International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(14s) 2026 pp. 590-599
https://doi.org/10.70917/ijcisim-2026-4269
Article

# STRUCTURAL CHARACTERIZATION OF RECURSIVE PENTAGONAL SHELL GRAPHS USING RECURSIVE GROWTH PARAMETERS

**Rohit Auxilia E[1], G Jayalalitha[2]**

[1]Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
Email: rosyfrancis10@gmail.com, https://orcid.org/0009-0001-2894-8958
[2]Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
Email: g.jayalalithamaths.sbs@vistas.ac.in, https://orcid.org/0000-0002-3427-4697

**Abstract:** Recursive graph families are essential in graph theory because of their hierarchical structure and recursive growth patterns. In this paper, a new planar graph family called the Recursive Pentagonal Shell Graph (RPSG) is developed and built from a triangular seed graph using sequential recursive expansions. Each cycle adds three new vertices, resulting in an extra pentagonal shell while keeping the previous graph structure. The recursive construction is explained using vertex labelling, adjacency representation and recursive growth rules. To characterize the structural behaviour of the suggested graph, three new factors are introduced namely the Pentagonal Shell Index, Expansion Rate and Outer Boundary Size. Several structural features such as vertex development, recursive shell generation, connectedness, planarity and boundary preservation are supported by strong mathematical proofs. The suggested framework offers a systematic approach to analyse recursively layered planar graphs and serves as the foundation for future research into recursive graph families and their structural features.
**Keywords:** Planar graph, Recursive graphs, Pentagonal Shell graphs, Recursive graph construction, Structural graph theory, Recursive growth.

## 1. INTRODUCTION

Graph theory is a subfield of mathematics and computer science that uses diagrams with points and lines to visually describe mathematical concepts. A graph is a visual and mathematical depiction of a collection of things connected by links. Vertices or nodes represent interrelated items, whereas edges, arcs or lines join them together **[2]**. Graph theory is the study of graphs, which are made up of points and lines connected by vertices and edges. A graph is a graphical representation of a collection of things connected by links. In computer science, graph theory is the study of mathematical structures that represent pair-wise relationships between objects in a collection**[3]**. Graphs often known as trees are used to create data structures by connecting vertices and edges together.

Fractals can be found anywhere. They exist in both live and inanimate forms and are utilized in science and technology. Scientists discovered the properties of fractals relatively recently. Fractals provide a new world of gorgeous shape. By examining them, individuals attempt to construct the geometry of non-geometric objects **[4]**. For instance, create a mathematical model of the same fern leaf. Benoit Mandelbrot, a mathematician who revolutionized science, industry and art has paved the way for new directions. He owns the most famous fractal, the Mandelbrot set. The smallest pieces are remarkably similar to the total and self-similarity goes infinitely deep**[6]**.

Fractals of many types were initially examined as mathematical objects. Natural environments are full of approximate fractals. These objects exhibit self- similar structure throughout a broad but finite scale range **[5]**. Examples include clouds, snowflakes, mountains, river networks, cauliflower, broccoli and blood channel systems. Trees and ferns are fractal in nature and may be represented on a computer using a recursive technique.

In recent years, there has been a growing interest in creating novel graph families whose recursive structure naturally reflects hierarchical or fractal evolution. These graph families allow us to examine structural features that cannot be properly characterized using simply classical graph invariants like vertex degree, connectedness or chromatic number**[1]**. Instead, the recursive construction drives the development of graph family specific structural descriptors that can measure recursive growth and layered organization. Motivated by these results, this study presents the Recursive Pentagonal Shell Graph (RPSG), a new recursively built planar graph family. The construction starts with a triangular seed graph and expands recursively, adding three additional vertices with each iteration. These vertices are connected to the outermost shell of the existing graph using a fixed recursive rule, resulting in the development of one extra pentagonal shell with each recursive expansion. Throughout the creation, all prior vertices and edges are kept, which resulted in a connected, planar and recursively layered graph.

## 2. PROPOSED DEFINITIONS

The proposed definition is created for the new graph family.

**Definition 2.1**

Let $G_n$ be the Recursive Pentagonal Shell Graph obtained from the $n^{th}$ recursive iteration, where $n \geq 1$. The Pentagonal Shell Index of $G_n$, denoted by PSI ($G_n$), is defined as the total number of complete pentagonal shells formed during the recursive construction of $G_n$.

**Definition 2.2**

Let $G_n$ and $G_{n-1}$ be two consecutive Recursive Pentagonal Shell graph. The Expansion Rate is denoted by ER ($G_n$) is defined as the number of new vertices introduced during the transition from $G_{n-1}$ to $G_n$. **[1]**

**Definition 2.3**

The Outer Boundary Size of the Recursive Pentagonal Shell graph, indicated as OB ($G_n$) is defined as the number of newly inserted boundary vertices that form the outermost shell during the transition from $G_{n-1}$ to $G_n$. Because each recursive step adds exactly three new boundary vertices, OB ($G_n$) = 3, where n is more than two.

**Definition 2.4**

The Recursive Pentagonal Shell Graph $G_n$ is a recursive graph that begins with a triangular seed graph $G_1$. Each iteration adds three new vertices to construct the corresponding outer pentagonal shell, while retaining the previous iteration's vertices and edges.

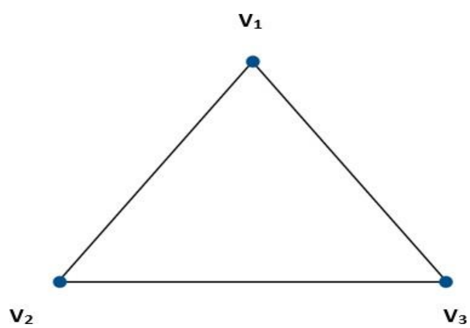


**Figure 2.1:** Triangular seed graph

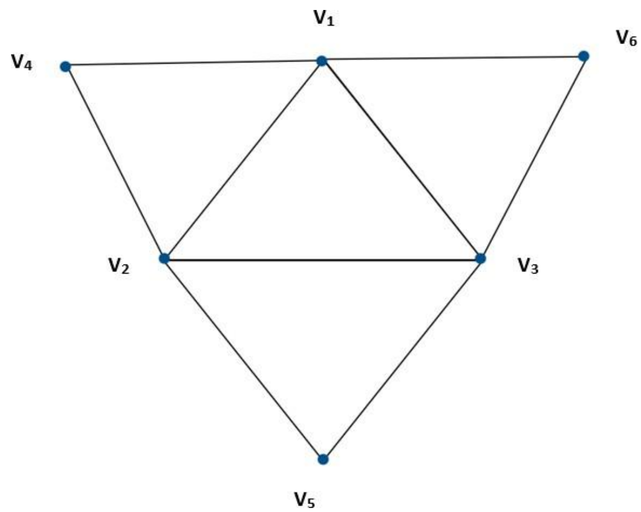


**Figure 2.2:** A graph with 6 vertices

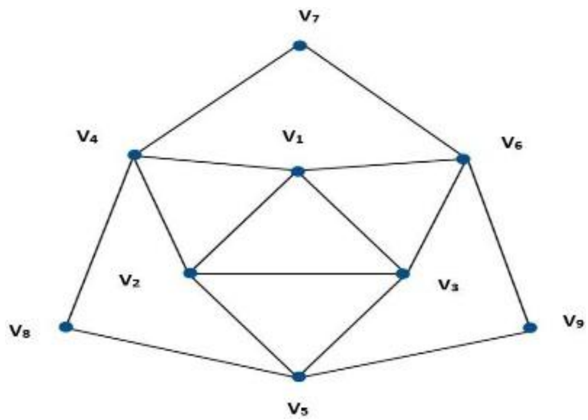


**Figure 2.3:** A graph with 9 vertices

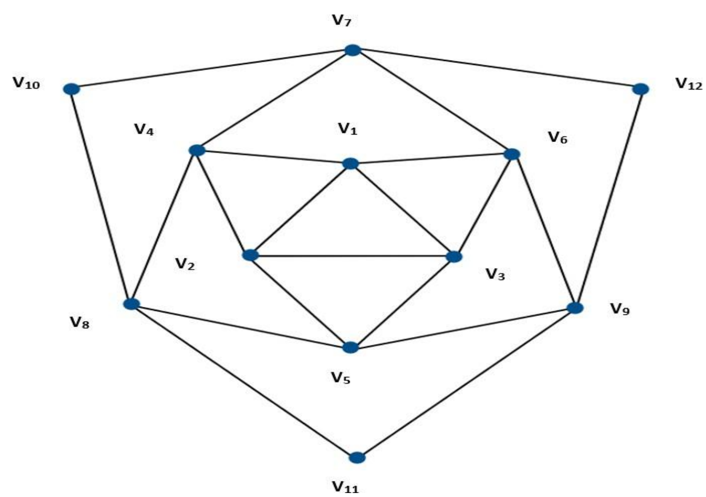


**Figure 2.4:** A graph with 12 vertices

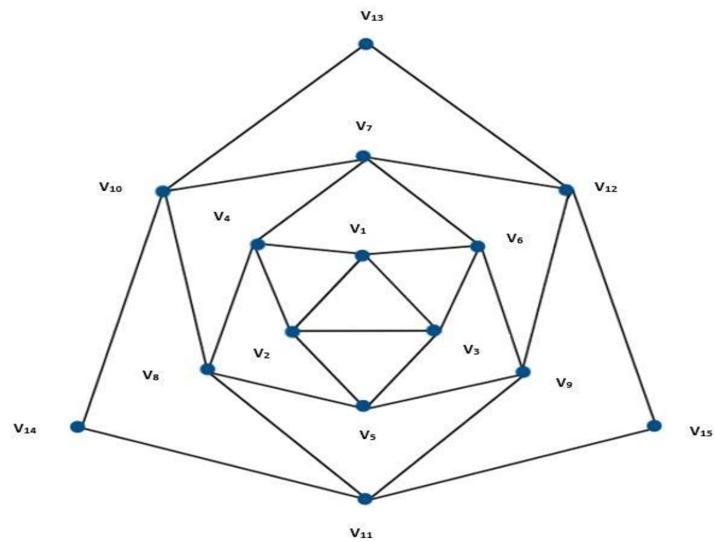


**Figure 2.5:** A graph with 15 vertices

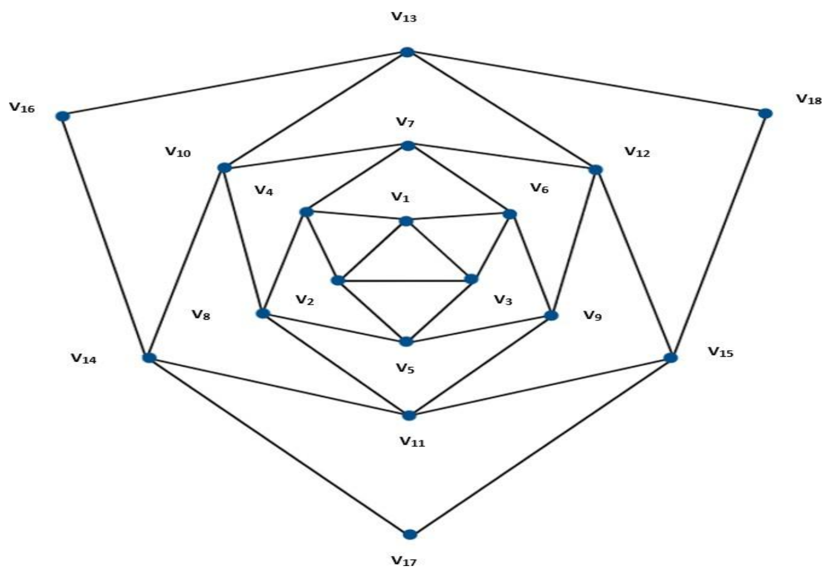


**Figure 2.6:** A graph with 18 vertices

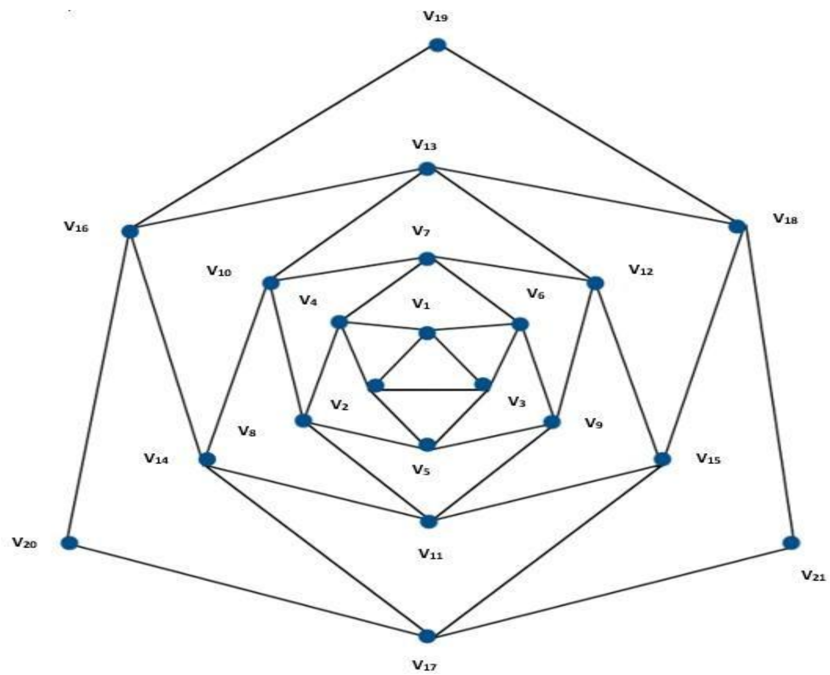


**Figure 2.7:** A graph with 21 vertices

## 3. RECURSIVE CONSTRUCTION OF THE PENTAGONAL SHELL GRAPH

- In the first iteration, the graph is constructed of a triangular seed with vertices V1, V2 and V3.
- In the second iteration, three extra vertices V4, V5 and V6 are added to G1's outer boundary. Their connections form the initial pentagonal shell around the triangular seed.
- The third iteration adds three new vertices V7, V8 and V9. These vertices connect to the vertices of the preceding outer shell, forming the second pentagonal shell.
- In the fourth iteration, vertices V10, V11 and V12 are added to build the third pentagonal shell around the existing graph.
- In the fifth iteration, three new vertices V13, V14 and V15 are added to the graph. These vertices are joined to the vertices of the existing outermost shell using the recursive algorithm, resulting in the fourth pentagonal shell while retaining the previously established structure.
- During the sixth iteration, the vertices V16, V17 and V18 are added beyond the fourth shell. Their inclusion expands the graph by creating the fifth pentagonal shell, which entirely encloses all inner shells without changing the previous graph.
- In the final iteration, vertices V19, V20 and V21 are added to complete the recursive structure. Connecting these vertices to the outer boundary of G6 yields the sixth outermost pentagonal shell, resulting in the complete Recursive pentagonal shell graph, which consists of 21 vertices arranged in six nested pentagonal shells encircling the initial triangular seed.

### *3.1 LABELLING EACH VERTEX*

- ✓ The graph G1 has 3 vertices, G1 = {V1, V2, V3}
- ✓ The graph G2 has 6 vertices, G2 = {V1, V2, V3, V4, V5, V6}
- ✓ The graph G3 has 9 vertices, G3 = {V1, V2, V3, V4, V5, V6, V7, V8, V9}
- ✓ The graph G4 has 12 vertices, G4 = {V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12}
- ✓ The graph G5 has 15 vertices, G5 = {V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15}
- ✓ The graph G6 has 18 vertices, G6 = {V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18}
- ✓ The graph G7 has 21 vertices, G7 = {V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V20, V21.}

## *3.2 ADJACENCY REPRESENTATION IN THE RECURSIVE PENTAGONAL SHELL GRAPH*

The recursive construction of the Recursive Pentagonal Shell Graph (RPSG) determines the arrangement of vertices and edges at each iteration. To investigate the structural aspects of the proposed graph its connectivity is represented by an adjacency list, an adjacency matrix and a degree sequence. These representations provide a thorough mathematical description of the graph and form the basis for determining its structural features.

### 3.2.1 ADJACENCY LIST:

The neighbourhood of each vertex in the graph is shown in the adjacency list. The neighbourhood of a vertex in a graph G = (V, E) is defined by N(v) = {u  V (G)| uv  E (G)}where the set of all vertices adjacent to v is represented by N (v). ∈∈

As a result, every vertex in the graph that is directly related to every other vertex is listed in the adjacency list. The adjacency relationship is symmetric because the Recursive pentagonal Shell graph is undirected, ensuring that if Vi is adjacent to Vj, then Vj is similar to Vi. By using the recursive approach to determine the neighbouring vertices that corresponds to each of the 21 vertices in the graph G7, the entire adjacency list is constructed in the table 3.1

**Table 3.1:** Adjacency list of Graph G7

| | $V_1$ | $V_2$ | $V_3$ | $V_4$ | $V_5$ | $V_6$ | $V_7$ | $V_8$ | $V_9$ | $V_{10}$ | $V_{11}$ | $V_{12}$ | $V_{13}$ | $V_{14}$ | $V_{15}$ | $V_{16}$ | $V_{17}$ | $V_{18}$ | $V_{19}$ | $V_{20}$ | $V_{21}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $V_1$ | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_2$ | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_3$ | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_4$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_5$ | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_6$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_7$ | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_8$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_9$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_{11}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_{12}$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $V_{13}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| $V_{14}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $V_{15}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $V_{16}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| $V_{17}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| $V_{18}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| $V_{19}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| $V_{20}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $V_{21}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

### 3.2.2 ADJACENCY MATRIX:

The adjacency list is then represented using an adjacency matrix. For a graph with n vertices, the adjacency matrix is defined as

$$A\,(G) = [a_{ij}]_{n} \times_{n} \text{ where}$$

$$a_{ij} = \begin{cases} 1, & \text{if } V_i \text{ is adjacent to } V_j \\ 0, & \text{otherwise} \end{cases}$$

Since the Recursive Pentagonal Shell graph is an undirected graph, the adjacency matrix is symmetric, i.e $a_{ij} = a_{ji}$ and all diagonal entries are zero because no vertex is adjacent to itself. For the $7^{th}$ iteration, the adjacent matrix is of order $21 \times 21$, providing a complete matrix representation of the graph connectivity.

### 3.2.3 DEGREE SEQUENCE

The degree sequence is obtained directly from the adjacency representation. The degree of a vertex is equal to the number of vertices adjacent to it and is represented by deg (v). Equivalently, the degree of a vertex is calculated by adding the entries from the corresponding row (or column) of the adjacency matrix. The degree sequence for the graph G7 is 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2 indicating that the first 18 vertices have degree four, whereas the following three vertices have degree two.

## 4. PROPOSED STRUCTURAL PARAMETERS

This section introduces three structural factors that characterize the recursive evolution of the proposed Recursive Pentagonal Shell Graph. These parameters define recursive shell construction, the graph's uniform expansion and the evolution of its outer boundary. These parameters are defined and their mathematical properties are described in the theorems that follow.

**Theorem 4.1**

Let $G_n$ be the Recursive Pentagonal Shell Graph obtained from the $n^{th}$ recursive iteration, where $n \geq 1$. Then the Pentagonal Shell Index of $G_n$ is

$$\mathrm{PSI}\,(G_n) = \begin{cases} 0 & , \; n = 1 \\ n-1 & , \; n \geq 2 \end{cases}$$

where PSI ($G_n$) denotes the total number of pentagonal shells formed in $G_n$.

**Proof**

The graph G1 is the initial triangular seed graph consisting of three vertices and three edges. Since no pentagonal shell exists in the initial construction, PSI ($G_1$) = 0.

During the recursive construction, each iteration adds one new pentagonal shell to the previously constructed graph. Consequently, the second iteration yields one pentagonal shell, the third iteration produces two pentagonal shells and this cycle repeats in the same way.

Therefore, in the nth iteration, the graph contains one pentagonal shell for each recursive expansion beyond the initial seed graph.

Hence,

$$\mathrm{PSI}\,(G_n) = n - 1\,, \quad n \geq 2$$

**Theorem 4.2**

Let $G_{n-1}$ and $G_n$ be two consecutive Recursive Pentagonal Shell Graphs, where $n \geq 2$. Then the Expansion Rate of the Recursive Pentagonal shell graph is constant and is given by ER ($G_n$) = $|V(G_n)| - |V(G_{n-1})| = 3$.

**Proof:**

The expansion rate is defined as the difference between the number of vertices in two consecutive Recursive Pentagonal Shell graphs. During the recursive construction of the graph, the graph $G_n$ is obtained from $G_{n-1}$ by introducing exactly three new vertices, while all vertices and edges of $G_{n-1}$ are preserved. Consequently, no existing vertex is removed during the recursive expansion.

From the theorem 4.1, the number of vertices in the Recursive Pentagonal Shell graph is $|V(G_n)| = 3n$. Similarly, $|V(G_{n-1})| = 3(n-1)$.

Substituting these expressions,

$$\begin{aligned} \mathrm{ER}\,(G_n) &= |V(G_n)| - |V(G_{n-1})| \\ &= 3n - 3(n-1) \\ &= 3n - 3n + 3 \\ &= 3. \end{aligned}$$

Hence, the number of vertices added during each recursive iteration is always equal to three. Therefore, the Expansion Rate remains constant throughout the recursive construction of the Recursive Pentagonal Shell Graph.

Thus, ER ($G_n$) = 3, $n \geq 2$.

**Theorem 4.3**

Let Gn represent the Recursive Pentagonal Shell graph obtained in the nth iteration , where $n \geq 2$. Then, the Outer Boundary Size of Gn, indicated by OB ($G_n$) is constant and equals to 3. That is each iteration introduces three additional vertices that form the graph's outermost boundary.

**Proof:**

The Recursive Pentagonal Shell Graph is constructed recursively from the initial triangular seed graph G1. During each recursive iteration , the construction process adds exactly three new vertices of the preceding outer shell, completing the next pentagonal shell while retaining all previously existing vertices and edges.

As the same recursive constructing rule is used for each iteration, the number of newly introduced border vertices remains constant throughout the recursive process. After the new shell is produced, the previous graph's outer boundary becomes an interior shell and the expanded graph's outer boundary is formed by the three newly added vertices.

Therefore, for every recursive iteration OB (Gn) = 3, n 2. ≥

Hence the Outer Boundary Size remains constant during the recursive construction of the Recursive Pentagonal Shell graph.

## 5. STRUCTURAL PROPERTIES OF RECURSIVE PENTAGONAL SHELL GRAPHS

Let $G_n$ denote the Recursive Pentagonal Shell Graph obtained in the $n^{th}$ iteration. The graph $G_1$ consists of a triangular seed graph with vertex set $\{v_1, v_2, v_3\}$. For each $n \geq 2$, three more vertices are added and joined to the outermost shell of $G_{n-1}$, generating a larger pentagonal shell.

**Theorem 5.1**

*For every integer $n \geq 1$, the number of vertices in $G_n$ is $|V(G_n)| = 3n$.*

**Proof**

Let G be a graph. The graph G1 has exactly three vertices $v_1$, $v_2$ and $v_3$. By definition, each recursive cycle adds exactly three new vertices. So, at $n-1$ recursive expansion, the total number of vertices is $3 + 3(n-1)$. As a result, $|V(G_n)| = 3n$, indicating that the total order of the recursive pentagonal shell graph increases linearly with iteration number.

**Theorem 5.2**

*For every $n \geq 2$, the Recursive Pentagonal Shell graph adds exactly three vertices on each iteration. Equivalently,*

$$|V(G_n)| - |V(G_{n-1})| = 3n.$$

**Proof**

The Recursive Pentagonal Shell graph is formed by recursively expanding the outermost pentagonal shell. During each recursive step, three new vertices are added to build the next outer shell, but all vertices from the preceding iteration are remain unchanged. As a result of, the newly added vertices are determined for the graphs' increase order from $G_{n-1}$ to $G_n$ .

From Theorem 3.1, $|V(G_n)| = 3n$ and $|V(G_{n-1})| = 3(n-1)$.

Hence $|V(G_n)| - |V(G_{n-1})| = 3n - 3(n-1)$. Thus, every iteration contributes exactly three additional vertices to the graph.

Therefore, $|V(G_n)| - |V(G_{n-1})| = 3n$. Hence proved.

**Theorem 5.3**

*For every integer $n \geq 2$, the Recursive Pentagonal Shell graph $G_n$ contains the graph $G_{n-1}$ as a subgraph. That is,*

$$G_{n-1} \subseteq G_n.$$

**Proof**

The recursive pentagonal graph is created by successively increasing the outermost pentagonal shell while retaining all previously created vertices and edges. Only three new vertices and their corresponding connecting edges are added to $G_{n-1}$ s' outer boundary during the creation process. It includes no deletion or modification of $G_{n-1}$ vertex or edge.

As a result, very vertex and edge from $G_{n-1}$ are retained in $G_n$ . $G_{n-1}$ is a subgraph of $G_n$ , and the recursive construction results in the nested sequence as, $G_1 \subseteq G_2 \subseteq G_3 \subseteq \ldots \subseteq G_n$

Hence, $G_{n-1} \subseteq G_n$ is proved.

**Theorem 5.4**

*Let $G_n$ be the recursive pentagonal shell graph formed after the nth iteration, with $n \geq 2$. Then the number of pentagonal shells in $G_n$ is $S(G_n) = n - 1$, where $S(G_n)$ is the number of pentagonal shells.*

**Proof**

The graph $G_1$ consists of the triangular seed graph produced by the vertices $V_1$, $V_2$ and $V_3$. Since no pentagonal boundary exists at this point, the graph has no pentagonal shells. During the building of $G_2$, three new vertices $V_4$, $V_5$ and $V_6$ are added and attached to the triangular seeds. These new connections form the first pentagonal shell, which surrounds the basic triangle.

Each iteration $G_n$ ($n \geq 3$) adds three more vertices to the outermost shell of $G_{n-1}$ . This recursive expansion creates exactly one new pentagonal shell while retaining existing shells. As a result, each repetition adds one more pentagonal shell to the total. As a result, after the first iteration, there are no pentagonal shells and each subsequent $n - 1$ repetition adds one more shell. Thus, the total number of pentagonal shells is $S(G_n) = n - 1$.

**Theorem 5.5**

*For every integer $n \geq 1$, the Recursive Pentagonal Shell graph $G_n$ is connected.*

**Proof**

*Using **the connected graph definition** A graph $G = (V, E)$ is said to be connected if every pair of different vertices in the graph has a path connecting them. In other words, there is a sequence of adjacent vertices beginning at u and ending at v for any two vertices $u, v \in V(G)$. Thus, each vertex in a connected graph can be reached from any other vertex by one or more edges.*

The triangular seed graph $G_1$ has vertices $V_1$, $V_2$ and $V_3$. Since $G_1$ is connected there is a path connecting each pair of vertices in a triangle. Assume that for a given integer $k \geq 1$, $G_k$ is connected.

The next outer pentagonal shell is formed by iteratively adding three more vertices to the graph $G_{k+1}$. These additional vertices are directly attached to the current connected graph by connecting to vertices on $G_k$'s outermost shell. Additionally, every newly added vertex is reachable from every other vertex in $G_k$ since it is connected to the current graph across the outer shell. Consequently, every pair of vertices in $G_{k+1}$ is connected by a path and therefore $G_{k+1}$ is connected

Hence, by the principle of mathematical induction, the recursive Pentagonal Shell graph $G_n$ is connected for every integer $n \geq 1$.

**Theorem 5.6**

*For every integer $n \geq 1$, the Recursive Pentagonal Shell graph $G_n$ is planar*.

**Proof**

*Using **the planar graph definition** the graph G is said to be planar if it can be drawn on a plane in such a way that no two edges intersect at their common end vertices, this representation of the graph is called a planar embedding.*

The theorem is proved using mathematical induction on iteration number n. In the base case, the graph $G_1$ is the triangular seed graph consisting of the vertices $V_1$, $V_2$ and $V_3$. Since a triangle can be embedded in the plane without any intersecting edges, $G_1$ is planar.

Assume that the graph $G_k$ admits a planar embedding for some integer. To form the additional shell, new vertices are added outside $G_k$ boundary and connected to the corresponding boundary vertices. The planar representation of $G_k$ remains unchanged due to the recursive extension, which do not intersect edges.

As a result, the graph $G_{k+1}$ is also planar embeddable. According to the mathematical induction any Recursive Pentagonal Shell graph $G_n$ is planar for all integers $n \geq 1$.

## 6. RESULTS AND DISCUSSIONS

The Recursive Pentagonal Shell Graph (RPSG) was constructed effectively using seven recursive iterations starting with a triangular seed graph. Each recursive iteration adds three additional vertices, producing a consistent and predictable growth pattern. The recursive construction keeps all previously formed vertices and edges while producing one more pentagonal shell at each stage. The adjacency representation and degree sequence support the proposed graph's recursive connectedness. To describe its structural behaviour, three new parameters were proposed and mathematically defined, the Pentagonal Shell Index (PSI), Expansion Rate (ER) and Outer Boundary Size (OBS). Closed form equations and structural theorems for these parameters were developed and extensively tested. The Recursive Pentagonal Shell Graph's connectedness and planarity were confirmed using mathematical induction. The collected findings show that the proposed graph has a consistent hierarchical structure and uniform recursive growth. These findings establish a mathematical foundation for studying recursively layered planar graph families and validate the suggested structural framework.

## 7. CONCLUSION

This work introduces the Recursive Pentagonal Shell Graph (RPSG), a new recursively generated planar graph family formed by sequential pentagonal shell expansions. To describe the graph production process, a systematic recursive construction technique was described which included vertex labelling and adjacency representation. Three new structural factors, the Pentagonal Shell Index, Expansion Rate and Outer Boundary Size were presented to describe the graph's recursive evolution. Rigorous mathematical proofs established fundamental structural features such as vertex development, recursive shell generation, connectedness and planarity.

The proposed graph demonstrates uniform recursive expansion while maintaining structural integrity at each iteration. Beyond typical graph invariants, the provided parameters add new perspectives to the analysis of recursively stacked graph families. Thu, the Recursive Pentagonal Shell Graph provides a valuable mathematical foundation for the future research into recursive graph structures, graph variants, spectral analysis and hierarchical network modelling.

**References:**


1. Rosen, K. H. *Discrete mathematics and its applications*. McGraw-Hill Education. (2019).
2. Chandrakant R. Gujar, Graph Theory and its Applications in Computer Science and Technology published by JETIR. ORG January 2023, Volume 10 Issue 1 www.jetir.org (ISSN-2349-5162).
3. Chris, C. J., Mary, L., & Kamali, R. AN ANALYSIS OF FRACTAL IN VARIOUS TYPES OF GRAPHS. *Chem. Bull*, *2023*, 94–102. https://doi.org/10.48047/ecb/2023.12.si13.009
4. Skums, P., & Bunimovich, L. (2019). *Graph fractal dimension and structure of fractal networks: a combinatorial perspective*. http://arxiv.org/abs/1912.11385
5. P. Tharaniya, G. Jayalalitha, Pethuru Raj and B. Sundaravadivazhagan. "*Advanced Fractal Graph Theory and Applications*. CRC Press, Taylor and Francis Group, London , New York, 2025.
6. Xu, J. "*Maximal Planar Graph Theory and the Four-Color Conjecture*" Peking University Beijing, China. https://doi.org/10.1007/978-981-96-4745-3. 2025
7. Narsingh Deo, Graph theory with Applications to Engineering and Computer Science, Dover Publications, INC. Mineola, New York, 2016.