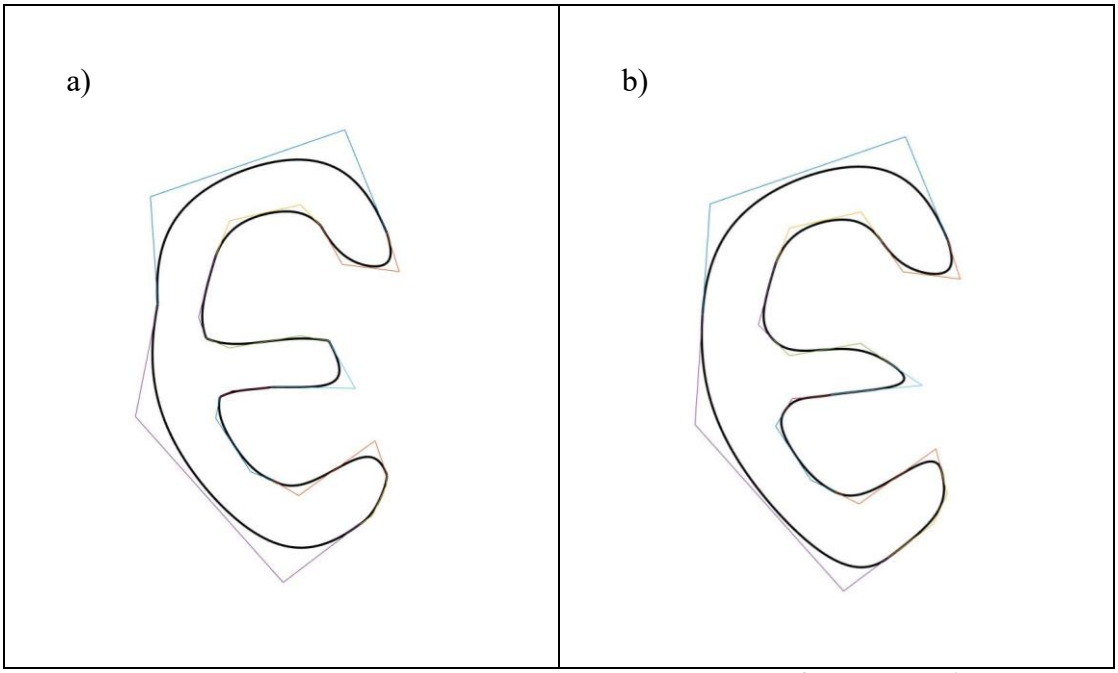


Figure 6. 11 The Visualization of 'Epsilon' Applying (a) C^0 and (b) C^1 Continuity Using Fitted Proposed Spline

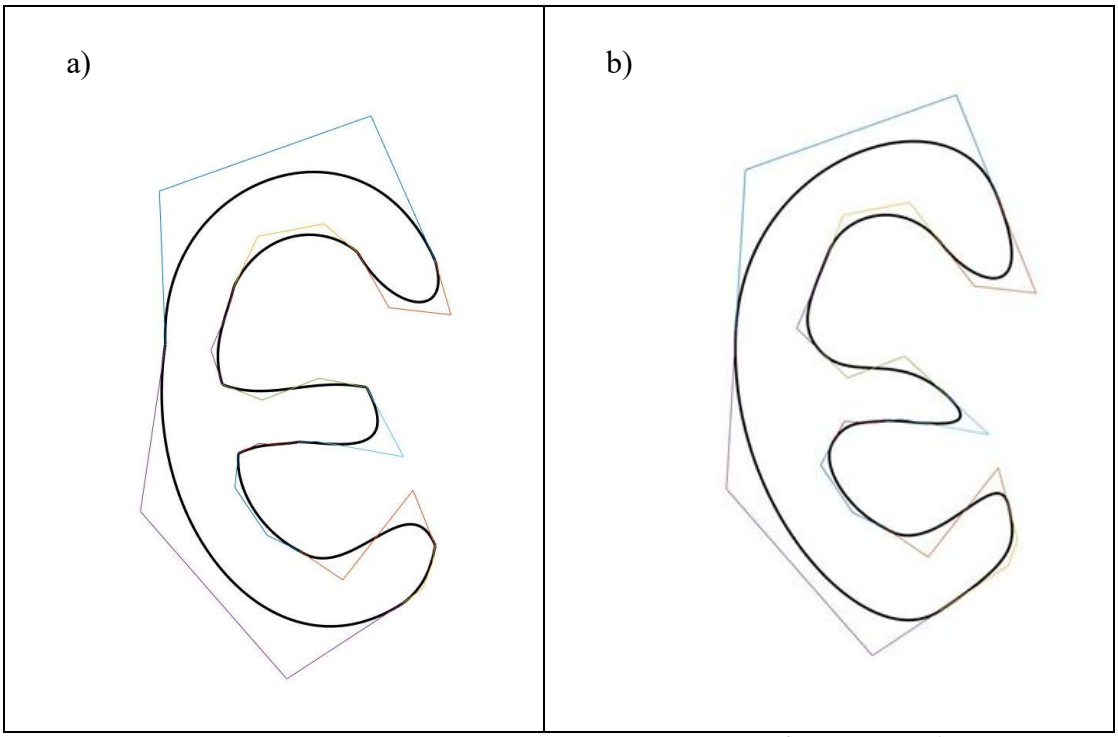


Figure 6. 12 The Visualization of 'Epsilon' Applying (a) C^0 and (b) C^1 Continuity Using Majeed and Qayyum Spline

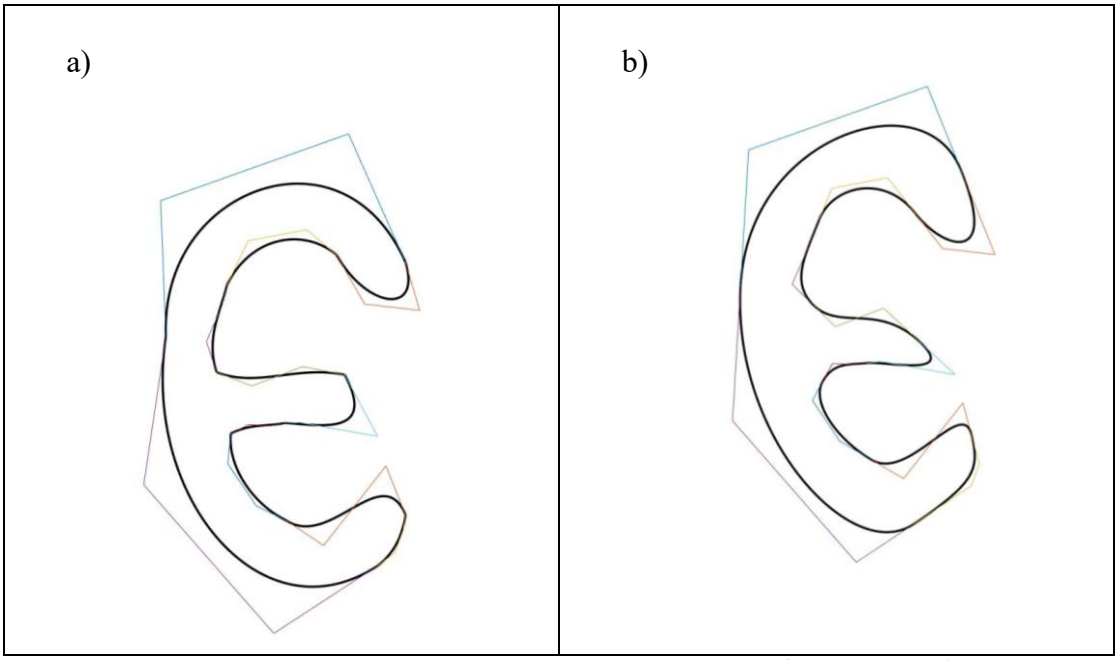


Figure 6. 13 The Visualization of 'Epsilon' Applying (a) C^0 and (b) C^1 Continuity Using Han Spline

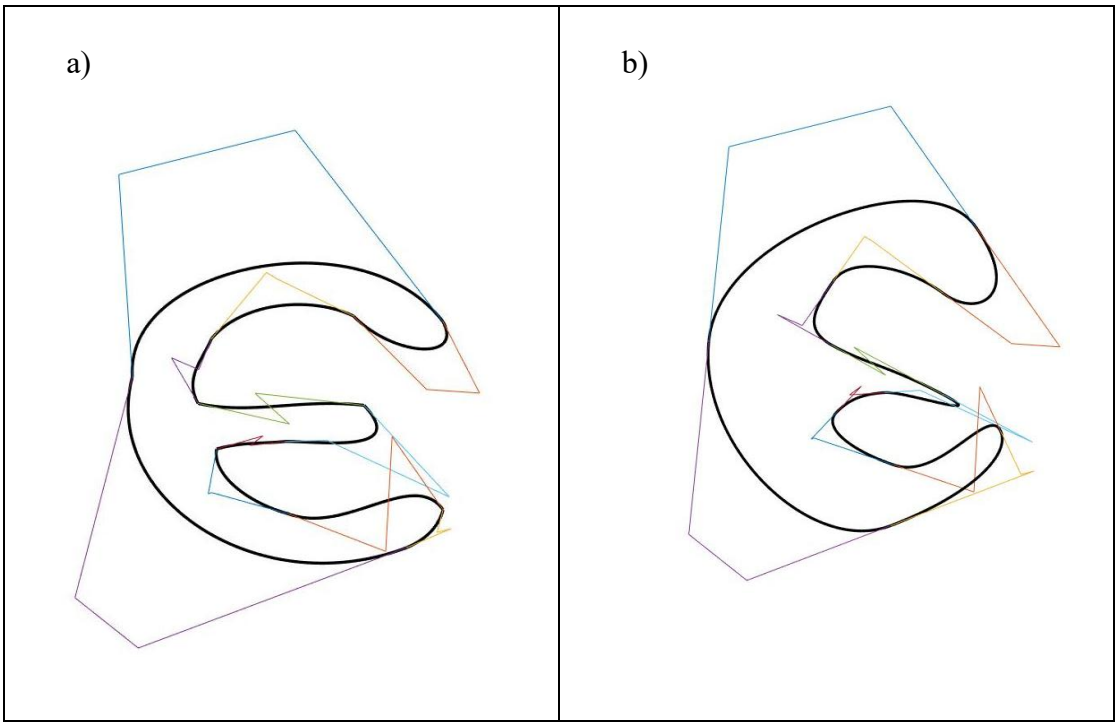


Figure 6. 14 The Visualization of 'Epsilon' Applying (a) C^0 and (b) C^1 Continuity Using Usman Spline

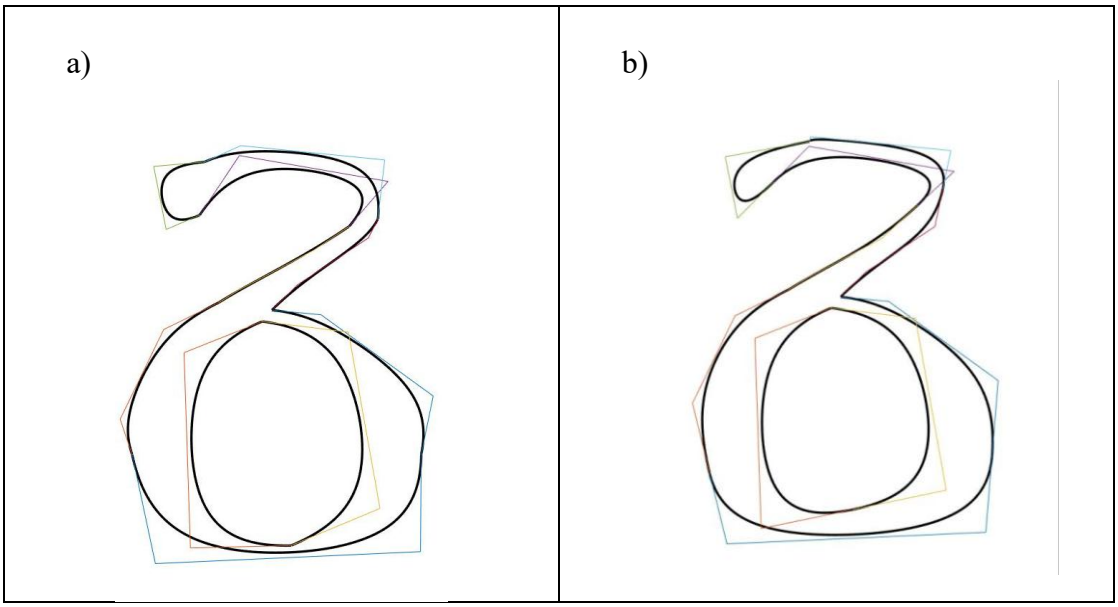


Figure 6. 15 The Visualization of 'Delta' Applying (a) C^0 and (b) C^1 Continuity Using Fitted Proposed Spline

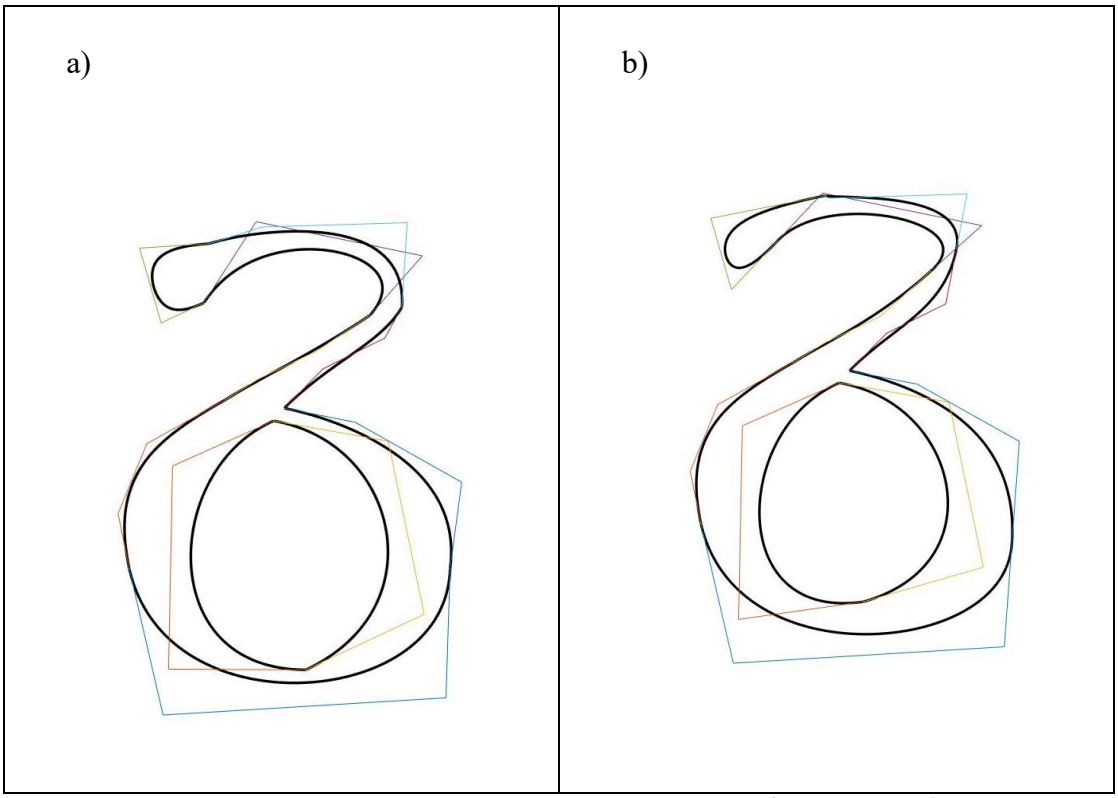


Figure 6. 16 The Visualization of 'Delta' Applying (a) C^0 and (b) C^1 Continuity Using Majeed and Qayyum Spline

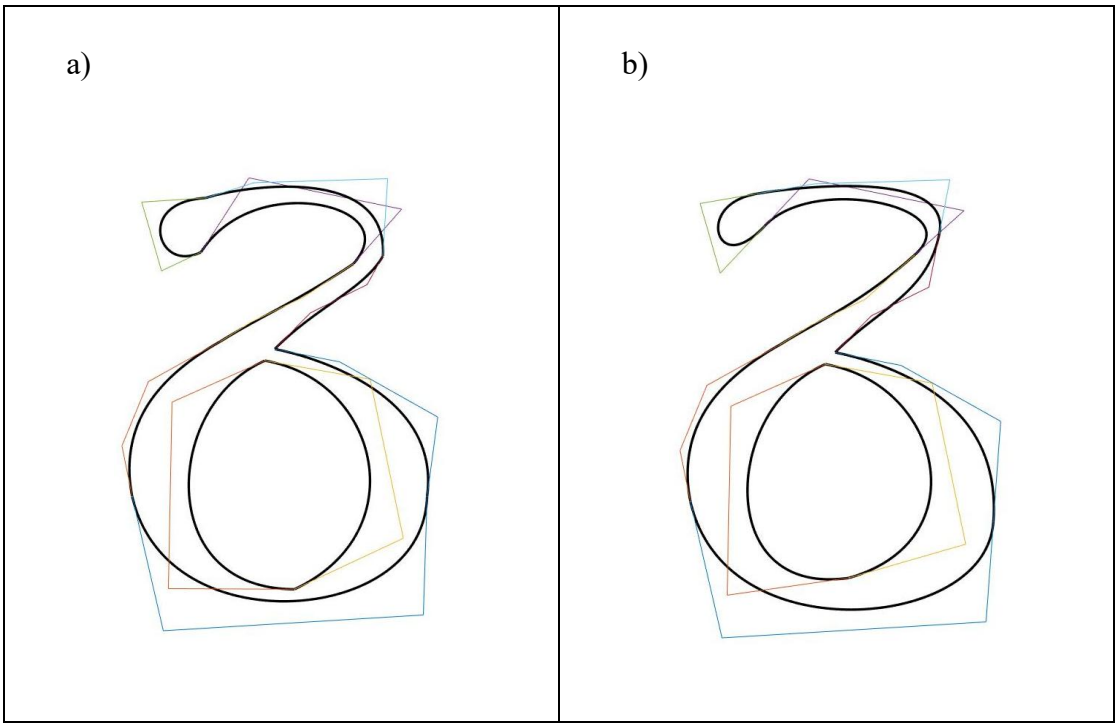


Figure 6. 17 The Visualization of 'Delta' Applying (a) C^0 and (b) C^1 Continuity Using Han Spline

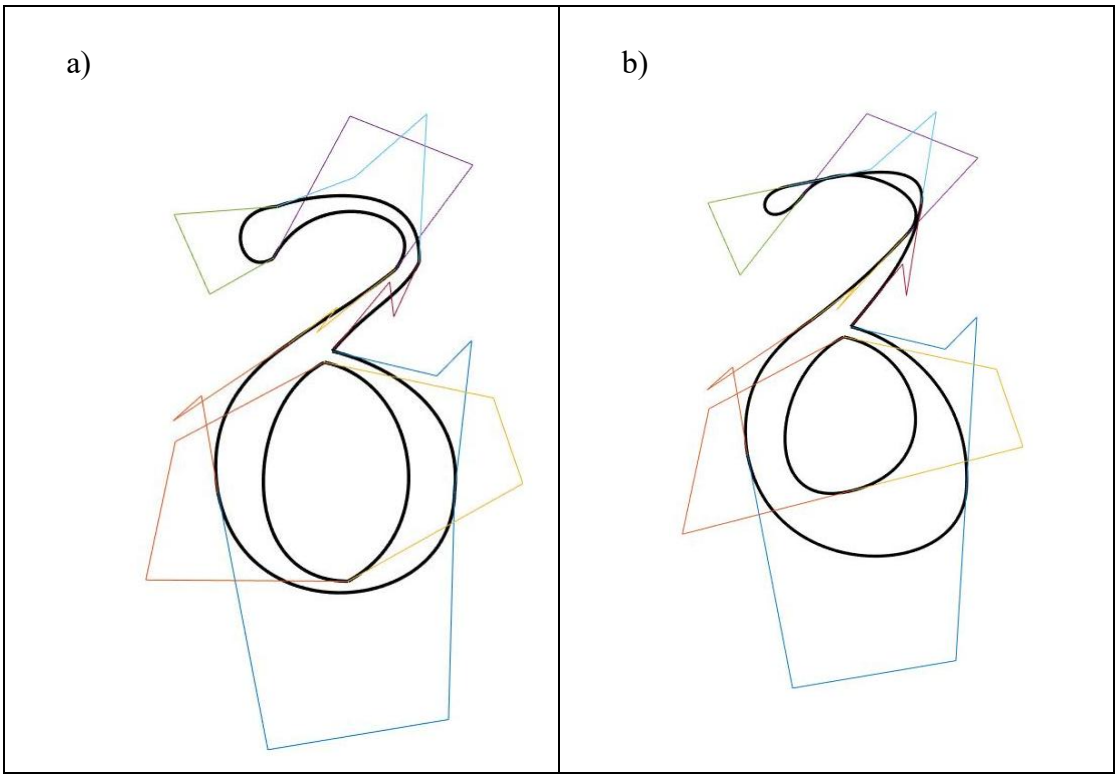


Figure 6. 18 The Visualization of 'Delta' Applying (a) C^0 and (b) C^1 Continuity Using Usman Spline

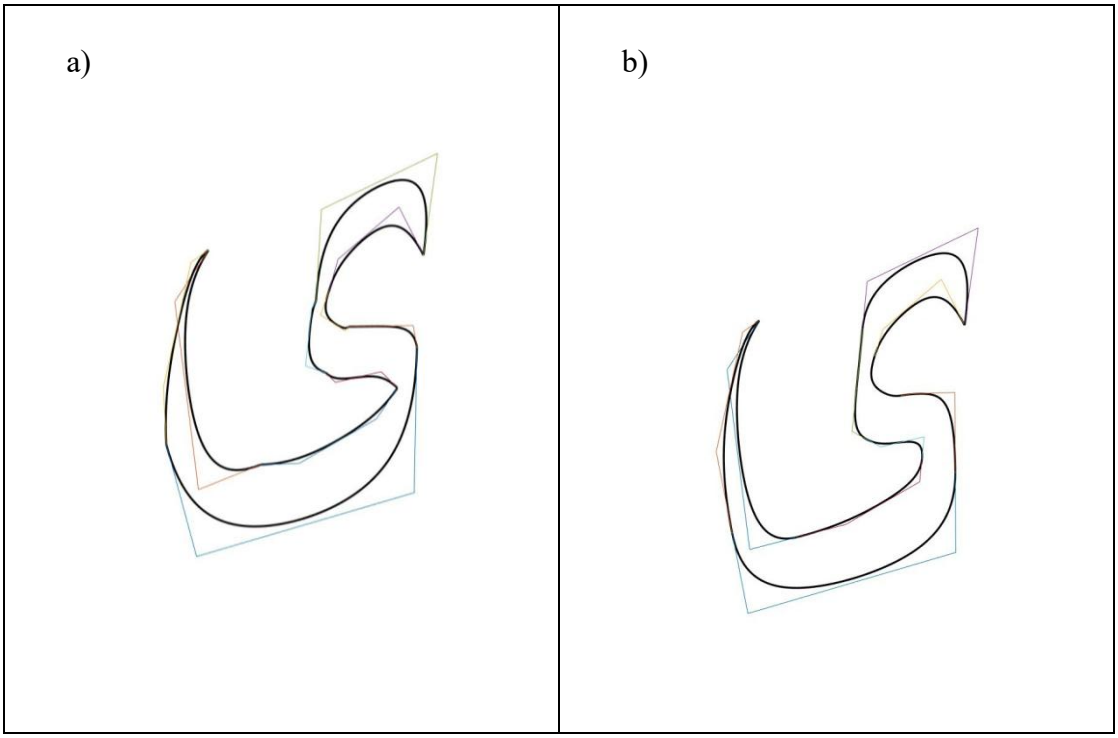


Figure 6. 19 The Visualization of 'ى' Applying (a) C^0 and (b) C^1 Continuity Using Fitted Proposed Spline

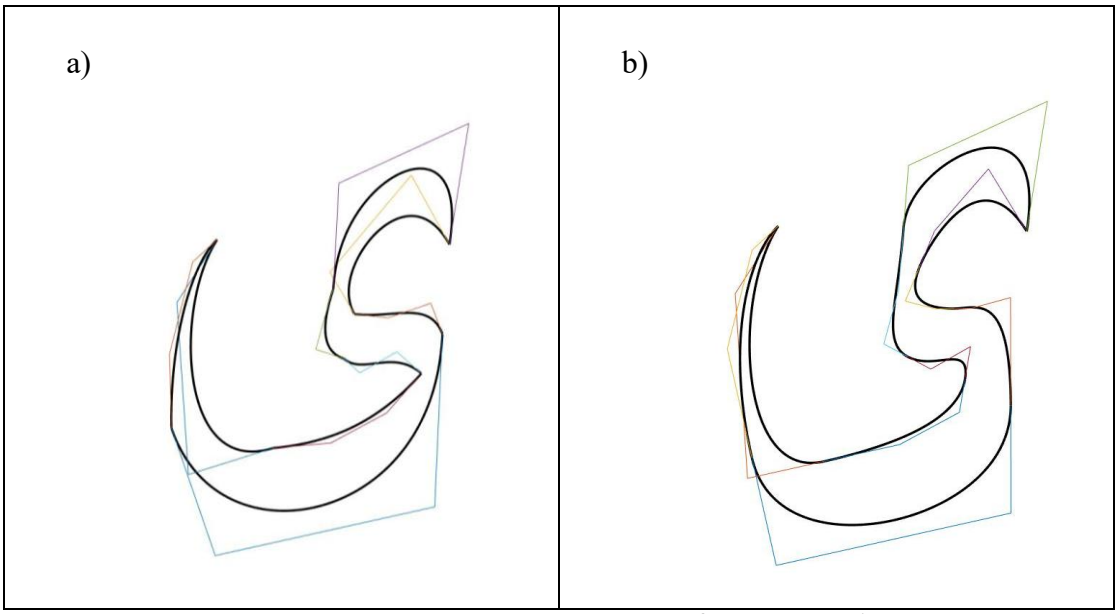


Figure 6. 20 The Visualization of 'ى' Applying (a) C^0 and (b) C^1 Continuity Using Majeed and Qayyum Spline

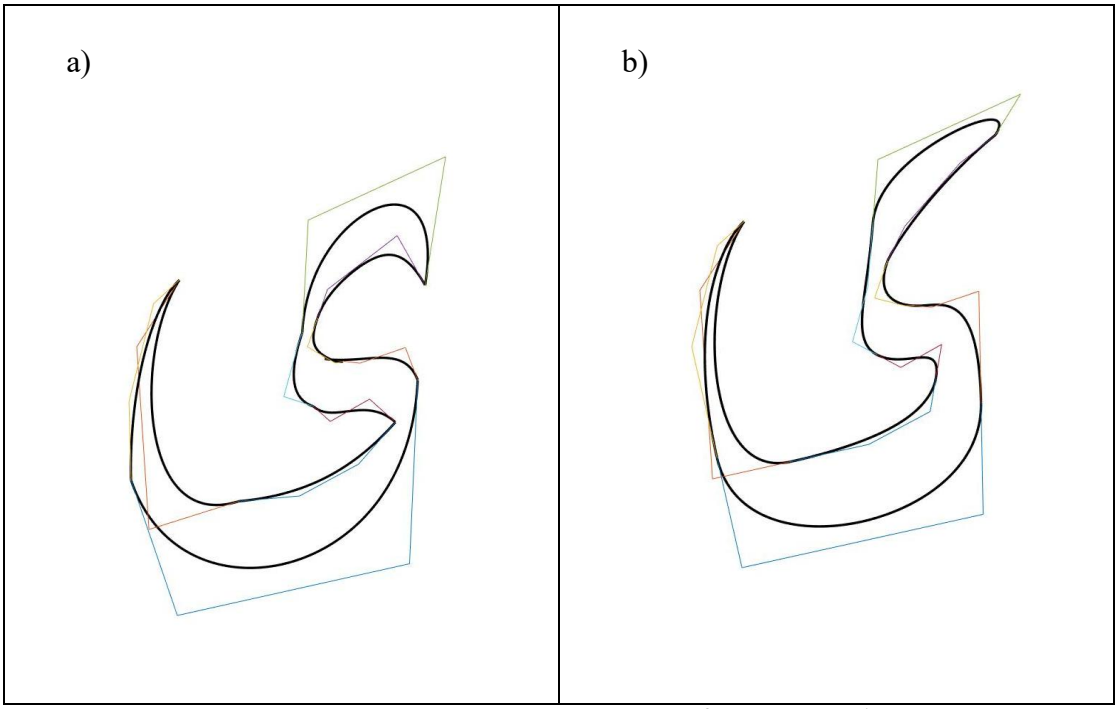


Figure 6. 21 The Visualization of 'ى' Applying (a) C^0 and (b) C^1 Continuity Using Han Spline

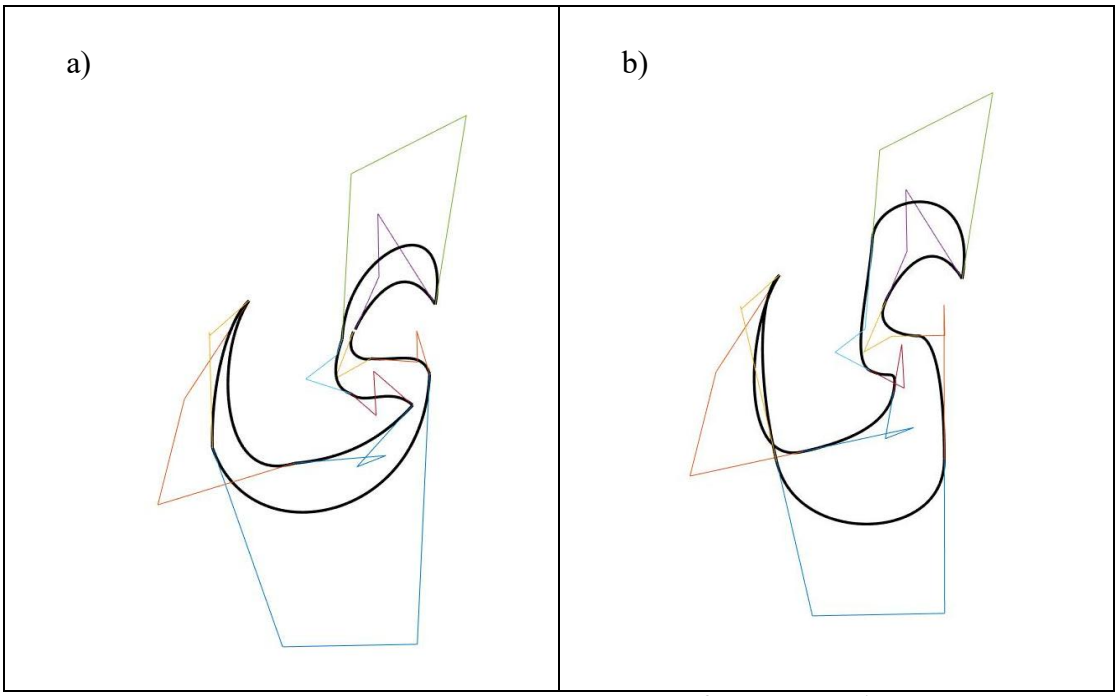


Figure 6. 22 The Visualization of 'ى' Applying (a) C^0 and (b) C^1 Continuity Using Usman Spline

In Figure 6.11 – 6.22, the refinement process is done segmentally using C^1 continuity. The jagged points at the connections between the curve segments have been refined. Nevertheless, from the figures, it can be observed that some of the image produced have changes in shape compared to real image due to tighten in maintaining the continuity condition.

6.7 Approximation Error

The ultimate goal is to obtain the best possible curve approximation with the least number of errors. Reduction in the boundary spacing between the fitted cubic trigonometric spline and the actual image's border are required. The achievement of the goals and the method's efficacy must be determined through a thorough result analysis. So, evaluations of results need to be conducted by computing the total, maximum, minimum, average error, along with the number of bad points.

First, the accuracy is computed by calculating the sum of the squared distance between every contour segment point and the fitted curve. The formula is listed in equation (6.7) and (6.19).

Error

$$= \sqrt{(X_{contour\ point} - X_{fitted\ curve})^2 + (Y_{contour\ point} - Y_{fitted\ curve})^2} \quad (6.19)$$

For each segment corresponding to the letter's 'epsilon', ε, 'delta', δ, and 'ى', the total error was computed before and after refinement process. The maximum and minimum errors were determined from the highest and lowest squared distances of the data points within each segment, while the average error was calculated by dividing the total error by the number of data points in the segment.

Concurrently, in this research, a-bad point detection threshold was established to improve the reliability of the results by reducing the influence of outliers. The threshold value was set at 1.5, derived from the average error value obtained during preliminary analyses. This selection ensures that the criterion is data-driven rather than arbitrarily chosen, thereby aligning the filtering process with the underlying characteristics of the dataset. Data points with squared distance values exceeding this threshold are classified as bad points, as they deviate significantly from the expected range. The total number of bad points is then determined using Equation (6.7), applied exclusively to instances where the values surpass the established threshold. This approach enhances the robustness of the proposed method by systematically identifying and limiting the impact of outlying data points.

6.8 Results and Discussions

The process of reconstruction has been done for all four tested cubic trigonometric splines. The region of interest's outlines is first taken out of the 2D image. Segmentation and curve fitting are the two key components of the methodology. The process of segmentation entails digitizing the image, extracting the image's boundary or contour, and then obtaining equidistant data points to represent the boundary. The outlines' corner data points are recognized. The contoured outline is divided into sections at the corners in this initial stage.

The following step was to fit curves to these segments. A precise determination of the corner points is essential for a suitable fitting or portrayal of the design. Corner points are interpolated during curve fitting that is connected by using C^0 continuity and the remaining values are approximated. The cubic trigonometric spline must enhance

the order of continuity in order to guarantee the smoothness of the spline. All the fitted curves are undergoing the refinement process from C^0 continuity to the C^1 continuity. Piecewise fitting is performed between the segments of cubic trigonometric spline using the iteratively computed C^1 continuity. The value of the shape parameter and the scalar factor is determined to produce curves that are as near to the digitized data points as possible. The newly fitted cubic trigonometric spline illustrates that the curves are generated in a slight deviation from the initial data points to meet the requirement of continuity condition. Considering the values of the errors and bad points is carried out to assess the new cubic trigonometric spline interpolation's capabilities.

A comparison with other approaches has been conducted to analyse the advantages and disadvantages of the method. In comparison research, the benefits of the proposed method with regards to shape preservation along with minimal approximation error are examined. A visual comparison is a useful tool that can draw attention to elements such as the smoothness, continuity of the curve, and proximity to the original curve. But because human observers' assessments vary, a more quantitative comparison is necessary. The quantitative comparison includes the total error, maximum error, average error, and number of bad points.

Tables and figures below show the results of approximation errors for all points in 'epsilon', 'delta', and 'ى' before and after applying C^1 continuity constraint. The initial total approximation error for each segment is listed. After enforcing C^1 continuity, the tables show the updated total error with the maximum, minimum, and average error values per segment. Next, the comparative study of the proposed algorithm with another algorithm is presented.

6.8.1 Approximation Error of Epsilon by Using Proposed Cubic Trigonometric Spline

Table 6. 2
Approximation Error of 'Epsilon' Before and After C^1 Refinement (I)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	357.9594	495.0956	7.4168	0.000017697	3.3228
2	104.1973	109.9929	4.4105	0	0.5555
3	62.4566	83.151	3.3995	0	0.3274
4	33.8805	58.382	4.5623	0	0.202
5	61.1541	131.6706	6.6356	0	0.3741
6	476.8053	527.4576	16.32943	0	1.2181
7	24.1441	38.3557	3.6659	0	0.0836
8	45.3022	51.7642	3.2658	0	0.1037
9	88.7681	139.1813	6.5541	0	0.2468
10	22.8352	98.808	19.3285	0	0.1686
11	378.0329	946.2525	18.9332	0	1.2718
TOTAL	1656.338	2680.111	19.3285	0	0.71585

From Table 6.2, the total error before and after the refinement process for each segment is presented. The total error is increasing for most segments due to the smoothing spline process in achieving C^1 continuity. This is expected, as enforcing derivative continuity introduces additional constraints on the curve segments, reducing their local flexibility and slightly increasing the approximation error. The total error increases from 1656.338 (before) to 2680.11 (after), reflecting the global effect of enforcing smoothness across segment boundaries. Furthermore, it can be seen that a large total error produces is in segment 1, 6 and 11. This is cause by the longer curve segments which carry on a lot of number of data points compared to the other segments. Thus, the approximation error is higher.

6.8.2 Approximation Error of Epsilon by Using Majeed & Qayyum (2020)

Table 6. 3
Approximation Error of 'Epsilon' Before and After C^1 Refinement (II)

	Before	After			
Segment	**Total error**	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	209.6435	1001.2	19.0253	1.7426	6.7195
2	98.6398	282.333	17.7035	0	1.4259
3	57.4607	166.4388	6.8512	0	0.6553
4	34.2501	128.0876	6.962	0	0.4432
5	59.4750	238.7918	10.8625	0	0.6784
6	425.2028	466.1095	10.3963	0	1.0765
7	23.7724	57.807	5.5218	0	0.1259
8	46.1168	61.8071	5.1949	0	0.1239
9	94.9671	193.5294	9.2428	0	0.3431
10	22.3928	158.6423	23.6804	0	0.2707
11	195.8320	1137.6	23.1676	0	1.5291
TOTAL	1267.753	3892.3465	23.6804	0	1.2174

Table 6.3 demonstrates a significant increase in total error for selected segments, particularly segments 1,2,3,4,5, and 11 which show a substantial increase in total error after the C^1 continuity condition is applied. The total error has an increment of 2624.5935 before and after the refinement continuity process. The maximum error for 'epsilon' generated by Majeed & Qayyum (2020) is 23.6804, the minimum error is 0, and the average error is 1.2174. Despite the increase in total error, the average error per segment remains relatively small (mostly below 1.6). This suggests that although certain points were affected more strongly, the general accuracy of the fitting process is retained.

6.8.3 Approximation Error of Epsilon by Using Han et al., (2010)

Table 6. 4
Approximation Error of 'Epsilon' Before and After C^1 Refinement (III)

	Before	After			
Segment	**Total error**	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	209.6435	1001.2	19.0253	1.7426	6.7195
2	98.6398	282.333	17.7035	0	1.4259
3	57.4607	166.4388	6.8512	0	0.6553
4	34.2501	128.0876	6.962	0	0.4432
5	59.4750	238.7918	10.8625	0	0.6784
6	425.2028	466.1095	10.3963	0	1.0765
7	23.7724	57.807	5.5218	0	0.1259
8	46.1168	67.4147	5.1949	0	0.1351
9	179.0367	274.5967	9.2486	0	0.4869
10	22.3928	158.6423	23.6804	0	0.2707
11	195.832	195.832	2.555	0	0.2632
TOTAL	1354.8226	3037.253	23.6804	0	1.1164

Table 6.4 illustrates the total differences between the approximation error generated using C^0 and C^1 continuity is 1682.430. Segments 1, 2, 3, 4, 5, 6, and 10 show a noticeable rise in total error. Others, such as Segment 11, maintain identical error values before and after correction, suggesting that this segment fits smoothly without requiring much adjustment. The maximum error for 'epsilon' generated by Han et al., (2010) is 23.6804, the minimum error is 0, and the average error is 1.1164. The largest total error produce is in segment 1.

6.8.4 Approximation Error of Epsilon by Using Usman et al., (2020)

Table 6. 5
Approximation Error of 'Epsilon' Before and After C^1 Refinement (IV)

	Before	After			
Segment	**Total error**	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	209.6421	2750.7	38.0506	9.8958	18.4607
2	98.6398	690.648	36.7091	0	3.4881
3	57.4607	371.1324	13.0291	0	1.4612
4	34.2496	347.9703	13.924	0	1.204
5	59.4759	639.2028	21.557	0	1.8159
6	425.2029	729.2476	19.1775	0	1.6842
7	23.7724	137.2118	11.0436	0	0.2989
8	46.1173	216.6469	10.674	0	0.4342
9	94.9672	459.7925	17.7619	0	0.8152
10	22.3927	479.4005	47.3609	0	0.8181
11	195.8316	3533.3	46.8435	0	4.749
TOTAL	1267.7522	10355.25	47.3609	0	3.2027

The changes in the total approximation error for C^0 and C^1 continuity is 9087.4978 as shown in Table 6.5. The maximum error for 'epsilon' generated by Usman et al., (2020) is 47.3609, the minimum error is 0, and the average error is 3.2027. The significant total error produce is in segment 1 and 11. Most other segments (2–10) also show increased total error but at a much smaller scale compared to segments 1 and 11.

6.8.5 Approximation Error of Delta by Using Proposed Cubic Trigonometric Spline

Table 6. 6.
Approximation Error of 'Delta' Before and After C^1 Refinement (I)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	141.5587	309.5481	9.9797	0.984	4.0201
2	42.4129	110.2862	8.9967	0	0.9507
3	29.035	32.8849	2.8767	0	0.2418
4	82.261	133.5404	6.8328	0	0.7419
5	26.96	55.9364	7.9997	0	0.2729
6	46.0953	109.7259	6.8436	0	0.4389
7	29.4026	47.5273	4.3402	0	0.1728
8	55.7599	68.9589	5.249	0	0.2142
9	85.8882	162.4827	4.5003	0	0.2962
10	73.4268	136.736	4.7335	0	0.253
TOTAL	612.8004	1167.627	9.9797	0	0.7603

From Table 6.6, it can be seen that the difference in the total error generated with C^0 and C^1 is increased 554.8266. The maximum error for the 'delta' generated is 9.9797, the minimum error is 0, and the average error is 0.7603. The largest total error produce is in segment 1.

6.8.6 Approximation Error of Delta by Using Majeed & Qayyum (2020)

Table 6. 7
Approximation Error of 'Delta' Before and After C^1 Refinement (II)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	91.2319	150.7111	4.9804	0.0000021245	1.9573
2	41.0577	41.0577	1.5837	0	0.3539
3	28.1238	33.4442	2.8011	0	0.2459
4	71.2479	142.5347	7.8591	0	0.7919
5	25.0503	89.7952	10.3444	0	0.438
6	46.1071	164.3414	9.0895	0	0.6574
7	28.5380	58.8094	5.5426	0	0.2139
8	50.2004	72.4707	5.1428	0	0.2251
9	78.4830	180.323	6.2505	0	0.4588
10	69.2675	155.6566	4.7335	0	0.3391
TOTAL	529.3076	1089.144	10.3444	0	0.56813

Table 6. 7 shows the increment is 559.8364 before and after applying C^1 continuity. The maximum error for the 'delta' generated by Majeed & Qayyum (2020) is 10.3444, the minimum error is 0, and the average error is 0.56813. The largest total error produce is in segment 9.

6.8.7 Approximation Error of Delta by Using Han et al., (2010)

Table 6. 8
Approximation Error of 'Delta' Before and After C^1 Refinement (III)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	91.2319	91.2319	2.1642	0.0000021245	1.1848
2	41.0577	41.0577	1.5837	0	0.3539
3	28.1238	33.4442	2.8011	0	0.2459
4	71.2479	142.5347	7.8591	0	0.7919
5	25.0503	57.4026	6.5397	0	0.28
6	46.1071	126.2494	6.396	0	0.505
7	28.5380	58.8094	5.5426	0	0.2139
8	50.2004	50.2004	1.7382	0	0.1559
9	78.4830	106.1247	4.5698	0	0.27
10	69.2675	136.1097	4.7335	0	0.2965
TOTAL	529.3076	843.1647	7.8591	0	0.42978

Table 6. 8 demonstrates the changes in the total error before and after achieving C^1 is 313.8571. The maximum error for the 'delta' generated by Han et al., (2010) is 7.8591, the minimum error is 0, and the average error is 0.42978. The largest total error produce is in segment 4.

6.8.8 Approximation Error of Delta by Using Usman et al., (2020)

Table 6. 9
Approximation Error of 'Delta' Before and After C^1 Refinement (IV)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	91.2322	325.5034	10.1218	0.0000021245	4.2273
2	41.0577	41.0577	1.5837	0	0.3539
3	28.1235	48.3729	5.6022	0	0.3557
4	71.2480	342.3324	15.0718	0	1.9018
5	25.0501	156.2226	13.7281	0	0.7621
6	46.1066	308.9782	12.7919	0	1.2359
7	28.5380	131.5139	11.8967	0	0.4782
8	50.2007	173.4914	10.2857	0	0.5388
9	78.4833	320.0175	13.5227	0	0.8143
10	69.2677	397.1 403	13.7702	0	0.8652
TOTAL	529.3078	2244.63	15.0718	0	1.15332

Based on Table 6.9, it can be seen that after the refinement process, to C^1 continuity, the difference in the total is 1715.3222. The maximum error for the 'delta' generated by Usman et al., (2020) is 15.0718, the minimum error is 0, and the average error is 1.15332. The largest total error produce is in segment 10.

6.8.9 Approximation Error of ى by Using Proposed Cubic Trigonometric Spline

Table 6. 10
Approximation Error of 'ى' Before and After C^1 Refinement (I)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	113.0892	356.9793	12.1674	0.4145	3.7185
2	19.5361	58.7405	8.8759	0	0.4978
3	59.4833	63.5272	10.1813	0	0.4925
4	35.8813	35.5181	1.6842	0	0.2234
5	93.4662	165.3142	8.2501	0	0.7761
6	15.6935	36.3029	5.9621	0	0.1585
7	20.2486	46.4707	5.9060	0	0.1851
8	40.5197	62.6693	4.9884	0	0.2139
9	96.6149	130.9642	4.5908	0	0.3742
10	42.1683	78.4305	6.8865	0	0.2006
TOTAL	536.7011	1034.917	12.1674	0	6.8406

From Table 6.10, it can be seen that the total error generated with C^0 and C^1 is increased 498.2159. The maximum error for 'ى' generated is 12.1674, the minimum error is 0, and the average error is 6.8404. The largest total error produce is in segment 1.

6.8.10 Approximation Error of ى by Using Majeed & Qayyum (2020)

Table 6. 11
Approximation Error of 'ى' Before and After C^1 Refinement (II)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	124.4578	498.1909	14.8214	1.2737	5.1895
2	19.2314	89.1325	12.0679	0	0.7554
3	10.9854	18.8434	3.3378	0	0.1461
4	33.0674	29.3101	1.3781	0	0.1843
5	76.6799	184.4066	9.6449	0	0.8658
6	15.5319	49.0306	7.8419	0	0.2141
7	19.9345	70.4380	8.5914	0	0.2806
8	40.6686	89.8398	7.6905	0	0.3066
9	72.6383	114.7545	4.5452	0	0.3279
10	41.8743	105.5041	8.0203	0	0.2698
TOTAL	455.0695	1249.451	14.8214	0	8.5401

Table 6. 11 illustrates the total difference between the approximation error generated using C^0 and C^1 continuity is 794.3715. The maximum error for 'ى' generated by Majeed & Qayyum (2020) is 14.8214, the minimum error is 0, and the average error is 8.5401. The largest total error produce is in segment 1.

6.8.11 Approximation Error of ى by Using Han et al., (2010)

Table 6. 12
Approximation Error of 'ى' Before and After C^1 Refinement (III)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	124.4578	494.393	14.6171	1.2688	5.1499
2	23.2276	113.9	12.3192	0	0.9336
3	13.6513	12.9981	1.6135	0	0.1008
4	49.253	140.0871	16.9426	0	0.8811
5	76.6799	361.4399	16.4936	0	1.6969
6	15.5319	49.0306	7.8419	0	0.2141
7	19.9345	70.438	8.5914	0	0.2806
8	40.6686	89.8398	7.6905	0	0.3066
9	72.6383	114.7545	4.5452	0	0.3279
10	41.8743	105.5041	8.0203	0	0.2698
TOTAL	477.9172	1552.385	14.6171	0	10.1613

Table 6. 12 shows the total change between the approximation error generated using C^0 and C^1 continuity is 1074.4678. The maximum error for 'ى' generated by Han et al., (2010) is 14.6171, the minimum error is 0, and the average error is 10.1613. The largest total error produce is in segment 1.

6.8.12 Approximation Error of 'ى' by Using Usman et al., (2020)

Table 6. 13
Approximation Error of 'ى' Before and After C^1 Refinement (IV)

Segment	Before	After			
	Total error	**Total error**	**Max. error**	**Min. error**	**Ave. error**
1	124.4556	1514.3	29.6426	9.5175	15.7739
2	19.2316	254.005	26.7491	0	2.1526
3	10.9852	39.1918	6.8649	0	0.3038
4	33.0673	23.9085	1.3557	0	0.1504
5	76.6799	404.3404	19.2897	0	1.8983
6	15.5315	126.3935	17.3158	0	0.5519
7	19.9342	186.3955	17.1828	0	0.7426
8	40.6690	350.6236	16.2707	0	1.1967
9	72.6381	227.3553	9.0692	0	0.6496
10	41.8742	260.4301	16.0406	0	0.6661
TOTAL	455.0666	3386.944	29.6426	0	0.240859

From Table 6.13, the increase of in total error before and after generating cubic trigonometric spline curves is 2931.8774. The maximum error for 'epsilon' generated by Usman et al., (2020) is 29.6426, the minimum error is 0, and the average error is 0.240859. The largest total error produce is in segment 1.

6.8.13 Results Analysis and Comparative Study

This section displays the output outcomes including the image representations and the numerical results. The performance is compared between the proposed scheme and other methods. Table 6.14 -Table 6.16 show the error analysis for all four functions. The results are shown visually and numerically illustrated in Figure 6.23, Figure 6.24 and Figure 6.25 and Table 6.14, Table 6.15, and Table 6.16, respectively.

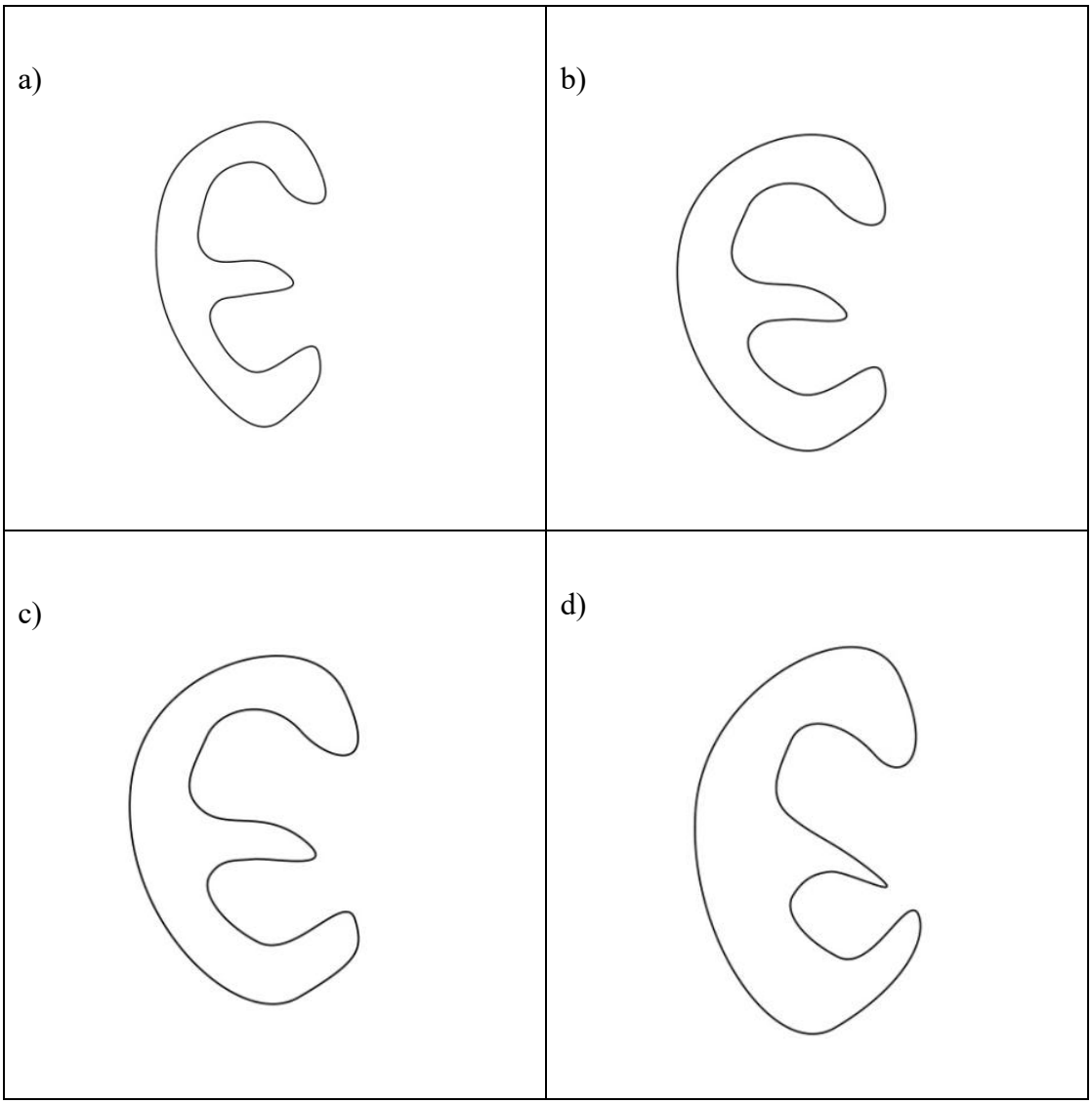


Figure 6. 23 Representations of ‘Epsilon’ Using Different Cubic Trigonometric (a) Proposed Spline, (b) Majeed Spline , (c) Han Spline and (d) Usman Spline

Table 6. 14
Estimation Error for Different Representations of ‘Epsilon’

	Proposed spline	**Majeed (2020)**	**Han (2010)**	**Usman (2020)**
Total Error	**2680.111**	3892.3465	3037.253	10355.28
Max. Error	**19.3285**	23.6804	23.6804	47.3609
Min. Error	**0**	0	0	0
Ave. Error	**0.71585**	1.2174	1.1164	3.2027
Bad points (>1.5)	**511**	657	585	726

Figure 6.23 illustrates a comparison between the proposed cubic trigonometric spline and the other three different cubic trigonometric splines. The data shown in

Table 6. 14 indicates that the suggested spline exhibits the smallest average error as well as bad points. This is followed by Han et al., (2010), Majeed & Qayyum (2020), and Usman et al., (2020). The outcomes of the proposed spline exhibited superior results compared to the others. In visual, all the images showed the continuity and smoothness of the curve; for Usman et al., (2020), some parts of the curves are not curvy.

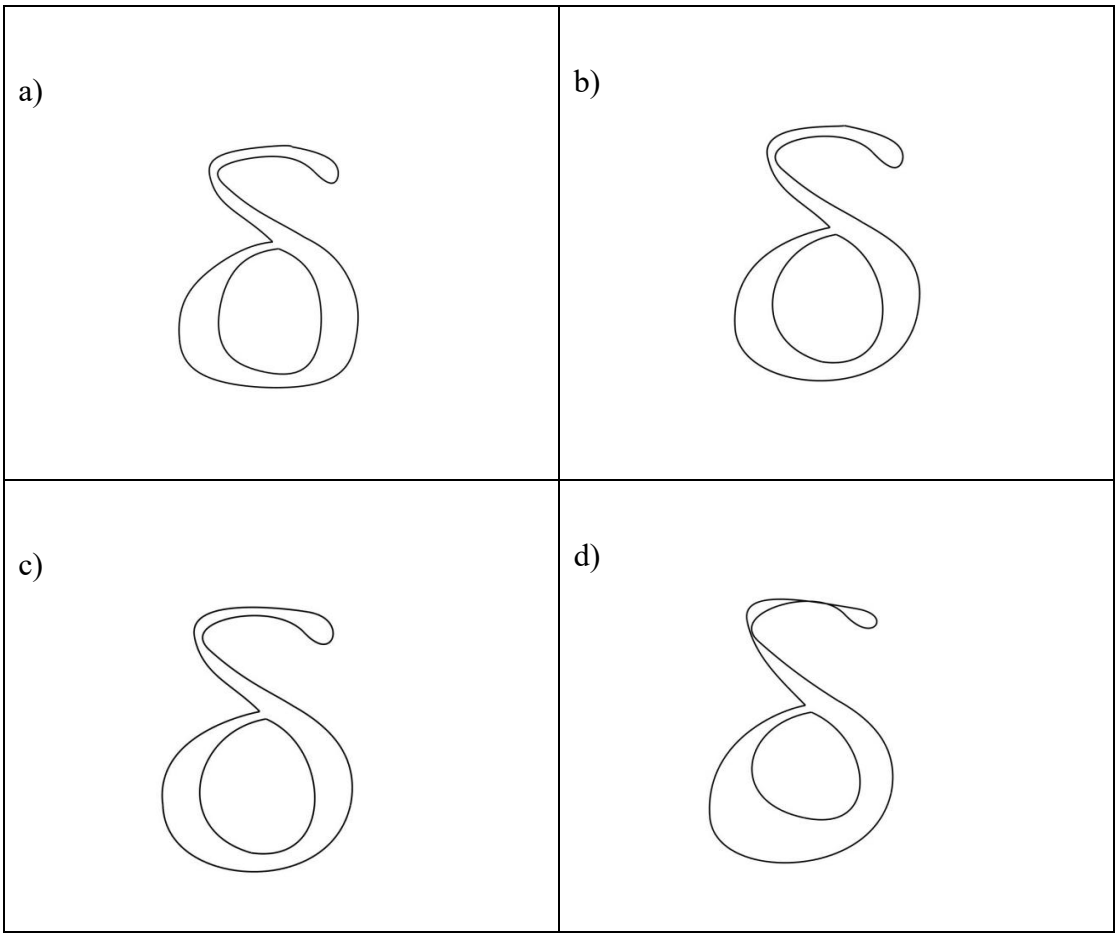


Figure 6. 24 Representations of 'Delta' Using Different Cubic Trigonometric (a) Proposed Spline, (b) Majeed Spline , (c) Han Spline and (d) Usman Spline

Table 6. 15
Estimation Error for Different Representations of ‘Delta’

	Proposed spline	**Majeed (2020)**	**Han (2010)**	**Usman (2020)**
Total Error	1167.627	1089.144	**843.1647**	2244.63
Max. Error	9.9797	10.3444	**7.8591**	15.0718
Min. Error	0	0	**0**	0
Ave. Error	0.7603	0.56813	**0.42978**	1.15332
Bad points (>1.5)	259	271	**222**	354

Figure 6.24 and Table 6.15 show that Han et al., (2010) shows a better performance which gives the lowest average error, followed by the proposed spline, Majeed & Qayyum (2020), and Usman et al., (2020). Han et al., (2010) has the best accuracy in reconstructing the outline of the ‘delta’. In some parts, the outline image of Usman et al., (2020) looks to overlap.

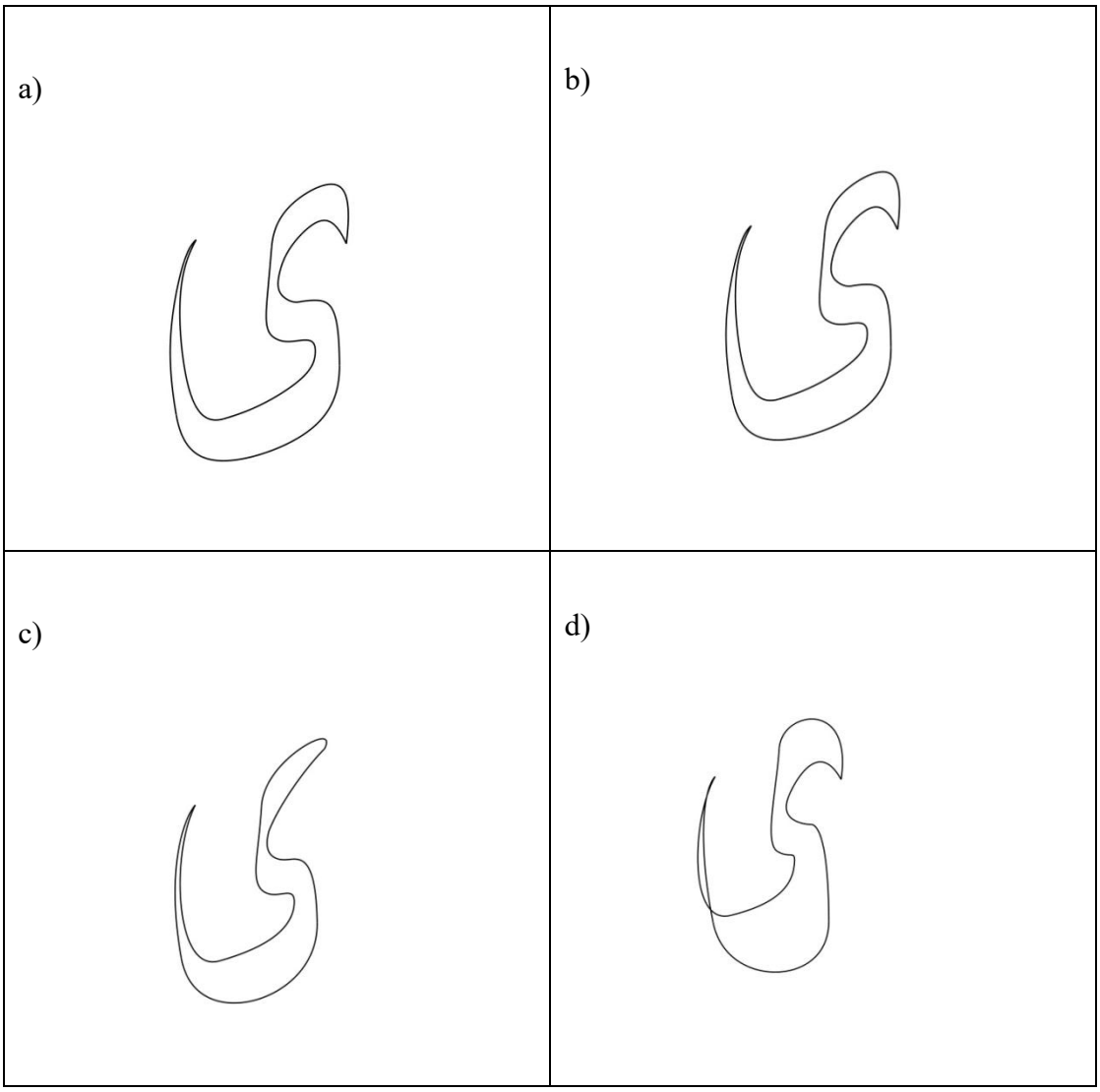


Figure 6. 25 Representations of 'ى' Using Different Cubic Trigonometric (a) Proposed Spline, (b) Majeed Spline , (c) Han Spline and (d) Usman Spline

Table 6. 16
Estimation Error for Different Representations of 'ى'

	Proposed spline	**Majeed (2020)**	**Han (2010)**	**Usman (2020)**
Total Error	**1034.917**	1249.451	1552.385	3386.9437
Max. Error	**12.1674**	14.8214	14.6171	29.6426
Min. Error	**0**	0	0	0
Ave. Error	**0.68406**	0.85401	1.01613	2.40859
Bad points (>1.5)	**249**	265	303	335

As can be seen from Figure 6.25 and Table 6.16, for outline ى, the proposed spline produced a minor error and the smallest number of bad points compared to the others Usman et al., (2020) showed the largest average error produced. The outline image of Usman et al., (2020) showed an overlap in visuals.

Based on all the presented results above, Usman et al., (2020) showed the largest value of error and bad points in contrasting with the remaining three splines. The possible cause is the shortcoming of the basis spline that results in a significant distance among the curve with the control polygon as shown in Figure 6.5. Consequently, this results in significant errors in the depiction of the curve. In addition, due to maintaining the continuity and smoothness at the joints, the image of 'delta' and 'ى' reconstructed by Usman et al., (2020) performed overlap at certain joining segments. In an ideal situation, the fitted curve would be smooth and would accurately reproduce the image's contours.

Table 6. 17
The Best Scheme According to Error Analysis

Shape	**Epsilon**	**Delta**	**ى**
Best algorithm	The proposed scheme	Han et al., (2010)	The proposed scheme
Average error	0.71585	0.42978	0.68406
Bad points	511	222	249

By utilizing the proposed spline to reconstruct 'epsilon' and 'ى', both show the least value of approximation errors when compared to the other splines. Han et al., (2010) only gives the best-fitted curve (lowest approximation errors) for reconstructing the delta. Therefore, the comparative study demonstrates that the proposed spline provides consistently accurate reconstructions, confirming its effectiveness for modelling letter shapes. In other words, the proposed spline effectively achieves the desired goal, which is to reduce the borders' distance measurement as determined by the original image and the fitted cubic trigonometric spline.

6.9 Summary

This chapter reproduces a mathematical sketch of a 2D object image of the alphabet letters based on contours by using four different cubic trigonometric splines. The splines were utilized throughout the entire process of curve fitting, which began with data collecting and continuing till error computation. Throughout the process, the approximation errors are computed and displayed for each segment of reconstructed images. Base on the outcome, several segments exhibited a significantly higher number of total errors in comparison to the remaining segments. This circumstance arises due to the selection of larger segments that encompass significant number of data points. The greater number of data points, the larger the computation of approximation error. To address this concern, it is necessary to partition the large segments into a number of smaller segments. By increasing the total number of curve segments will decrease the approximation error and significantly enhance the precision of the fitted curves. As the approximation error decreases, the fitted curve appears closely aligned with the actual data.

Next, from the result, the findings indicate that the suggested cubic trigonometric spline demonstrated strong reconstruction capability and delivers accurate results for tested font design. The proposed method has worked comparatively well. The proposed trigonometric spline maintains stable behaviour and produces smooth approximations and precise visualisation with minor errors. By comparing with the existing spline formulations, the proposed approach shows improvements in terms of accuracy, robustness, and reliability. The proposed spline generated curves that were most precise to the real image and nearest to the control polygon.

As illustrated visually, all generated shapes of the letter epsilon, delta, and ی that are performed by four different cubic trigonometric splines are smooth and continuous except for some overlapping cases at a certain part of segments. These overlapping cases occurred due to maintaining and achieving the continuity of the curves.

CHAPTER 7
RESEARCH SUMMARY AND FUTURE WORK

7.1 Introduction

This chapter represents the last part of this thesis. The overview and reflections of the entire study are presented. This chapter also includes suggestions and recommendations for upcoming scholars.

7.2 Research Summary

This thesis dealt with the development of trigonometric functions with a shape parameter and their application in designing parametric curves and surfaces as well as handling the reconstruction process. Two types of new basis functions, quadratic and cubic trigonometric, with one shape parameter have been proposed. Both basis spline functions have been developed which share the primary geometric properties of the traditional Bezier curve. These quadratic and cubic trigonometric basis functions with the properties and their proofs were discussed. Although quadratic trigonometric curves are computationally efficient, they have certain drawbacks. The quadratic trigonometric curves were not created at the endpoints. Therefore, cubic trigonometric curves have been preferred in this research, since the degree is high enough to satisfy specific limitations besides the ability to form curves at the endpoints. Due to the fact that this research inherits the properties of trigonometric splines, it is certain to have shape preservation, shape adjustability, and high-order continuity, making it appropriate for curve and surface design.

An approach is employed to generate the curves in addition to the control points-based cubic trigonometric Bezier-like curves. This has been accomplished by using parametric and geometric continuity, C^0, C^1, C^2, G^0, G^1 and G^2. The continuity requirements have been achieved. Two situations included in the thesis. The first condition requires that the shape parameter, m constant for connecting two segments,

whereas the second condition allows the use of different values of m for each segment. The preferred method for representing complex curves that are challenging to depict with a single curve of high-degree is through the use of piecewise representation. These conditions give designer flexibility in achieving desired curve. Additionally, the presence of shape parameter and the scalar factors assisting the designer in customizing the curve's shape. The necessity of the shape parameter and the scaling factor is crucial since their presence allows and provides the curve and surface alteration easily and conveniently. As shown in this research, by only modifying and determining the values, the desired shape can be obtained.

In this work, a trigonometric function application for shape-preserving scientific data visualization has also been investigated. The development of a piecewise C^0, C^1, C^2, G^0, G^1 and G^2cubic trigonometric spline is employed to establish positive curve interpolation techniques. Several schemes are developed in order to face the shape-preserving problems. The schemes have been tested with two different data which is one of the data is near zero dataset. The formulation has been derived to address the challenges of maintaining the original shape and preserve the characteristic of the data. This representation includes one shape parameter and the scalar factors to preserve the shape characteristics of typical functional 2D data. The proposed approach not only retains the fundamental qualities of the data, but also produces a smooth curve and visually appealing outcomes.

Different curves and surfaces can be created by employing mathematical concepts. The mathematical techniques used in this study will be useful for industrial applications. For instance, they will help produce appealing 3D shapes and objects, which assist designers in producing highly desirable curves and surfaces. The smooth connection between the surface patches created by the parametric and geometric continuity requirements implemented throughout this research led to the construction of two and more developable surfaces for this study.

Next, an algorithm for curve fitting using C^1 cubic trigonometric spline is developed. The algorithm begins with data acquisition until the curve fitting process. This study exploits the proposed cubic trigonometric spline as the fitted curve. The least number of control points from the data points have been taken into account by applying the LSM. The distance between the control points makes it easy to visualize how the curve moves when the shape parameter values change. The best feature of this method is the ability to alter images using shape parameters. The approximation error is then

used to gauge the fitted curve's accuracy. Typically, the distance between a data point and its associated curve point is used to compute the approximation error.

The constructed curve fitting algorithm employing the proposed cubic trigonometric spline is demonstrated in the results chapter. The algorithms then are applied on 2D fonts of 'epsilon', 'delta', and 'ى'. The curve fitting technique has been applied and compared with the other three existing cubic trigonometric splines. These four algorithms are executed on the same platform to get the total approximation error. Lastly, the result is compared visually and numerically. The purposes are to have a representation that is accurate, consistent, and flexible.

Based on the review analysis that have been made, the suggested cubic trigonometric shows that the proposed spline method outperforms compared to the previous methods due to its high accuracy. In visual, the reconstructed font by using the proposed cubic trigonometric spline can provide accurate representations that are continuous, smooth, and pleasant curves that do not form overlap specifically. According to the evaluation of each of the performances, the construction of basis functions of the spline must take into consideration the properties of the spline which the distance of the generated curve must be drawn nearer to the control polygon. This factor gives affects the performance of the curve in the design and reconstruction process.

Overall, this study show that the proposed spline consistently produces smooth and accurate shapes, demonstrating that a simpler formulation that have only one shape parameter can offer both efficiency and strong performance.

7.3 Future Work

Future study might focus on constructing more intricate curves and surfaces that inherit all the Bezier-like properties while ensuring the smooth connections using the generated continuity conditions. Besides that, it should be further investigating the numerical stability, curvature behaviour, and error performance on trigonometric spline formulations, particularly through systematic quantitative evaluations against classical polynomial schemes. Furthermore, it is suggested that future researchers apply the proposed cubic trigonometric spline for 3D surface fitting for real data using higher number of continuity condition which are $C^1, \dots, C^n$ or $G^1, \dots G^n$. With the recent finding of mathematical components in the construction of curves and surfaces, it is

hoped that more attractive and unique designs and algorithms can be developed yet being useful to the industries.

REFERENCES

Abbas, M., Majid, A. A., & Ali, J. M. (2013). A rational spline for preserving the shape of positive data. *International Journal of Computer and Electrical Engineering*, *5*, 442-446.

Ai, H., Yang, C., Wu, S., Zeng, Y., & Bo, S. (2010). Automatic segmentation and 3D reconstruction of human liver based on CT image. *2010 4th International Conference on Bioinformatics and Biomedical Engineering, ICBBE 2010*, 1–4.

Amat, N. F. I. C., Yahya, Z. R., Rusdi, N. A., Helmee, N. A., & Muhamad, W. Z. A. W. (2019). Arabic fonts representation in cubic Bézier curve using different soft computing algorithm. *IOP Conference Series: Materials Science and Engineering*, *705*(1). 1-9.

Ameer, M., Abbas, M., Abdeljawad, T., & Nazir, T. (2022). A novel generalization of Bézier-like curves and surfaces with shape parameters. *Mathematics, 10*(376). 1-19.

Ammad, M., & Misro, M. Y. (2020). Construction of local shape adjustable surfaces using quintic trigonometric Bézier curve. *Symmetry, 12*(8). 1-21.

Ammad, M., Misro, M. Y., Abbas, M., & Majeed, A. (2021). Generalized developable cubic trigonometric Bézier surfaces. *Mathematics*, *9*(3), 1–17.

Barsky, B. A., & DeRose, T. D. (1989). Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics and Applications*, *9*(6), 60–69.

Bashir, U., & Ali, J. M. (2014). A rational trigonometric spline to visualize positive data. *AIP Conference Proceedings*, *1605*, 286-291.

Bashir, U., Abbas, M., & Ali, J. M. (2013). The G^2 and C^2 rational quadratic trigonometric Bézier curve with two shape parameters with applications. *Applied Mathematics and Computation*, *219*(20), 10183–10197.

Bercovier, M., & Volpin, O. (1999). G^1 hierarchical Bézier surface over arbitrary meshes. *Computer Graphics Forum*, *18*(4), 223–236.

Bibi, S., Abbas, M., Misro, M. Y., & Hu, G. (2019). A novel approach of hybrid trigonometric Bézier curve to the modeling of symmetric revolutionary curves and symmetric rotation surfaces. *IEEE Access*, *7*, 165779–165792.

Bibi, S., Abbas, M., Miura, K. T., & Misro, M. Y. (2020). Geometric modeling of novel generalized hybrid trigonometric Bézier-Like curve with shape parameters and its applications. *Mathematics*, *8*(6), 1-30.

Bibi, S., Misro, M. Y., Abbas, M., Majeed, A., & Nazir, T. (2021). G^3 shape adjustable GHT-Bézier developable surfaces and their applications. *Mathematics*, *9*(19), 1–23.

Brodlie, K. W., Asim, M. R., & Unsworth, K. (2005). Constrained visualization using the shepard interpolation family. *Computer Graphics Forum*, *24*(4), 809–820.

Chambers, J. B., Wang, Z., Cooke, R. A., & Black, M. M. (1996). A comparison of valve resistance, the continuity equation, and the Gorlin formula against directly observed orifice area in bioprosthetic valves in the mitral position: an in vitro study. *The Journal of Heart Valve Disease*, *5*(2), 136-143.

Chambers, P., Hagen, H., & Mclnerney, T. (1995). Introduction to Curves and Surfaces. In *Graphics Gems V*.

Choubey, N., & Ojha, A. (2008). Trigonometric splines with variable shape parameter. *Rocky Mountain Journal of Mathematics*, *38*(1), 91–105.

Chu, C. H., & Chen, J. T. (2004). Geometric design of developable composite Bézier surfaces. *Computer-Aided Design and Applications*, *1*, 531-539.

Dejdumrong, N., & Tongtar, S. (2007). The generation of G^1 cubic Bézier curve fitting for Thai consonant contour. *Geometric Modelling and Imaging, GMAI 2007, Proceedings*, 48–53.

Diao, L., Zhang, S., Liu, L., Li, H., & Lin, Z. (2010). Analysis of conversion matrix between geometrically continuous splines and Bézier representation. *5th International Conference Pervasive Computing and Applications (ICPCA),* 291-293.

Dube, M., & Rana, P. S. (2014). Positivity preserving interpolation of positive data by rational quadratic trigonometric spline. *IOSR Journal of Mathematics*, *10*(2), 42–47.

Dube, M., & Sharma, R. (2013). Quartic trigonometric Bézier curve with a shape parameter. *International Journal of Mathematics and Computer Applications Research (IJMCAR), 3*(3), 89-86.

Dube, M., & Yadav, B. (2014). The quintic trigonometric Bézier curve with single shape parameter. *International Journal of Scientific and Research Publications, 4*(3), 1-4.

Farin, G. (2002). *Curves and Surfaces for CAGD* (R. Fiske, Ed.; Fifth). Morgan Kaufmann.

Faris, A. K., Yahya, Z. R., Rusli, N., & Rusdi, N. (2016). Rational cubic spline for preserving the positivity of 3D positive data. *American Institute of Physics, 1775,* 1-9.

Ferris, L. W. (1968). A Standard Series of Developable Surfaces. *Marine Technology and SNAME News*, *5*(01), 52–62.

Gerald Farin. (2002). *Curves and surfaces for CAGD: A practical guide* (5th ed.). Morgan Kaufmann.

Hadi, N. A. (2014). G^2 *Parametric Curve and Surface Fitting Using Beta Spline*. Universiti Teknologi Mara.

Hadi, N. A., & Alias, N. (2020). The Performance of 3D Multi-slice Branched Surface Reconstruction on CPU-GPU Platform. *Proceedings of the Second International Conference on Science, Engineering and Technology (ICoSET 2019),* 49-54.

Halim, M. S. A., Hadi, N. A., & Julius, R. (2024). Trigonometric triangular spline surface: A systematic review. *AIP Conference Proceedings*, *2905*(1), 1-11.

Han, X. (2002). Quadratic trigonometric polynomial curves with a shape parameter. *Computer Aided Geometric Design*, *19*(7), 503–512.

Han, X. (2004). Cubic trigonometric polynomial curves with a shape parameter. *Computer Aided Geometric Design*, *21*(6), 535–548.

Han, X. (2005). C^2 quadratic trigonometric polynomial curves with local bias. *Journal of Computational and Applied Mathematics*, *180*(1), 161–172.

Han, X. (2015). Shape-preserving piecewise rational interpolant with quartic numerator and quadratic denominator. *Applied Mathematics and Computation*, *251*, 258–274.

Han, X. A., Huang, X., & Ma, Y. (2010). Shape analysis of cubic trigonometric Bézier curves with a shape parameter. *Applied Mathematics and Computation*, *217*(6), 2527–2533.

Han, X. A., Ma, Y. C., & Huang, X. L. (2009). The cubic trigonometric Bézier curve with two shape parameters. *Applied Mathematics Letters*, *22*(2), 226–231.

Han, X., & Zhu, Y. (2012). Curve construction based on five trigonometric blending functions. *BIT Numerical Mathematics*, *52*(4), 953–979.

Han, X., & Zhu, Y. (2016). A practical method for generating trigonometric polynomial surfaces over triangular domains. *Mediterranean Journal of Mathematics*, *13*(2), 841–855.

Hasan, Z.A., Yahya, Z. R., Rusdi, N. A., & Roslan, N. (2018). Curve reconstruction in different cubic functions using differential evolution. *MATEC Web of Conferences*, *150*, 1-6.

Heo, H., & Chae, O.-S. (2004). Segmentation of tooth in CT images for the 3D reconstruction of teeth. *Image Processing: Algorithms and Systems III*, *5298,* 455- 466.

Hoschek, J., & Lasser, D. (1993). *Fundamentals of computer aided geometric design*. A K Peters.

Hu, G., Bo, C., Wei, G., & Qin, X. (2020). Shape-adjustable generalized Bézier surfaces: Construction and it is geometric continuity conditions. *Applied Mathematics and Computation*, *378*, 125-215.

Hu, G., Cao, H., Wang, X., & Qin, X. (2017). G^2 continuity conditions for generalized Bézier-like surfaces with multiple shape parameters. *Journal of Inequalities and Applications*, *248*, 1-17.

Hussain, M. Z., & Ali, J.M. (2006). Positivity-preserving piecewise rational cubic interpolation. *MATEMATIKA*, *22*(2), 147-153.

Hussain, M. Z., & Hussain, M. (2011). C^1 positivity preserving scattered data interpolation using rational Bernstein-Bézier triangular patch. *Journal of Applied Mathematics and Computing*, *35*(1–2), 281–293.

Hussain, M. Z., & Sarfraz, M. (2009). Monotone piecewise rational cubic interpolation. *International Journal of Computer Mathematics*, *86*(3), 423–430.

Hussain, M. Z., Sarfraz, M., & Hussain, M. (2010). Scientific data visualization with shape preserving C^1 rational cubic interpolation. *European Journal of Pure and Applied Mathematics*, *3*(2), 194–212.

Karim, S. A. A. (2014). Positivity preserving interpolation by using GC^1 rational cubic spline. *Applied Mathematical Sciences*, *8*(41–44), 2053–2065.

Karim, S. A. A., Saaban, A., Skala, V., Ghaffar, A., Nisar, K. S., & Baleanu, D. (2020). Construction of new cubic Bézier-like triangular patches with application in scattered data interpolation. *Advances in Difference Equations*, *2020*(1), 1-22.

Karim, S. A.A., & Pang, K.V. (2016). Shape preserving interpolation using C^2 rational cubic spline. *Journal of Applied Mathematics*, *2016*(1), 1-14.

Karim, S. A.A., Saaban, A., & Nguyen, V. T. (2021). Construction of C^1 rational bi-quartic spline with positivity-preserving interpolation: numerical results and analysis. *Frontiers in Physics, 9,* 1-18.

Karim, S.A.A. (2018). Construction new rational cubic spline with application in shape preservations. *Cogent Engineering*, *5*(1),1-19.

Karim, S.A.A., Asli, M. F., On, C. K., Mustafa, G., Khan, F., Hasan, M. K., Sulaiman, J., & Kherd, A. (2024). Cubic Bézier-like triangular patches for rainfall scattered data interpolation and visualization. *ITM Web of Conferences, 63*(01016), 1-11.

Kvasov, B. I. (2000). *Methods of Shape-Preserving spline Approximation*. World Scientific Publishing Co. Pte. Ltd.

Levent, A., & Sahin, B. (2018). Cubic Bezier-like transition curves with new basis functions. *Proceedings of the Institute of Mathematics and Mechanics, National Academy of Sciences of Azerbaijan, 44*(2), 222-228.

Li, J., & Pakdemirli, M. (2016). A class of cubic trigonometric automatic interpolation curves and surfaces with parameters. *Mathematical and Computational Applications*, *21*(2), 1-11.

Li, J., Ji, Y., & Zhu, C. (2021). De casteljau algorithm and degree elevation of toric surface patches. *Journal of Systems Science and Complexity*, *34*(1), 21–46.

Li, Q.S., Li, J. P., & Chen, L. (2019). A Bezier curve-based font generation algorithm for character fonts. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems*, 1156-1159.

Mahzir, S. S., & Misro, M. Y. (2023). Shape preserving interpolation of positive and range-restricted data using quintic trigonometric Bézier curves. *Alexandria Engineering Journal*, *80*, 122–133.

Majeed, A., & Qayyum, F. (2020). New rational cubic trigonometric B-spline curves with two shape parameters. *Computational and Applied Mathematics*, *39*(3), 1-24.

Majeed, A., Abbas, M., Sittar, A. A., Kamran, M., Tahseen, S., & Emadifar, H. (2021). New cubic trigonometric Bezier-like functions with shape parameter: Curvature and its spiral segment. *Journal of Mathematics*, *2021*,1-13.

Maqsood, S., Abbas, M., Hu, G., Ramli, A. L. A., & Miura, K. T. (2020). A novel generalization of trigonometric Bézier curve and surface with shape parameters and its applications. *Mathematical Problems in Engineering*, *2020*, 1-25.

Maqsood, S., Abbas, M., Miura, K. T., Majeed, A., BiBi, S., & Nazir, T. (2021). Geometric modeling of some engineering GBT-Bézier surfaces with shape parameters and their applications. *Advances in Difference Equations*, *2021*(1), 1-36.

Maqsood, S., Abbas, M., Miura, K. T., Majeed, A., Hu, G., & Nazir, T. (2021). Shape-adjustable developable generalized blended trigonometric Bézier surfaces and their applications. *Advances in Difference Equations*, *2021*(1), 1-32.

Menasri, R., Oulhadj, H., Daachi, B., Nakib, A., & Siarry, P. (2014). A genetic algorithm designed for robot trajectory planning. *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 228–233.

Peng, L., & Zhu, Y. (2019). A class of trigonometric Bézier basis functions with six shape parameters over triangular domain. *IAENG International Journal of Applied Mathematics, 49*(4), 1-7.

Peng, W., Xu, C., & Feng, Z. (2016). 3D face modeling based on structure optimization and surface reconstruction with B-Spline. *Neurocomputing*, *179*, 228–237.

Piah, A.R.M., Saaban, A., & Majid, A. A. (2006). Range restricted positivity-preserving scattered data interpolation. *Journal of Fundamental Sciences 2,* 63-75.

Piegl,L., & Tiller, W. (1997). *The NURBS book* (2nd ed.). Springer.

Raseli, S. S., & Ali, J. Md. (2012). Boundary extraction of 2D image. *Journal of Basic and Applied Scientific Research*, *2*(5), 5374–5376.

Razali, N. K. B., Draman, N. N. B. C., Nor-Al-Din, S. M. B., & Sukri, N. B. M. (2021). Cubic curve fitting method in reconstruction of Chinese calligraphy outline. *Journal of Physics: Conference Series*, *2084*(1), 1-11.

Romani, L., Saini, L., & Albrecht, G. (2014). Algebraic-trigonometric Pythagorean-hodograph curves and their use for hermite interpolation. *Advances in Computational Mathematics*, *40*(5–6), 977–1010.

Rusdi, N., & Yahya, Z. R. (2015). Reconstruction of generic shape with cubic Bézier using least square method. *AIP Conference Proceedings*, *1660*.

Saini, D., & Kumar, S. (2014). Free-form surface reconstruction from arbitrary perspective images. *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014*, *2*, 1054–1059.

Sarfraz, M., & Khan, M. A. (2002). Automatic outline capture of Arabic fonts. *Information Sciences*, *140*(3–4), 269–281.

Sarfraz, M., Hussain, M. Z., & Hussain, F. (2015). Shape preserving curves using quadratic trigonometric splines. *Applied Mathematics and Computation*, *265*, 1126–1144.

Sarfraz, M., Hussain, M. Z., & Hussain, M. (2012). Shape-preserving curve interpolation. *International Journal of Computer Mathematics*, *89*(1), 35–53.

Sarfraz, M., Hussain, M. Z., & Nisar, A. (2010). Positive data modeling using spline function. *Applied Mathematics and Computation*, *216*(7), 2036–2049.

Sarfraz, M., Irshad, M., & Hussain, M. Z. (2013). Reverse engineering of planar objects using GAs. *Sains Malaysiana*, *42*(8), 1167–1179.

Sarfraz, M., Jabeen, N., Samreen, S., & Hussain, M. Z. (2018). A rational quadratic trigonometric spline with interval shape control. *Proceedings - 2017 14th International Conference on Computer Graphics, Imaging and Visualization, CGiV 2017*, *1*, 8–13.

Schneider, R., & Kobbelt, L. (1999). Discrete fairing of curves and surfaces based on linear curvature distribution. *Curve and Surface Design: Saint-Malo Proceedings*, 371–380.

Sharma, A., Kumar, R., & Mansotra, V. (2016). Proposed stemming algorithm for Hindi information retrieval. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO Certified Organization)*, *3297*(6), 11449–11455.

Sharma, R. (2007). Quasi-quartic trigonometric Bézier curves and surfaces with shape parameters. *International Journal of Innovative Research in Science, Engineering and Technology, 5*(4), 6333-6337.

Sharma, R. (2016). Cubic trigonometric Bézier curve with shape parameter. *International Journal of Innovative Research in Computer and Communication Engineering*, *4*(4), 7718-7723.

Sharma, R. (2019). Applications of quartic trigonometric Bézier curves and surfaces in CAGD. *International Journal of Research and Analytical Reviews, 6*(2), 462-475.

Sun, X., & Ji, X. (2020). Parametric model for kitchen product based on cubic T-Bezier curves with symmetry. *Symmetry*, *12*(4), 1-22.

Usman, M., Abbas, M., & Miura, K. T. (2020). Some engineering applications of new trigonometric cubic Bézier-like curves to free-form complex curve modeling. *Journal of Advanced Mechanical Design, Systems and Manufacturing*, *14*(4), 1–15.

Wang, K., & Zhang, G. (2018). New trigonometric basis possessing denominator shape parameters. *Mathematical Problems in Engineering*, *2018*, 1-25.

Wu, J., Zhang, X., & Peng, L. (2010). Positive approximation and interpolation using compactly supported radial basis functions. *Mathematical Problems in Engineering*, *2010,* 1-10.

Wu, X., Han, X., & Luo, S. (2007). Quadratic trigonometric spline curves with multiple shape parameters. *Proceedings of 2007 10th IEEE International Conference on Computer Aided Design and Computer Graphics, CAD/Graphics 2007*, 413–416.

Yahya, F. (2009). *Automatic G^1 Parametric Fitting of Curves and Surfaces to Outlines of Images*. Universiti Sains Malaysia.

Yahya, F., Ali, J.M., Majid, A.A., & Ibrahim, A. (2006). An automatic generation of G^1 curve fitting of Arabic characters. *IEEE International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)*, 542-547.

Yan, L. (2016). Cubic trigonometric nonuniform spline curves and surfaces. *Hindawi Publishing Corporation, Mathematical Problems in Engineering*, *2016*, 1-9.

Yang, H. M., Lu, J. J., & Lee, H. J. (2001). A Bezier curve-based approach to shape description for Chinese calligraphy characters. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR,* 276–280.

Yang, L., Li, J., & Chen, Z. (2011). Trigonometric extension of quartic Bézier curves. *2011 International Conference on Multimedia Technology, ICMT 2011*, 926–929.

Yang, L., Li, J., & Xie, C. (2012). A class of quasi-quartic trigonometric Bézier curves and surfaces. *Proceedings - 2012 IEEE Symposium on Electrical and Electronics Engineering, EEESYM 2012*, *4*, 121–124.

Zain, S. A. A. A. S. M., Misro, M. Y., & Miura, K. T. (2021). Generalized fractional Bézier curve with shape parameters. *Mathematics*, *9*(17), 1–32.

Zainudin, M. L., Yahya, Z. R., Muhamad, W. Z. A. W., & Hasan, Z. A. (2021). Curve reconstruction by cubic ball on Arabic fonts using butterfly optimization

algorithm. *Journal of Theoretical and Applied Information Technology*, *99*(23), 5700–5706.

Zhang, J., Lin, H., & Yu, J. (2007). A novel method for vectorizing historical documents of Chinese calligraphy. *Proceedings of 2007 10th IEEE International Conference on Computer Aided Design and Computer Graphics, CAD/Graphics 2007*, 219–224.

Zhu, Y., & Han, X. (2015). New trigonometric basis possessing exponential shape parameters. *Journal of Computational Mathematics*, *33*(6), 642–684.

Zhu, Y., Jiang, J., Tang, Z., Wang, F., Wang, R., Zhong, S., & Luo, X. (2018). An efficient geometric method for 3D reconstruction based on images. *International Journal of Grid and High-Performance Computing*, *10*(2), 34–46.

APPENDICES

APPENDIX 1

Cylinder

```
clc
clear all
close all
%control point for first surface
P0 = [1 0 0]; P1 = [1 2 0]; P2 = [2 2 0]; P3 = [2 0 0];
P4 = [1 0 1]; P5 = [1 2 1]; P6 = [2 2 1]; P7 = [2 0 1];
P8 = [1 0 2]; P9 = [1 2 2]; P10 = [2 2 2]; P11 = [2 0 2];
P12 = [1 0 3]; P13 = [1 2 3]; P14 = [2 2 3]; P15 = [2 0 3];
m=1;
hold on;

Q1=(2*P3)-P2
Q5=(2*P7)-P6
Q9=(2*P11)-P10
Q13=(2*P15)-P14

Q2=P1-((4.*m)+4)*P2+((4.*m)+4)*P3
Q6=P5-((4.*m)+4)*P6+((4.*m)+4)*P7
Q10=P9-((4.*m)+4)*P10+((4.*m)+4)*P11
Q14=P13-((4.*m)+4)*P14+((4.*m)+4)*P15

%control point for second surface
P3;Q1; Q2; Q3 = [1 0 0];
P7;Q5; Q6; Q7 = [1 0 1];
P11;Q9; Q10; Q11 = [1 0 2];
P15;Q13; Q14; Q15 = [1 0 3];

P1=(2*Q3)-Q2
P5=(2*Q7)-Q6
P9=(2*Q11)-Q10
P13=(2*Q15)-Q14

P2=Q1-((4.*m)+4)*Q2+((4.*m)+4)*Q3
```

```
P6=Q5-((4.*m)+4)*Q6+((4.*m)+4)*Q7
P10=Q9-((4.*m)+4)*Q10+((4.*m)+4)*Q11
P14=Q13-((4.*m)+4)*Q14+((4.*m)+4)*Q15

Gx=[P0(1) P1(1) P2(1) P3(1);P4(1) P5(1) P6(1) P7(1);P8(1) P9(1) P10(1) P11(1);P12(1)
      P13(1) P14(1) P15(1)];
Gy=[P0(2) P1(2) P2(2) P3(2);P4(2) P5(2) P6(2) P7(2);P8(2) P9(2) P10(2) P11(2);P12(2)
      P13(2) P14(2) P15(2)];
Gz=[P0(3) P1(3) P2(3) P3(3);P4(3) P5(3) P6(3) P7(3);P8(3) P9(3) P10(3) P11(3);P12(3)
      P13(3) P14(3) P15(3)];
Gxx=transpose(Gx);
Gyy=transpose(Gy);
Gzz=transpose(Gz);
Gx1=[P3(1) Q1(1) Q2(1) Q3(1);P7(1) Q5(1) Q6(1) Q7(1);P11(1) Q9(1) Q10(1)
      Q11(1);P15(1) Q13(1) Q14(1) Q15(1)];
Gy1=[P3(2) Q1(2) Q2(2) Q3(2);P7(2) Q5(2) Q6(2) Q7(2);P11(2) Q9(2) Q10(2)
      Q11(2);P15(2) Q13(2) Q14(2) Q15(2)];
Gz1=[P3(3) Q1(3) Q2(3) Q3(3);P7(3) Q5(3) Q6(3) Q7(3);P11(3) Q9(3) Q10(3)
      Q11(3);P15(3) Q13(3) Q14(3) Q15(3)];
Gx11=transpose(Gx1);
Gy11=transpose(Gy1);
Gz11=transpose(Gz1);

N=20;
t=(1:N)'/N;
p=(pi/2)*t;
T=[p.^0 sin(p) (sin(p).^2) (sin(p).^3) cos(p) (cos(p).^2) (cos(p).^3)];

A=[1 0 0 1;
   -(m+2) (2+m) 0 0;
   (1+2*m) -(2*m)-2 0 0;
   -m m 0 0;
   0 0 2+m -(m+2);
   0 0 -(2*m)-2 ((2*m)+1);
   0 0 m -m];

TT=transpose(T);
```

```
AA=transpose(A);
z1 = T*A*Gx*AA*TT;
z2 = T*A*Gy*AA*TT;
z3 = T*A*Gz*AA*TT;
z4 = T*A*Gx1*AA*TT;
z5 = T*A*Gy1*AA*TT;
z6 = T*A*Gz1*AA*TT;

X = [z1 z4];
Y = [z2 z5];
Z = [z3 z6];
X1 =[z4 z1];
Y1 =[z5 z2];
Z1 =[z6 z3];

surf(X,Y,Z)
hold on;
surf(X1,Y1,Z1)
hold on;
axis off
plot3(Gx,Gy,Gz,'o-r')
hold on;
plot3(Gx1,Gy1,Gz1,'o-r')
hold on;
plot3(Gxx,Gyy,Gzz,'o-r')
hold on;
plot3(Gx11,Gy11,Gz11,'o-r')
hold on;
view([-15,-5,15])
```

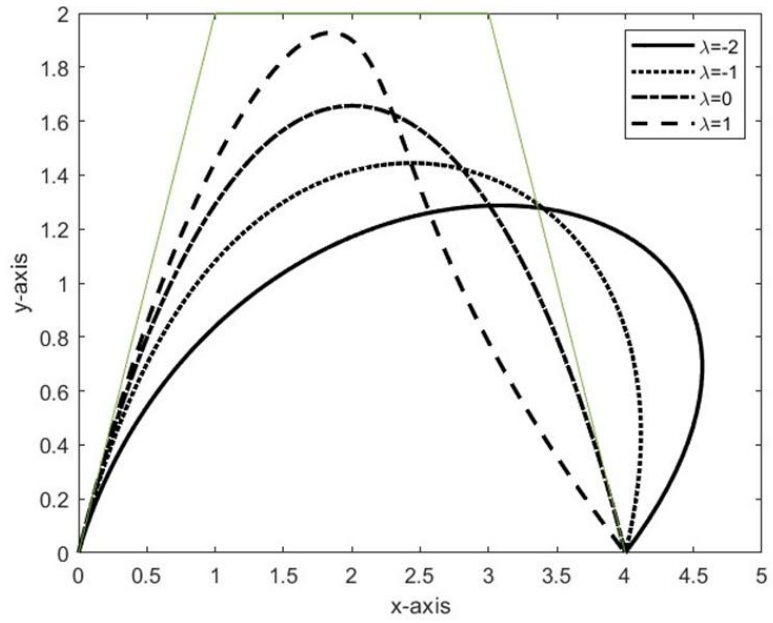
APPENDIX 2

Trigonometric Curve Han et. Al (2010)

```
clc
clear all
close all
%Plot curve
p0=[0 0];p1=[1 2];p2=[3 2];p3=[4 0];
x=[p0(1) p1(1) p2(1) p3(1)];
y=[p0(2) p1(2) p2(2) p3(2)];
m=-2;
t=0:0.01:1;
b00=((1-sin((pi/2)*t)).^2).*(1-(m.*sin((pi/2)*t)));
b10= sin((pi/2)*t).*(1-sin((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b20= cos((pi/2)*t).*(1-cos((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b30=((1-cos((pi/2)*t)).^2).*(1-(m.*cos((pi/2)*t)));
xDp=p0(1).*b00(1,:)+p1(1).*b10(1,:)+p2(1).*b20(1,:)+p3(1).*b30(1,:);
yDp=p0(2).*b00(1,:)+p1(2).*b10(1,:)+p2(2).*b20(1,:)+p3(2).*b30(1,:);
j0=plot(xDp,yDp,'k','linewidth',2);
hold on
dp=[xDp',yDp'];
m=-1;
t=0:0.01:1;

b00=((1-sin((pi/2)*t)).^2).*(1-(m.*sin((pi/2)*t)));
b10= sin((pi/2)*t).*(1-sin((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b20= cos((pi/2)*t).*(1-cos((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
```

```
b30=((1-cos((pi/2)*t)).^2).*(1-(m.*cos((pi/2)*t)));

xDp=p0(1).*b00(1,:)+p1(1).*b10(1,:)+p2(1).*b20(1,:)+p3(1).*b30(1,:);
yDp=p0(2).*b00(1,:)+p1(2).*b10(1,:)+p2(2).*b20(1,:)+p3(2).*b30(1,:);
j1=plot(xDp,yDp,'k:','linewidth',2);
hold on
dp=[xDp',yDp'];
m=0;
t=0:0.01:1;
b00=((1-sin((pi/2)*t)).^2).*(1-(m.*sin((pi/2)*t)));
b10= sin((pi/2)*t).*(1-sin((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b20= cos((pi/2)*t).*(1-cos((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b30=((1-cos((pi/2)*t)).^2).*(1-(m.*cos((pi/2)*t)));

xDp=p0(1).*b00(1,:)+p1(1).*b10(1,:)+p2(1).*b20(1,:)+p3(1).*b30(1,:);
yDp=p0(2).*b00(1,:)+p1(2).*b10(1,:)+p2(2).*b20(1,:)+p3(2).*b30(1,:);
j2=plot(xDp,yDp,'k-.','linewidth',2);
hold on
dp=[xDp',yDp'];
m=1;
t=0:0.01:1;
b00=((1-sin((pi/2)*t)).^2).*(1-(m.*sin((pi/2)*t)));
b10= sin((pi/2)*t).*(1-sin((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b20= cos((pi/2)*t).*(1-cos((pi/2)*t)).*(2+m-(m.*(sin((pi/2)*t))));
b30=((1-cos((pi/2)*t)).^2).*(1-(m.*cos((pi/2)*t)));

xDp=p0(1).*b00(1,:)+p1(1).*b10(1,:)+p2(1).*b20(1,:)+p3(1).*b30(1,:);
```

```
yDp=p0(2).*b00(1,:)+p1(2).*b10(1,:)+p2(2).*b20(1,:)+p3(2).*b30(1,:);

j3=plot(xDp,yDp,'k--','linewidth',2);

hold on

dp=[xDp',yDp'];

%plot control polygon

plot(x,y,'linewidth',0.5);

hold off

xlabel('x-axis');

ylabel('y-axis');

legend([j0,j1,j2,j3],'\lambda=-2','\lambda=-
1','\lambda=0','\lambda=1','Location','northeast')
```



AUTHOR'S PROFILE

Nur Azliana Azlin binti Ahmad Munir completed her PhD in Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA. She received her MSc. in Science Mathematics at School of Mathematical Sciences, University of Science, Malaysia.

LIST OF PUBLICATIONS

Munir, N. A. A., Hadi, N. A., and Nasir, M. A. S. (2022). Modelling fruit outline using cubic trigonometric spline. *AIP Conference Proceedings*, 2905: 030006.

Munir, N. A. A., Hadi, N. A., and Nasir, M. A. S. (2021). C^1 Cubic trigonometric spline with a shape parameter for positive shape preservation. *Malaysian Journal of Mathematical Sciences*, (2022) : 55-66

Munir, N. A. A., Hadi, N. A., and Nasir, M. A. S. (2020). Constrained for G^1 cubic trigonometric spline curve interpolation for positive data set. *Malaysian Journal of Fundamental and Applied Sciences*, 18(2022): 323-331.

Munir, N. A. A. A., Yahya, F., and Hadi, N. A. (2019). Cubic trigonometric spline for preserving positive data. *ASM Science Journal*, 12(1): 67-73.

Munir, N. A. A. A., Yahya, F., and Hadi, N. A. (2018). G^1 quadratic trigonometric beta spline with a shape parameter. *AIP Conference Proceedings*, 1974(1):030028.