Available online at https://ganitjournal.bdmathsociety.org/

GANIT: Journal of Bangladesh Mathematical Society

GANIT *J. Bangladesh Math. Soc.* 45.1 (2025) 050–063
DOI: https://doi.org/10.3329/ganit.v45i1.82275

Study of Subdivision Curve Schemes and Their Impact on Computer-Aided Geometric Design and Computer Graphics

Md. Nur Alam*

*Professor, Department of Mathematics, Pabna University of Science & Technology, Pabna-6600, Bangladesh.

ABSTRACT

Computer-Aided Geometric Design (CAGD) is a branch of applied mathematics that focuses on the computational modeling and representation of geometric shapes. It plays a crucial role in diverse fields such as geographic information systems, computer gaming, medical imaging, robotics, engineering, and traditional industries like automobile, aircraft, and ship design. A core challenge in CAGD is the creation of smooth curves and surfaces through efficient mathematical techniques. In recent years, subdivision schemes have emerged as a practical and elegant method for generating smooth limit curves. These techniques are widely used in computer animation (CA), CAGD, and computer graphics (CG) due to their simplicity, flexibility, and effectiveness. This study presents three prominent subdivision schemes which as the Chaikin Subdivision (CS), Corner-Cutting Subdivision (CSS), and Four-Point Subdivision (FPS) schemes. Each operates by refining an initial control polygon through iterative rules that add new points as weighted combinations of existing ones. Repeated application of these rules produces a limit curve with increasing smoothness. We begin by constructing various initial shapes—such as a jar, a mango, pi, and a car—and applying the CS scheme at multiple subdivision levels. Next, we apply the CSS scheme to shapes like a rabbit, a five-fingered hand, and a pi. Finally, the FPS scheme is applied to "U", mug, and pi shapes. The results convincingly demonstrate the capability of these schemes to produce smooth and visually appealing curves with high precision. Looking ahead, our future research aims to develop and investigate non-uniform variants of these schemes. This direction seeks to improve adaptability for handling complex and irregular geometries, while maintaining the core properties of convergence, smoothness, and visual fidelity—ultimately expanding their utility in advanced geometric modeling.

Received: February 27, 2025 **Accepted:** June 13, 2025 **Published Online:** June 30, 2025

Keywords: Subdivision; The Chaikin subdivision scheme; Corner cutting subdivision scheme; Four-point subdivision scheme; Geometrical modeling.

AMS Subject Classifications 2025: 65D17, 65D07, 65D05.

1. Introduction

Subdivision is a powerful algorithm used to generate curves and surfaces through the iterative refinement of control polygons and control meshes. The set of rules governing this refinement process is collectively known as a subdivision scheme [1, 2, 3, 4, 5]. These techniques are capable of producing highly smooth and accurate geometric representations by repeatedly applying simple topological or geometric operations. Beginning with an initial control polygon, subdivision methods introduce new control points at each iteration, gradually converging toward a smooth limit curve or surface. Subdivision schemes are widely used in CA, CAGD, CG, and signal analysis, due to their simplicity, efficiency, and visual fidelity. In recent years, the study and application of subdivision curves have become a significant research focus within CAGD and CG [6, 7, 8]. Early research primarily explored the relationships between control points and the resulting geometric shapes. Subdivision schemes can be broadly categorized into two types: interpolating and

*Corresponding author. E-mail address: nuralam.pstu23@gmail.com; nuralam23@pust.ac.bd

approximating [9, 10]. If the resulting limit curve passes through the original control points, the scheme is classified as interpolating; if not, it is termed approximating. This work aims to construct high-quality limit curves that satisfy various geometric constraints, providing insight into the current state of subdivision research and its practical applications.
In this paper, we apply three prominent subdivision schemes, which are CS [11, 12, 13], CSS [14, 15], and FPS [16, 17, 18] schemes, to a variety of initial models. We begin by applying the CS scheme to shapes such as a jar, a mango, a pi, and a car. Then, the CSS scheme is applied to shapes like a rabbit, a five-fingered hand, and a pi. Finally, the FPS scheme is applied to "U", mug, and pi shapes. The results illustrate the effectiveness of these schemes in producing smooth, visually pleasing curves suitable for advanced geometric modeling tasks. In summary, the main objectives of this research are as follows:

(a) To study three subdivision curve schemes, which are the CS scheme, the CSS scheme, and the FPS scheme.
(b) To construct the subdivision matrices for these schemes, with particular emphasis on their stationarity, meaning all refinement matrices remain consistent throughout the iterative process.
(c) To perform numerical experiments demonstrating that the resulting limit curves exhibit superior geometric quality.
(d) To emphasize the importance of the high-quality, smooth limit curves and continuity properties achieved by these schemes, which are vital for applications in CG and CAGD.
(e) To analyze the continuity and convergence properties of the schemes by examining the eigenvalues and eigenvectors of the local subdivision matrices.

2. Problem statements

This section provides a brief overview of the CS scheme, CSS scheme, and FPS scheme. These three techniques are widely used in computer graphics to generate smooth curves from an initial set of points or a polyline. The algorithms work by iteratively inserting new points along the line, progressively refining it to produce a smoother curve according to the following steps:

(a) **Input**: Polygon or Polyhedron meshes (Figure 2a).
(b) **Process**: Repeatedly refine (subdivide) Geometry (Figure 2b).
(c) **Output**: Smooth curve or surface (Figure 2c).

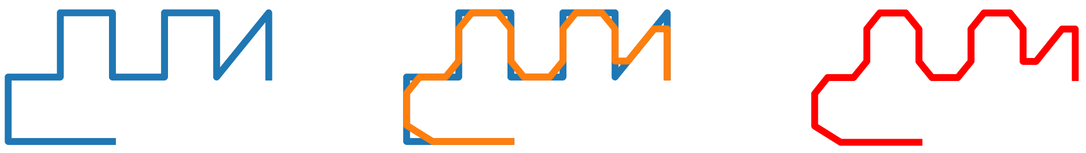

(a) Input (b) Process (c) Output

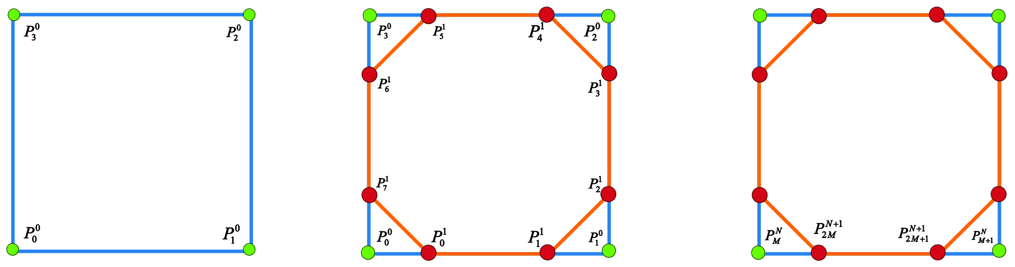
Figure 2: The rules for subdivision

3. Chaikin's subdivision scheme

This section provides a comprehensive examination of the CS scheme, focusing on its refinement rules, continuity analysis, and numerical simulations. The continuity of the scheme is rigorously analyzed using eigenvalue analysis of the associated local subdivision matrix. This analytical approach provides a solid theoretical foundation for evaluating the smoothness properties of the scheme, a crucial aspect for producing high-quality visual results in computer graphics and animation.

3.1. The topological and geometric rules for the CS scheme

The topological rule for the Chaikin curve subdivision scheme can be considered as an iterative procedure that begins with a set of input control points, as illustrated in Figure 3.1(a). At each iteration, the scheme upsamples the curve by inserting new control points between every pair of adjacent existing points. This process effectively doubles the number of control points—transforming an initial set of n control points into 2n, as shown in Figure 3.1(b).



(a) Initial control points (b) Topology rule (c) Geometric rule

Figure 3.1: The rules for the Chaikin subdivision curve scheme

The geometric rules of the Chaikin curve subdivision scheme can be formalized in a similar way. For level N with k control points P_M^N, after subdivision, a $2k$ new control points P_{2M}^{N+1} and $P_{2M+1}^{N+1}, M = 0,1,2,\cdots,k-1$ are computed as follows which is shown in Figure 3.1(c):

$$P_{2M}^{N+1} = \frac{3}{4}P_M^N + \frac{1}{4}P_{M+1}^N;\; P_{2M+1}^{N+1} = \frac{1}{4}P_M^N + \frac{3}{4}P_{M+1}^N. \tag{3.1}$$

Equation (3.1) can be written as the matrix representation is given below.

$$\begin{bmatrix} P_{2M}^{N+1} \\ P_{2M+1}^{N+1} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix} \begin{bmatrix} P_M^N \\ P_{M+1}^N \end{bmatrix}. \tag{3.2}$$

When we set the value of For $M = 0,1,2,\cdots,k-1$, then equation (3.2) becomes

$$\begin{bmatrix} P_0^1 \\ P_1^1 \\ P_2^1 \\ P_3^1 \\ \vdots \\ P_{k-1}^1 \\ P_k^1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 3 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 3 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 3 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 3 & 0 & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 3 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 3 \end{bmatrix} \begin{bmatrix} P_0^0 \\ P_1^0 \\ P_2^0 \\ P_3^0 \\ \vdots \\ P_{k-1}^0 \\ P_k^0 \end{bmatrix} \tag{3.3}$$

The Chaikin curve subdivision scheme is a simple and effective method for generating smooth curves. It is commonly used for line simplification and curve generation. To perform a mathematical proof of the continuity analysis for the Chaikin curve subdivision scheme, we will focus on C^0 and C^1 continuity [13]. For C^0 continuity, we need to ensure that the curve is continuous at the junction points. By applying the subdivision scheme and comparing the newly computed points with the original points at the junction, we can verify that they match. This demonstrates that the curve remains continuous at these points [13]. For C^1 continuity, we need to examine the derivatives at the junction points. By setting $P_M^N = P_{M+1}^N$ and $\frac{d}{dt}(P_M^N) = \frac{d}{dt}(P_{M+1}^N)$ for the original curve, we can verify that the derivatives are the same at the junction points, ensuring C^1 continuity [13].

3.2. *Continuity analysis for the CS scheme*

Substituting M=0 in equation (3.1), we have

$$P_0^{N+1} = \frac{3}{4}P_0^N + \frac{1}{4}P_1^N;\; P_1^{N+1} = \frac{1}{4}P_0^N + \frac{3}{4}P_1^N; \tag{3.4}$$

Now, we convert equation (3.4) into matrix form, and we obtain

$$\begin{pmatrix} P_0^{N+1} \\ P_1^{N+1} \end{pmatrix} = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix} \begin{pmatrix} P_0^{N} \\ P_1^{N} \end{pmatrix}.$$

This implies $P^{N+1} = SP^N$, where $P^{N+1} = \{P_M{}^{N+1}\}_{M=0}^{1}$ and $P^N = \{P_M{}^N\}_{M=0}^{1}$ are column matrices and S is the local subdivision matrix is defined below are $S = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix}$ has an invariant neighborhood size of two. Eigenvalues λ_i, where i=1, 2 of the matrix S are $\lambda_1 = 1$ and $\lambda_2 = \frac{1}{2}$ Eigenvectors v_i, where i=1, 2, corresponding to these eigenvalues are $v_1 = (1,1)^T$ and $v_2 = (-1,1)^T$, respectively. Therefore, we can define diagonal matrix Ω and non-singular matrix Q as $\Omega = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$ and $Q = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$. By diagonalization of matrix S, we get $S = Q\Omega Q^{-1}$, where $Q^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$. This can be proved by induction on N. Since Ω is diagonal matrix and for any diagonal matrix Ω^2 means square of diagonal entries and so on. Therefore, $\Omega^N = \begin{pmatrix} (1)^N & 0 \\ 0 & (\frac{1}{2})^N \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2^N} \end{pmatrix}$. This implies that $\lim_{N\to\infty} \Omega^N = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$. Since $P^{N+1} = SP^N = S(SP^{N-1}) = S^2 P^{N-1} = \ldots = S^N P^0$, then $P^{N+1} = (Q\Omega Q^{-1})P^0$. Taking limit, we get $P^{\infty} = Q(\lim_{N\to\infty} \Omega^N)Q^{-1})P^0$. This implies $\begin{pmatrix} P_0^{\infty} \\ P_1^{\infty} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} P_0^{0} \\ P_1^{0} \end{pmatrix}$. Thus, the limit stencil is $[\frac{1}{2}, \frac{1}{2}]$. This limit stencil can be used to find the limiting position of the control point $P_0{}^0$ on the limit curve. Key observations for the CS scheme include:

- The CS scheme is convergent because one of the eigenvalues is 1, and the other eigenvalues are less than 1.
- The first row of Q^{-1} is called a limit stencil, which is $[\frac{1}{2}, \frac{1}{2}]$.
- If we apply a limit stencil on two consecutive points, we obtain a point on the limit curve.
- We work at any level, and can snap the point to limit the curve can be used to estimate the distance to the limit curve
- In this CS scheme, even/odd stencils are symmetric to each other.

3.3. Numerical experiments for the CS scheme

In this section, several numerical experiments are performed to demonstrate the effectiveness and quality of the shapes generated using the Chaikin curve subdivision scheme. High-quality, smooth subdivision limit curves are illustrated with the aid of Python programming. To evaluate the quality of the resulting curves, we apply the Chaikin subdivision scheme to various real-world polygonal models. The experiments are conducted on four distinct shapes: a Jar, a Mongo, a Pi, and a Car. The Chaikin subdivision scheme is applied iteratively to the jar-shaped polygon. Subdivision results after the 1st, 2nd, 10th, and 20th iterations are shown in Figure 3.2. The 20th subdivision clearly yields a smooth and high-quality limit curve. Similarly, the Mongo-shaped polygon undergoes Chaikin subdivision. The results of the 1st, 2nd, 10th, and 20th subdivisions are illustrated in Figure 3.3, with the final (20th) subdivision producing a visually smooth and refined curve. The same procedure is followed for the Pi-shaped polygon. As shown in Figure 3.4, progressive subdivision leads to increasingly smooth curves, culminating in a high-quality result at the 20th iteration. Lastly, the Car-shaped polygon is subdivided using the same technique. Figure 3.5 displays the results after the 1st, 2nd, 10th, and 20th subdivisions. The final curve after 20 iterations exhibits a highly smooth and visually appealing shape. These experimental results confirm that the Chaikin curve subdivision scheme effectively produces smooth and aesthetically pleasing curves from a variety of polygonal shapes.

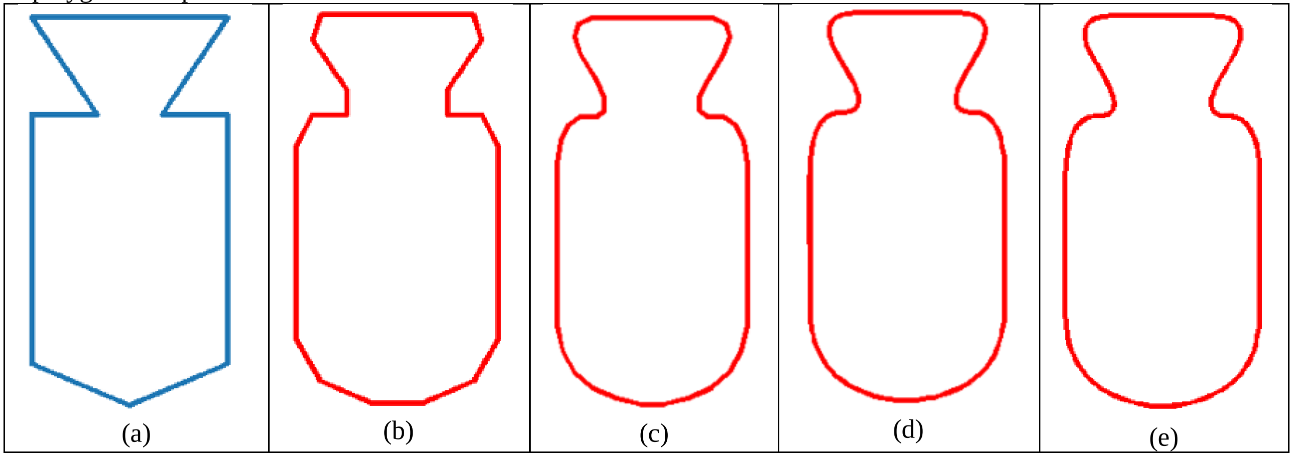


Figure 3.2: Refinement of a Jar-shaped polygon using the Chaikin subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

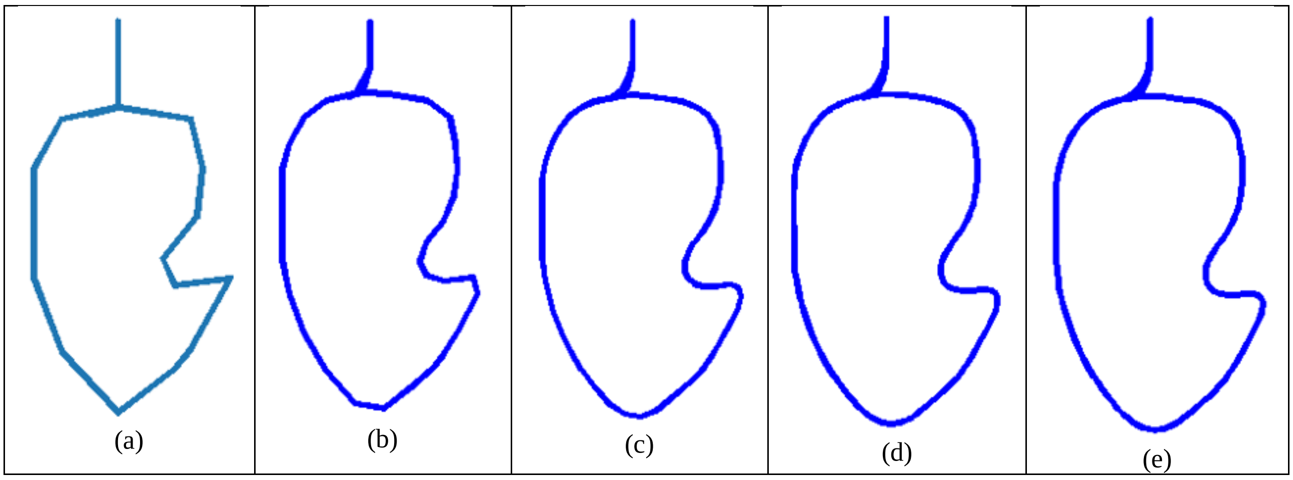


Figure 3.3: Refinement of a Mango-shaped polygon using the Chaikin subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

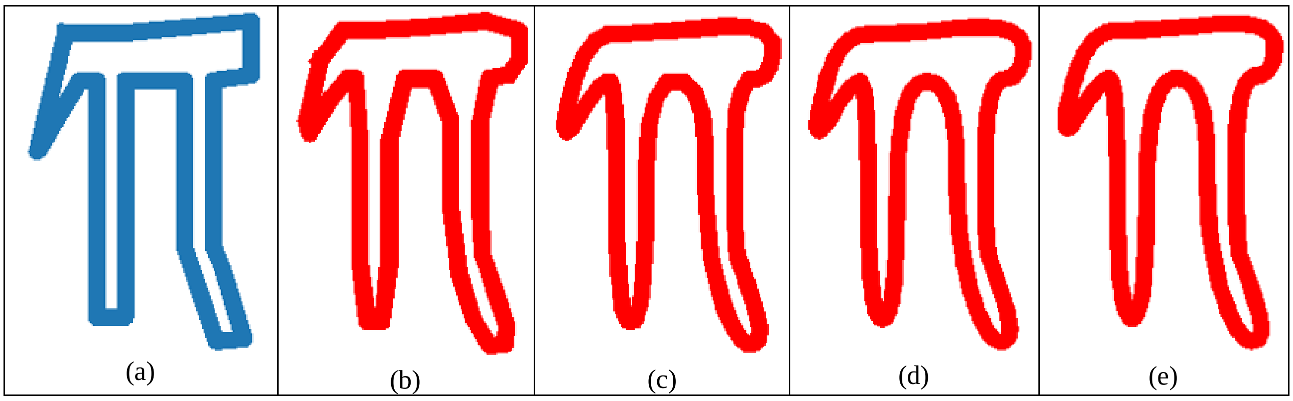


Figure 3.4: Refinement of a Pi-shaped polygon using the Chaikin subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

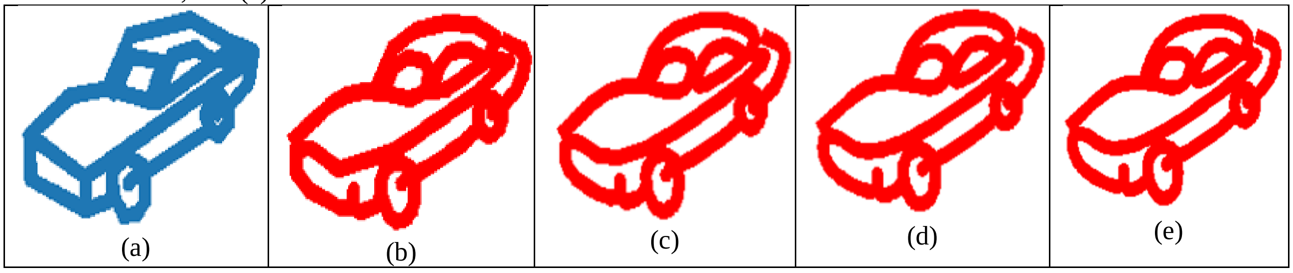


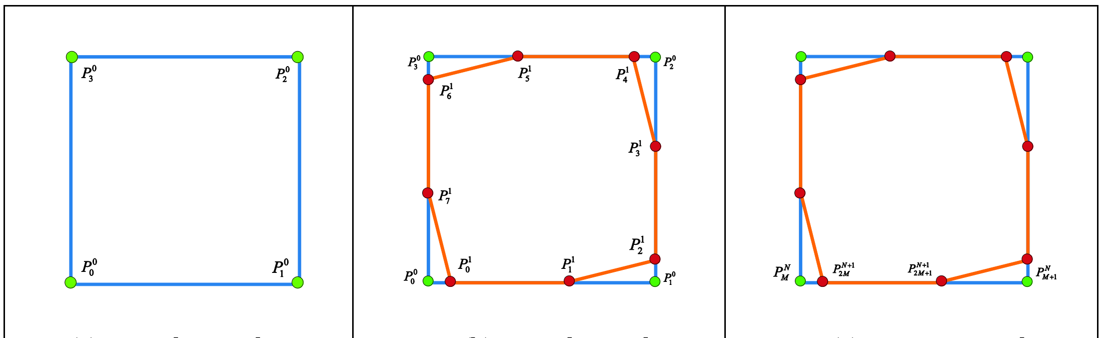
Figure 3.5: Refinement of a Car-shaped polygon using the Chaikin subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

4. Corner-Cutting subdivision scheme

This section comprehensively examines the CCS scheme, covering its refinement rules, continuity evaluation, and numerical simulations. The scheme's continuity is thoroughly analyzed through eigenvalue analysis of the associated local subdivision matrix. This method offers a rigorous theoretical basis for understanding the smoothness characteristics of the scheme, which are essential for producing high-quality visual outcomes in computer graphics and animation.

4.1. The topological and geometric rules for the CSS scheme

The topology and geometric rule of the corner cutting subdivision curve scheme is exactly as the Chaikin curve subdivision scheme. The corner-cutting subdivision curve scheme is the generalized form of the Chaikin curve subdivision scheme with the parameters $0 < R < S < 1$. If you set $R = \frac{3}{4}$ and $S = \frac{1}{4}$ of the equation (4.1), then the corner cutting subdivision curve scheme is the same as the Chaikin subdivision curve scheme. If $R \neq \frac{3}{4}$ and $S \neq \frac{1}{4}$ of the equation (4.1), then it is called the corner-cutting subdivision curve scheme. The topological rule for the corner cutting subdivision scheme can be described as an iterative process that starts with a given set of input control points, as shown in Figure 4.1(a). In each iteration, new control points are inserted between every pair of adjacent existing points, effectively upsampling the curve. This results in the doubling of control points from n to 2n as depicted in Figure 4.1(b).



(a) Initial control points (b) Topology rule (c) Geometric rule

Figure 4.1: The rules for Corner-cutting subdivision curve scheme

The geometric rules of the Corner-cutting subdivision curve scheme can be formalized in the similar way. For level N with k control points P_M^N, , after subdivision, a $2k$ new control points P_{2M}^{N+1} and $P_{2M+1}^{N+1}, M = 0,1,2,\cdots,k-1$ are computed is shown in the Figure 4.1(c), which as the following:

$$P_{2M}^{N+1} = (1-R)P_M^N + RP_{M+1}^N;\; P_{2M+1}^{N+1} = (1-S)P_M^N + SP_{M+1}^N,\; 0<R<S<1. \tag{4.1}$$

Equation (4.1) can be written as the matrix representation is given below.

$$\begin{bmatrix} P_{2M}^{N+1} \\ P_{2M+1}^{N+1} \end{bmatrix} = \begin{bmatrix} 1-R & R \\ 1-S & S \end{bmatrix} \begin{bmatrix} P_M^N \\ P_{M+1}^N \end{bmatrix}. \tag{4.2}$$

When we set the value of For $M = 0,1,2,\cdots,k-1$, then equation (4.2) becomes

$$\begin{bmatrix} P_0^1 \\ P_1^1 \\ P_2^1 \\ P_3^1 \\ \vdots \\ P_{k-1}^1 \\ P_k^1 \end{bmatrix} = \begin{bmatrix} 1-R & S & 0 & 0 & 0 & \cdots & 0 \\ 1-S & R & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1-R & R & 0 & 0 & \cdots & 0 \\ 0 & 1-S & S & 0 & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 1-R & R \\ 0 & \cdots & 0 & 0 & 0 & 1-S & S \end{bmatrix} \begin{bmatrix} P_0^0 \\ P_1^0 \\ P_2^0 \\ P_3^0 \\ \vdots \\ P_{k-1}^0 \\ P_k^0 \end{bmatrix} \tag{4.3}$$

The Corner cutting curve subdivision scheme, also known as the corner cutting algorithm, is used for generating smoother curves from an initial set of control points. To conduct a mathematical proof of the continuity analysis for the Corner cutting curve subdivision scheme, we will examine C^0 and C^1 continuity [15]. For C^0 continuity, we need to demonstrate that the curve remains continuous at the junction points. By applying the subdivision scheme and examining the newly computed points at the junction, we can verify that they match. This ensures that the curve remains continuous at these points. For C^1 continuity, we need to examine the derivatives at the junction points. Taking the derivative of the Corner cutting subdivision equations, we have: $\frac{d}{dt}(P_{2M}^{N+1}) = P_{M+1}^N - P_M^N$. This derivative is constant and does not depend on t. Therefore, the derivative is the same for the junction points, ensuring C^1 continuity [15]. This analysis demonstrates the C^0 and C^1 continuity for the Corner cutting curve subdivision scheme. The simplicity of the algorithm ensures that the resulting curves remain continuous and smooth across the subdivision iterations, making it an effective method for generating smoother approximations of complex curves.

4.2. Continuity analysis for the CCS scheme

Substituting M=0 in equation (3.2), we have

$$P_0^{N+1} = (1-R)P_0^N + RP_1^N;\; P_1^{N+1} = (1-S)P_0^N + SP_1^N; \tag{4.4}$$

Now, we convert equation (4.4) into matrix form, and we obtain

$$\begin{pmatrix} P_0^{N+1} \\ P_1^{N+1} \end{pmatrix} = \begin{pmatrix} 1-R & R \\ 1-S & S \end{pmatrix} \begin{pmatrix} P_0^N \\ P_1^N \end{pmatrix}.$$

This implies $P^{N+1} = SP^N$, where $P^{N+1} = \{P_M{}^{N+1}\}_{M=0}^1$ and $P^N = \{P_M{}^N\}_{M=0}^1$ are column matrices and S is the local subdivision matrix is defined below are $S=\begin{pmatrix}1-R & R\\ 1-S & S\end{pmatrix}$ has invariant neighborhood size of two. Eigenvalues λ_i, where i=1, 2 of the matrix S are $\lambda_1=1$ and $\lambda_2=S-R$. Eigenvectors v_i, where i=1, 2, corresponding to these eigenvalues are $v_1=(1,1)^T$ and $v_2=(\frac{R}{S-1},1)^T$, respectively. Therefore, we can define diagonal matrix Ω and non-singular matrix Q as $\Omega=\begin{pmatrix}1 & 0\\ 0 & S-R\end{pmatrix}$ and $Q=\begin{pmatrix}1 & \frac{R}{S-1}\\ 1 & 1\end{pmatrix}$. By diagonalization of matrix S, we get $S=Q\Omega Q^{-1}$, where $Q^{-1}=\begin{pmatrix}\frac{1-S}{1-S+R} & \frac{R}{1-S+R}\\ \frac{S-1}{1-S+R} & \frac{1-S}{1-S+R}\end{pmatrix}$. This can be proved by induction on N. Since Ω is diagonal matrix and for any diagonal matrix Ω^2 means square of diagonal entries and so on. Therefore, $\Omega^N=\begin{pmatrix}(1)^N & 0\\ 0 & (S-R)^N\end{pmatrix}$, $0<R<S<1$. This implies that $\lim_{N\to\infty}\Omega^N=\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$. Since $P^{N+1}=SP^N=S(SP^{N-1})=S^2P^{N-1}=\cdots=S^NP^0$, then $P^{N+1}=(Q\Omega Q^{-1})P^0$. Taking limit, we get $P^\infty=Q(\lim_{N\to\infty}\Omega^N)Q^{-1})P^0$. This implies $\begin{pmatrix}P_0^\infty\\ P_1^\infty\end{pmatrix}=\begin{pmatrix}\frac{1-S}{1-S+R} & \frac{1-S}{1-S+R}\\ \frac{1-S}{1-S+R} & \frac{1-S}{1-S+R}\end{pmatrix}\begin{pmatrix}P_0^0\\ P_1^0\end{pmatrix}$. Thus the limit stencil is $\{\frac{1-S}{1-S+R},\frac{S-1}{1-S+R}\}$. This limit stencil can be used to find the limiting position of the control point P_0^0 on the limit curve.

4.3. *Numerical experiments for the CSS scheme*

In this section, we present a series of numerical experiments to evaluate the effectiveness of the Chaikin curve subdivision scheme and the corner-cutting subdivision curve scheme in generating high-quality shapes. The results are visualized through smooth subdivision limit curves produced using Python programming. To illustrate the quality of the generated curves, we apply the corner-cutting subdivision scheme to several real-world polygonal models. The corner-cutting scheme is applied to a polygon resembling a rabbit. The results after the 1st, 2nd, 10th, and 20th subdivisions are shown in Figure 4.2. As evident from the figure, the 20th subdivision yields a highly smooth and visually refined limit curve. The same subdivision process is applied to a five-finger-shaped polygon. Figure 4.3 displays the results at various stages of subdivision (1st, 2nd, 10th, and 20th iterations). The final result exhibits a smooth and high-quality curve after 20 subdivisions. Finally, the corner-cutting scheme is used on a Pi-shaped polygon. As illustrated in Figure 4.4, the 1st, 2nd, 10th, and 20th subdivisions progressively smooth the shape, with the 20th subdivision producing an excellent quality limit curve. These experimental results demonstrate the capability of the corner-cutting subdivision scheme in generating smooth and visually appealing curves from a variety of polygonal input shapes.

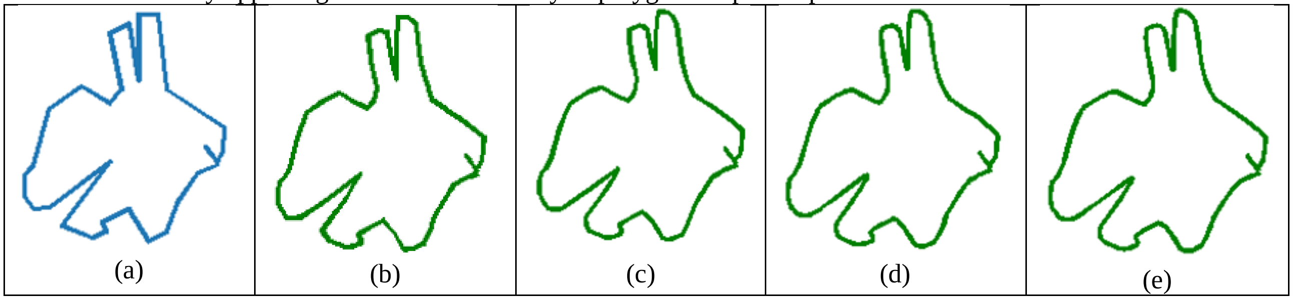


Figure 4.2: Refinement of a Rabbit -shaped polygon using the corner-cutting subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

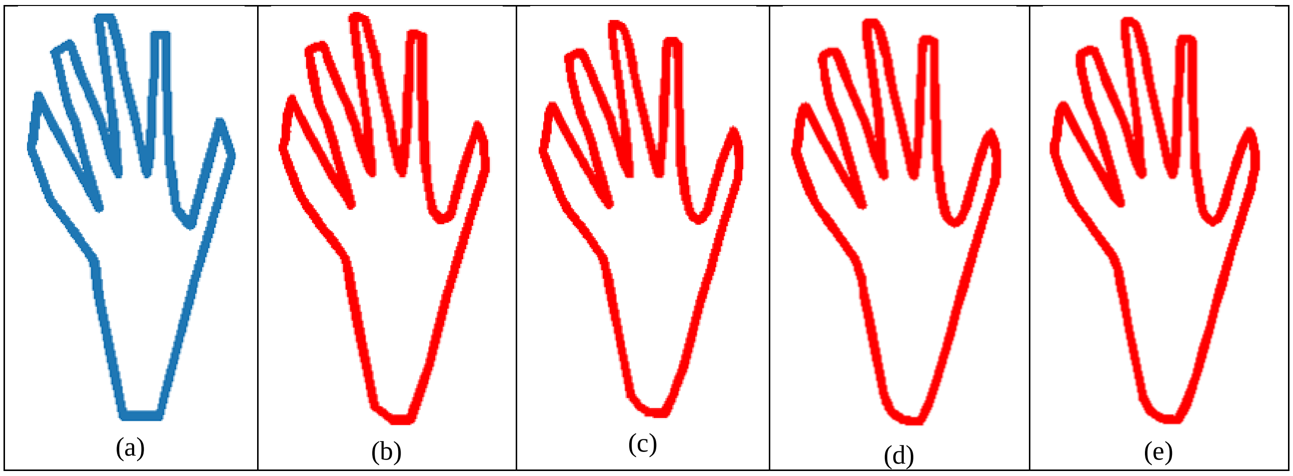


Figure 4.3: Refinement of a five finger-shaped polygon using the corner-cutting subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

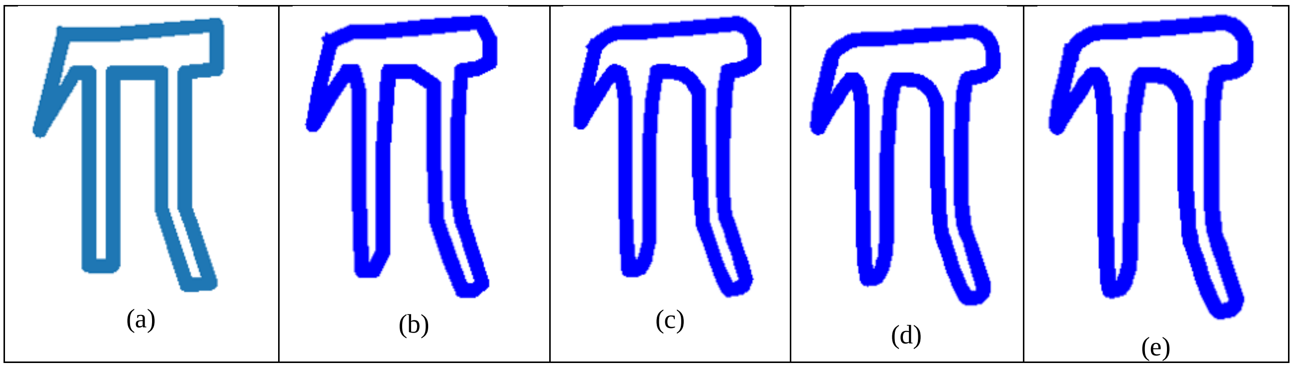


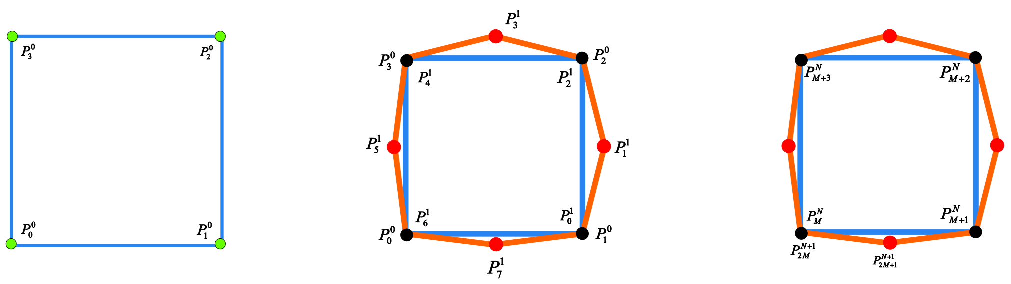
Figure 4.4: Refinement of a Pi-shaped polygon using the corner-cutting subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

5. FPS scheme

This section comprehensively examines the FPS scheme, covering its refinement rules, continuity evaluation, and numerical simulations. The scheme's continuity is thoroughly analyzed through eigenvalue analysis of the associated local subdivision matrix. This method offers a rigorous theoretical basis for understanding the smoothness characteristics of the scheme, which are essential for producing high-quality visual outcomes in computer graphics and animation.

5.1. The topological and geometric rules for the FPS scheme

The topological rule for the four-point subdivision scheme can be considered as an iterative process that begins with a set of input control points, as illustrated in Figure 5.1(a). At each iteration, the curve is upsampled by inserting a new control point between every pair of adjacent control points. This operation doubles the total number of control points from n to 2n as depicted in Figure 5.1(b).



(a) Initial control points (b) Topology rule (c) Geometric rule

Figure 5.1: The rules for the four-point subdivision curve scheme

The geometric rules of the four-point subdivision curve scheme can be formalized similarly. For level N with k control points P_M^N, $M = 0,1,2,\cdots,k-1$, after subdivision, a $2k$ new control points P_{2M}^{N+1} and $P_{2M+1}^{N+1}, M = 0,1,2,\cdots,k-1$ are computed as follows which is shown in Figure 5.1(c):

$$P_{2M}^{N+1} = P_M^N;\; P_{2M+1}^{N+1} = (W+0.5)(P_M^N + P_{M+1}^N) - W(P_{M-1}^N + P_{M+2}^N), \tag{5.1}$$

where W is the tension parameter. If we set $W = \dfrac{1}{16}$, then equation (5.1) becomes

$$P_{2M}^{N+1} = P_M^N; P_{2M+1}^{N+1} = \tfrac{9}{16}(P_M^N + P_{M+1}^N) - \tfrac{1}{16}(P_{M-1}^N + P_{M+2}^N). \tag{5.2}$$

Which represents the best continuity properties and positions P_{2M}^{N+1} on the Lagrange cubic through P_{M-1}^N, P_M^N, P_{M+1}^N and P_{M+2}^N. The subdivided polygon is then considered as a new control polygon and the same methods implement to it and much more repeatedly. Repeating this sequence of polygons that converges to a limit curve. It is understood that the uniform four point subdivision curve scheme makes a C^1 continuity limit curve for $0 < W < \dfrac{1}{8}$.

5.2. *Continuity analysis for the FPS scheme*

Substituting M=-1 in equation (5.2), we have

$$P_{-2}^{N+1} = P_{-1}^N;\; P_{-1}^{N+1} = -\frac{1}{16}P_{-2}^N + \frac{9}{16}P_{-1}^N + \frac{9}{16}P_0^N - \frac{1}{16}P_1^N. \tag{5.3}$$

Substituting M=0 in equation (5.2), we have

$$P_0^{N+1} = P_0^N;\; P_{-1}^{N+1} = -\frac{1}{16}P_{-1}^N + \frac{9}{16}P_0^N + \frac{9}{16}P_1^N - \frac{1}{16}P_2^N. \tag{5.4}$$

Substituting M=1 in equation (5.2), we have

$$P_2^{N+1} = P_1^N. \tag{5.5}$$

Now, we convert equations (5.3), (5.4), and (5.5) into matrix form, and we obtain

$$\begin{pmatrix} P_{-2}^{N+1} \\ P_{-1}^{N+1} \\ P_0^{N+1} \\ P_1^{N+1} \\ P_2^{N+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} P_{-2}^N \\ P_{-1}^N \\ P_0^N \\ P_1^N \\ P_2^N \end{pmatrix} = S \begin{pmatrix} P_{-2}^N \\ P_{-1}^N \\ P_0^N \\ P_1^N \\ P_2^N \end{pmatrix},$$

This implies $P^{N+1} = SP^N$, where $P^{N+1} = \{P_M^{N+1}\}_{M=2}^{-2}$ and $P^N = \{P_M^N\}_{M=2}^{-2}$ are column matrices and S is the local subdivision matrix is defined below are

$$S=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

has invariant neighborhood size of five. Eigenvalues λ_i, where i=1, 2, 3, 4, 5 of the matrix S are $\lambda_1=1$, $\lambda_2=\frac{1}{2}$, $\lambda_3=\frac{1}{4}$, $\lambda_4=\frac{1}{4}$ and $\lambda_5=\frac{1}{8}$. Eigenvectors corresponding to these eigenvalues are $v_1=(1,1,1,1,1)^T$, $v_2=(-1,-\frac{1}{2},0,\frac{1}{2},1)^T$, $v_3=(1,\frac{1}{4},0,\frac{1}{4},1)^T$, $v_4=(0,0,0,0,0)^T$ and $v_5=(-1,-\frac{1}{8},0,\frac{1}{8},1)^T$, respectively. Therefore, we can define diagonal matrix Ω and non-singular matrix Q as

$$\Omega=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{8} \end{pmatrix} \text{ and } Q=\begin{pmatrix} 1 & -1 & 1 & 0 & -1 \\ 1 & -\frac{1}{2} & \frac{1}{4} & 0 & -\frac{1}{8} \\ 1 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{8} \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Similarly, for the CS and CSS scheme discussed in Sections 3 and 4, diagonalization of the local subdivision matrix S, we get $S=Q\Omega Q^{-1}$ and $\lim_{N\to\infty}\Omega^N S^\infty=\lim_{N\to\infty}Q\Omega^N Q^{-1}=Q(\lim_{N\to\infty}\Omega^N)Q^{-1}$. After this, consider the subdivision system, we get $P^\infty=Q(\lim_{N\to\infty}\Omega^N)Q^{-1})P^0$, which is the complete procedure to find out the limit stencil that can be used to find the limiting position of the control point P_0^0 on the limit curve. This approach is valid for all interpolating schemes. We establish the C^1 continuity of the FPS scheme using eigenvalue analysis of its local subdivision matrix S. The matrix yields eigenvalues $\lambda_1=1$ and $\lambda_2=\frac{1}{2}$, while all remaining eigenvalues satisfy $|\lambda_i|<\frac{1}{2}$. This confirms that the scheme preserves he C^1 continuity and generates visually smooth curves.

5.3. Numerical experiments for the FPS scheme

In this section, a series of numerical experiments is conducted to demonstrate the effectiveness of the FPS scheme in generating high-quality curves. Smooth limit curves are obtained and visualized using Python programming, highlighting the scheme's ability to produce visually appealing and refined shapes. To evaluate the performance of the FPS scheme, it is applied to several real-world polygonal models. The FPS scheme is applied to a “U”-shaped polygon. The results after the 1st, 2nd, 10th, and 20th subdivisions are illustrated in Figure 5.2. As observed, the 20th subdivision produces a highly smooth and visually pleasing limit curve. Similarly, the Mug-shaped polygon is subjected to the FPS scheme. Figure 5.3 shows the curve evolution across the 1st, 2nd, 10th, and 20th subdivisions. The final result demonstrates a high-quality smooth curve after 20 iterations. Finally, the FPS scheme is applied to a Pi-shaped polygon. The outcomes of the 1st, 2nd, 10th, and 20th subdivisions are presented in Figure 5.4. The 20th subdivision yields a smooth and refined limit curve, confirming the effectiveness of the scheme. These experimental results validate the capability of the Four-Point

Subdivision scheme in producing smooth, high-quality curves from various polygonal input shapes.

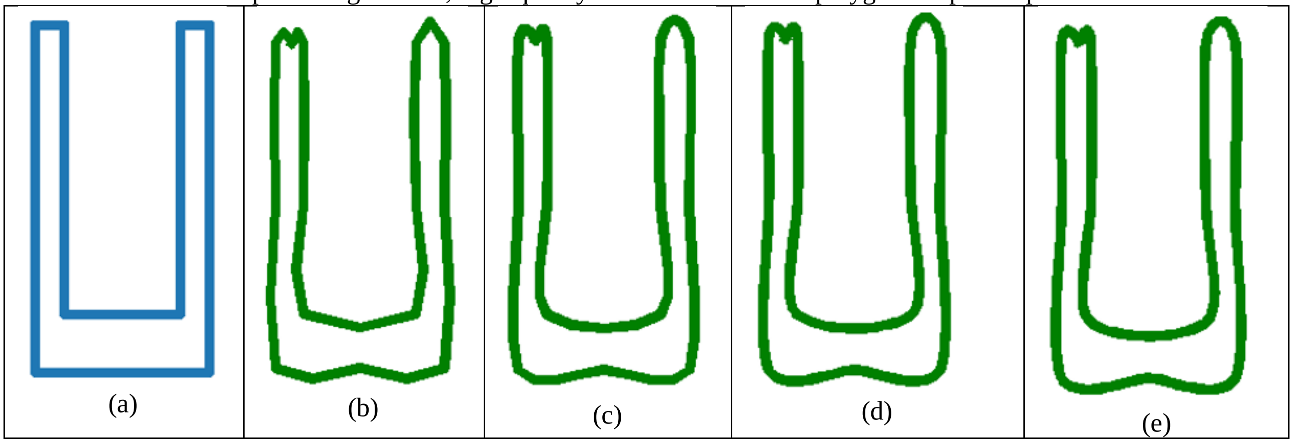


Figure 5.2: Refinement of a U-shaped polygon using the four-point subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

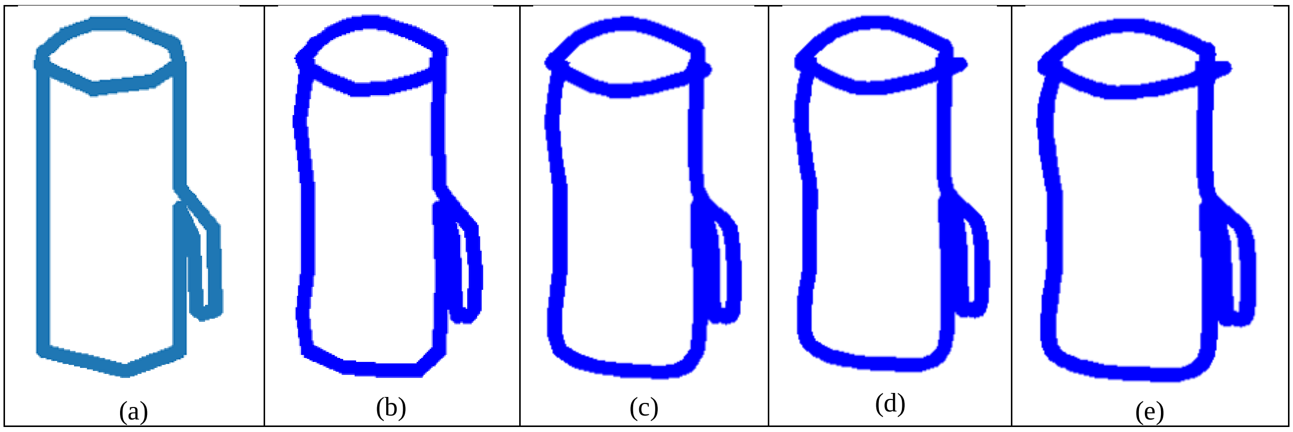


Figure 5.3: Refinement of a Mug-shaped polygon using the four-point subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

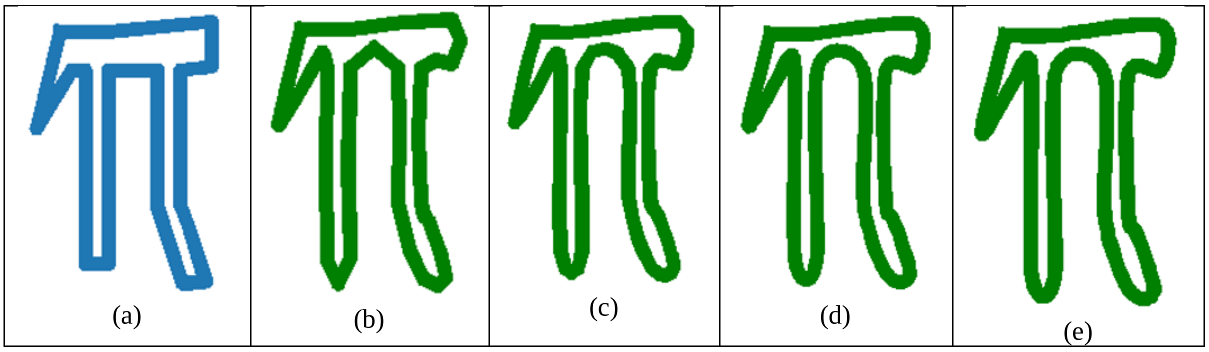


Figure 5.4: Refinement of a Pi-shaped polygon using the four-point subdivision scheme is illustrated in the sequence of figures from left to right: (a) the initial polygon, (b) after the 1st subdivision, (c) after the 2nd subdivision, (d) after the 10th subdivision, and (e) after the 20th subdivision.

6. Conclusion

In this presentation, we have successfully studied three prominent subdivision curve schemes: the CS scheme, the CSS scheme, and the FPS scheme. Each of these methods has demonstrated the ability to generate high-quality, smooth limit curves suitable for geometric modeling. To validate their performance, we constructed initial control polygons for several shapes, including the jar, mango, Pi, rabbit, five-finger, "U", and mug shapes. The experiments were conducted as follows:

- The CS scheme was applied to the jar, mango, Pi, and car shapes. The resulting curves after the 1st, 2nd, 10th, and 20th subdivisions are shown in Figures 3.2–3.5.
- The CSS scheme was implemented for the rabbit, five-finger, and Pi shapes, with results illustrated in Figures 4.2-4.4.
- The FPS scheme was applied to the "U", mug, and Pi shapes, with the corresponding results presented in Figures 5.2–5.4.

Through a comparative analysis of the resulting smooth limit curves, we observed that all three schemes effectively produce visually appealing and high-quality curves. These findings reaffirm the significance of subdivision techniques in geometric modeling, particularly in fields such as CG, CAGD, and CA. However, a number of challenges remain open for further research. There is a growing need to develop novel subdivision schemes to meet the demands of emerging practical applications. As future work, we aim to explore non-uniform subdivision schemes to achieve even higher-quality limit curves tailored to complex modeling scenarios.

Conflict of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work presented in this paper.

Acknowledgements

The authors express their sincere gratitude to the referees and the editor for their meticulous review and insightful comments, which have significantly improved the quality of this paper.

Funding

None.

References

[1] Cavaretta A. S., Dahmen W. and Micchelli C. A., Stationary Subdivision, Mem. American Math. Soc. 93 (1991), 453.

[2] Warren J. and Weimer H., Subdivision Methods for Geometric Design, Morgan, Kaufmann, 2002.

[3] Liao W., Li H. and Li T., Subdivision surface modeling technology. Springer, 2017.

[4] Li X. and Zheng J., Interproximate curve subdivision, J. Comput. Appl. Math. 244 (2013), 36-48.

[5] Li X. and Zheng J., An alternative method for constructing interpolatory subdivision from approximating subdivision, Comput. Aided Geom. Des. 29 (2012), 474-484.

[6] Boehm W., Farin G. and Kahmann J., A survey of curve and surface methods in CAGD, Comput. Aided Geom. Des. 1 (1984), 1-60.

[7] Farin G., A History of curves and surfaces in CAGD, Handbook of Computer Aided Geometric Design, Elsevier B.V., 2002.

[8] Liu Y., Shou H. and Ji K., Review of Subdivision Schemes and their Applications, Recent Pat. Eng. 16 (2022), e291221199620.

[9] Hameed R., Mustafa G., Hameed R., Younis J. and Salamd M. A. A. E., Modeling of curves by a design-control approximating refinement scheme, Arab J. Basic Appl. Sci. 30 (2023), 164-178.

[10] Alam M. N. and Li X., Non-uniform Doo-Sabin subdivision surface via eigen polygon, J. Syst. Sci. Complex 34 (2021), 3–20.

[11] Chaikin G. M., An algorithm for high speed curve generation, Comput. Graph. Image Process. 3 (1974), 346-349.

[12] Riesenfeld R., On Chaikin's algorithm, IEEE Comput. Graph. Appl. 4 (1975), 304–310.

[13] Lian J., A new four-point circular-invariant Corner-Cutting subdivision for curve design, Appl. Appl. Math. (AAM) 7 (2012), 464-487.

[14] Yang X., Point-normal subdivision curves and surfaces, Comput. Aided Geom. Des. 104 (2023), 102207.

[15] Dyn N., Levin D. and Gregory J. A., A 4-point interpolatory subdivision scheme for curve design, Comput. Aided Geom. Des. 4 (1987), 257-268.

[16] Amat S., Ruiz J., Trillo J. C. and Yáñez D. F., On a stable family of four-point nonlinear subdivision schemes eliminating the Gibbs phenomenon, J. Comput. Appl. Math. 354 (2019), 310-325.

[17] Lipovetsky E. and Dyn N., A weighted binary average of point-normal pairs with application to subdivision schemes, Comput. Aided Geom. Des. 48 (2016), 36-48.

[18] Dyn N., Kuijt F., Levin D. and Damme R. M. J. V., Convexity preservation of the four-point interpolatory subdivision scheme, Comput. Aided Geom. Des. 16 (1999), 789-792.