Visualization of Cyclic, Dihedral, and Symmetric Groups

[*]B.M.S.M. Jayathilaka and R. Sanjeewa

Department of Mathematics, Faculty of Applied Sciences, University of Sri Jayewardenepura, Sri Lanka

DOI: https://doi.org/10.51584/IJRIAS.2025.100700050

Received: 26 June 2025; Accepted: 02 July 2025; Published: 06 August 2025

ABSTRACT

Group theory plays an important role in mathematics, providing a framework to understand symmetries and structures across various fields. This study explores visualization techniques for cyclic, dihedral, and S_2, S_3, S_4 symmetric groups. Visual tools such as Circle Representations and Cayley graphs are employed to illustrate and analyze group properties, including element orders, inverses, group operations, and subgroup structures. In the case of cyclic groups, in addition to Cayley graphs, the circle representation was also used to geometrically model the group structure. For more complex groups such as symmetric groups S_2, S_3, S_4 and dihedral groups D_{2n}, Cayley graphs were studied to understand how group generators relate to group elements and to visualize their structural symmetries and subgroup formations. While Cayley graphs for cyclic, dihedral, and symmetric groups have been previously visualized, for instance, through platforms like Group Explorer (Carter, n.d.), there remains a lack of algorithms and tools that enable users to perform group operations, compute element orders and inverses, and identify subgroup structures directly from these visualizations. For cyclic, dihedral, and symmetric groups S_4, algorithms were developed to automate tasks such as determining the order and inverse of elements, performing group operations, and identifying subgroup elements using Cayley graphs. These algorithms were realized through Python-based web applications developed with the Flask framework. The applications allow users to interactively explore group visualizations and perform computations related to group properties, enhancing both understanding and usability for learners and researchers. The results demonstrate that visual representations, when supported by algorithmic analysis, provide powerful tools for grasping abstract group theoretic concepts. The developed applications successfully link theoretical foundations with computational exploration, offering an effective means for learning, teaching, and further research in abstract algebra. This study highlights how visualization bridges intuition and formalism in group theory and contributes to educational tools and computational mathematics.

Keywords: Cayley Diagrams, Circle Representation, Group Visualization, Python Web Application

INTRODUCTION

Group theory is a key component of abstract algebra, which is essential for understanding mathematical structures and their symmetries. Visualization methods provide us with a unique perspective on group properties, offering insights that are often challenging to grasp through algebraic definitions alone. By converting algebraic structures into intuitive visual formats, visualization plays a crucial role in both learning and research, particularly in understanding complex group dynamics.

Over the past few decades, there has been growing interest in visualizing mathematical groups as a tool to enhance understanding, particularly in educational and research settings. By converting abstract group properties into intuitive visual formats, these methods aim to bridge the gap between theory and perception, allowing both students and researchers to explore group dynamics without requiring complex algebraic computations.

The foundation of the circle representation of cyclic groups was established long ago. The idea of representing elements of cyclic groups as equally spaced points on a circle has deep historical roots in classical mathematics. As early as the 18th and 19th centuries, foundational work by Leonhard Euler and Carl Friedrich Gauss contributed significantly to the mathematical idea behind modern circle representations of cyclic groups. Euler studied the n^{th} roots of unity: complex numbers of the form $e^{\frac{2\pi ik}{n}}$, and established their expression as points equally spaced around the unit circle using what is now known as Euler's formula. This introduced a natural angular interpretation of periodic or cyclic behavior. Gauss investigated the division of the circle into n equal parts in his work on cyclotomic equations, particularly in the context of constructing regular polygons, as detailed in his Disquisitiones Arithmeticae (Gauss, 1801/1966). The visual representation of group elements on a circle was not specifically addressed by either Euler or Gauss, but these techniques serve as the foundation for modern circle representations, in which each group element represents a distinct angle of the unit circle.

In addition, Cayley diagrams serve as a powerful tool for visualizing finite groups. The concept of Cayley diagrams was first introduced by Arthur Cayley (Cayley, 1878) in 1878 as a way to visualize abstract group structures. These diagrams, now widely used in group theory, represent group elements as nodes and group operations as colored or directed edges labeled by generators. Cayley's introduction laid the foundation for many modern approaches to visual group theory. Nathan Carter's Visual Group Theory (Carter, 2009) elaborates on the use of Cayley diagrams for cyclic groups, focusing on the relationship between elements and their generators. Also, it elaborates on the use of Cayley diagrams for dihedral groups as well as symmetric groups. Nathan Carter's Group Explorer (Carter, n.d.), offers effective examples of such diagrams of cyclic groups, dihedral groups and symmetric groups.

This study focuses on visualizing cyclic groups C_n , dihedral groups D_{2n} , and symmetric groups S_2, S_3, and S_4, utilizing methods such as circle representations and Cayley diagrams. Python Flask based web applications were developed to draw Cayley graphs for these groups, compute inverses, determine orders, perform group operations, and identify subgroup elements.

Although current visualization methods are effective for cyclic, dihedral, and small ordered symmetric groups, challenges remain in visualizing larger non-abelian groups and higher order symmetric groups like S_5 and S_6. The complexity increases with the size and structure of the group, making simple visual representations insufficient. Nevertheless, advancements in computational tools, such as dynamic graph visualization libraries and interactive web applications, offer promising directions. Future work could extend these techniques to visualize groups such as S_5 and S_6, multiplicative groups modulo n, and beyond, combining visual intuition with algorithmic automation.

The objectives of this study are:

- To explore various visualization techniques for groups, including circle representations and Cayley diagrams.
- To examine how visualization, enhance understanding of group operations, element orders, inverses, and subgroups.
- To analyze methods for identifying group properties directly from visual representations.
- To develop Python based web applications in which algorithms are written based on visual methods to automate the visualization and analysis of group properties for selected groups.

METHODOLOGY

Group Selection and Visualization Techniques

In this research, mainly 3 types of group visualizations were studied. They are cyclic, dihedral, and symmetric groups. In symmetric groups only S_2, S_3, and S_4 visualization was studied. Because in the symmetric group S_n the order is given by n!, the order will be higher when n is a higher integer. So, if we consider $n \geq 5$, visualization of these groups will be difficult. But for specifically S_4 visualization was studied since it only has 24 elements, but the Cayley diagram of it is a 3-D figure which has a rich structure. Cyclic and dihedral groups were also chosen to study, as they are finite, well understood, and have a rich structure.

There are 2 visualization methods studied in this research. The circle representation is especially useful for cyclic groups, as it highlights their cyclic nature and simplifies the understanding of element orders and subgroups. One of the most common visualizations is Cayley graphs, which offer a more general framework for visualizing groups, providing a clear picture of how elements relate to each other through group operations.

Cyclic Groups C_n

Consider a cyclic group C_n of order n. In the circle representation, the elements of a cyclic group $C_n = \{g_0, g_1, \ldots, g_{n-1}\}$ are placed as equally spaced points on a circle. Each element g^i corresponds to a rotation by an angle,

$$\theta_i = \frac{2\pi i}{n}, \quad 0 \leq i \leq n-1.$$

Thus, the group can also be represented as

$$C_n = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$$

For example, in the circle representation of the cyclic group C_{12}, the group elements are denoted as

$$C_{12} = \{\theta_0, \theta_1, \ldots, \theta_{11}\}$$

where each θ_i represents a rotation by an angle of

$$\theta_i = \frac{2\pi i}{12} = \frac{\pi i}{6}, \quad \text{for } i = 0,1, \ldots, 11$$

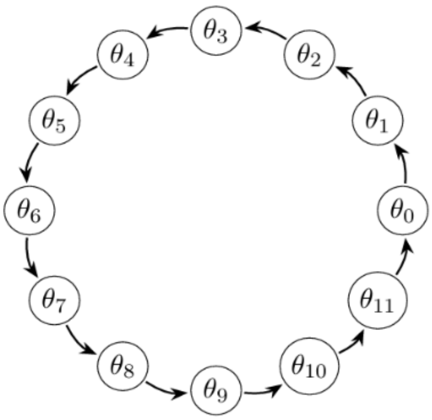

Figure 1: Circle representation of C_{12}

Let $\theta_a = \frac{2\pi a}{n}$ and $\theta_b = \frac{2\pi b}{n}$ where $a, b \in \mathbb{Z}_n$. We define group operation $\theta_a + \theta_b$ by

$$\theta_a + \theta_b = \frac{2\pi}{n}\big((a+b) \bmod n\big)$$

And we denote $\theta_a + \theta_b$ by $\theta_{a+b\ (\mathrm{mod}\ n)}$

$$\text{i.e. } \theta_a + \theta_b = \theta_{a+b\ (\mathrm{mod}\ n)}$$

We can perform the group operation on two elements θ_a and θ_b in C_n using the circle representation of C_n. We start from one of the elements (without loss of generality), θ_a, and traverse the elements in the counterclockwise direction by taking steps b (where b comes from the other element θ_b). The element we reach is the result of the group operation $\theta_a + \theta_b$.

The inverse of an element $\theta_k \in C_n = \{\theta_0, \theta_1, \ldots, \theta_{n-1}\}$, where $k \in Z_n$ is given by θ_{n-k}. To find the inverse of θ_k visually, starting from θ_0 we go in the clockwise direction by an angle of θ_k. The element we reach is the inverse of θ_k.

Another method is starting from θ_0, traversing through the elements in the clockwise direction by taking k steps (where k comes from element θ_k). The element we reach is the inverse of θ_k.

The order of an element θ_a in C_n is given by:

$$|\theta_a| = \frac{n}{\gcd(a, n)}$$

We can find the order of an element θ_a using the circle representation. Starting from θ_0, count how many times we have to traverse through the elements by step size 'a' to return to the identity element θ_0. Then that count is the order of the element θ_a.

By Lagrange's Theorem, if d | n, then C_n has a unique subgroup of order d given by:

$$H_d = \langle\theta_\lambda\rangle = \{\theta_0, \theta_\lambda, \theta_{2\lambda}, \ldots, \theta_{\lambda(d-1)}\}, \quad \text{where } \lambda = \frac{n}{d}$$

For any $\theta_{\lambda k}$ in H_d, the associated angle is:

$$\theta_{\lambda k} = \frac{2\pi k}{d} \quad \text{for } 0 \leq k \leq d-1$$

We can find the elements of subgroups of C_n visually by looking at the diagram. Starting from θ_0, if we traverse the elements of the diagram counterclockwise, with increments of $\lambda = \frac{n}{d}$, yield the elements of H_d, which is the subgroup of order d.

Cayley diagrams provide a different way of representing cyclic groups. In a Cayley diagram for a cyclic group, the elements are depicted as vertices of a graph, and the group operations are represented by directed edges between these vertices. In a Cayley diagram for a cyclic group, the edges form a simple cycle, which corresponds to the generator's action in generating the entire group.

Consider the cyclic group $C_n = \langle g\rangle$ of order n, generated by a single element g. Then in the Cayley graph of C_n, there exist edges from g^i to g^{i+1} for each $i \in \{0, 1, \ldots, n-1\}$, where indices are taken modulo n.

Since the cyclic group C_n is generated by a single generator g, only one color is needed to represent all the edges in its Cayley graph. In this chapter, edges corresponding to multiplication by g are colored Red.

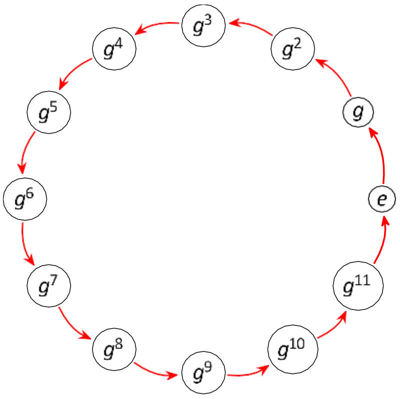

Figure 2: Cayley graph of the cyclic group C_{12} with generator g.

Let g^a and g^b be elements in $C_n = \langle g \rangle = \{e, g, g^2, \dots, g^{n-1}\}$, where g is the generator of the group. The group operation in C_n is defined by:

$$g^a \cdot g^b = g^{a+b \mod n}.$$

We can perform the group operation on two elements g^a and g^b in C_n using the Cayley diagram. Starting from the vertex representing g^a, we follow the directed edges labeled by the generator g, 'b' times. Since each edge corresponds to multiplication by g, the traversal leads to the vertex representing g^{a+b}, which is the result of the group operation $g^a \cdot g^b$ with the exponent taken modulo n.

The inverse of an element $g^k \in C_n$ where $k \in Z_n$, is given by g^{n-k}. To find the inverse of $g^k \in C_n$ using the Cayley diagram, we can proceed as follows:

Starting from the identity element $e = g^0$, we move along the edges in the clockwise direction (i.e., against the direction of the arrows) by taking k steps. The element we reach is g^{n-k}. The order of an element g^a in C_n is given by:

$$|g^a| = \frac{n}{\gcd(a, n)}$$

To find the order of an element $g^a \in C_n$ using the Cayley diagram, start at the identity element $e = g^a$, and repeatedly move along the arrows by steps of size a (i.e., follow every a^{th} arrow as if multiplying by g^a). Count how many such steps are needed to return back to the identity element e. This count is the order of the element g^a.

By Lagrange's Theorem, if d | n, then $C_n = \langle g \rangle$ has a unique subgroup of order d, given by:

$$H_d = < g^\lambda > = \{e, g, g^{2\lambda}, \dots, g^{(d-1)\lambda}\}, \text{ where } \lambda = \frac{n}{d}.$$

Elements of H_d, which is the Subgroup of $C_n = \langle g \rangle$ of order d, can be identified in the Cayley diagram by observing the elements generated by powers of g^λ, where $\lambda = \frac{n}{d}$ and d divides n.

Starting from the identity element e, Traversing the vertices of the diagram, in the counterclockwise direction with step size λ, yields the elements of the subgroup $H_d = \langle g^\lambda \rangle$, which has order d. The vertices we reach along the way in each step are the elements of the subgroup.

Dihedral Groups D_{2n}

The Dihedral group D_{2n} is the group of symmetries of a regular n-gon ($n \geq 3$). It consists of 2n elements: n rotations and n reflection.

The following edges exist in the Cayley graph:

- From rotation r^i to rotation r^{i+1}.
- From reflection sr^i to reflection sr^{i+1}.
- Between reflection sr^i and rotation r^i.

Here, $i \in \{0, 1, \dots, n-1\}$.

Since only two generators are there in D_{2n}, only two colors are enough to color the edges. In here, for generator r, blue color is used, and for generator s, green color is used.

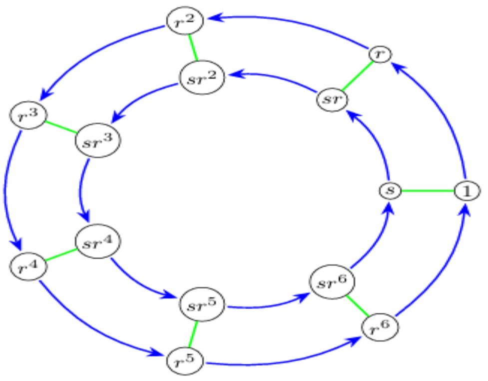

Figure 3: Cayley Diagram of the Dihedral Group D_{14}.

There are four types of how we can operate two elements in D_{2n}.

1. Rotation × Rotation

The product of two rotations r^a and r^b is given by:

$$r^a \cdot r^b = r^{(a+b) \bmod n}$$

We can identify the product of two rotation elements r^a and r^b using the Cayley graph. We have to start from one of the rotation elements (without loss of generality) r^b and traverse through the rotation elements in the counterclockwise direction by step a (Here, a is the power of the other rotation element). Then that rotation element is the answer of $r^a \cdot r^b$

2. Reflection × Reflection

The product of two reflections sr^a and sr^b is given by:

$$sr^a \cdot sr^b = s(r^a s)r^b = s \cdot sr^{-a} \cdot r^b = s^2 r^{-a+b} = 1 \cdot r^{-a+b} = r^{(b-a) \bmod n}$$

We can identify the product of two reflection elements sr^a and sr^b using the Cayley graph. We have to start from the second reflection element sr^b and go to the corresponding rotation element r^b. Then traverse through the rotation elements in the clockwise direction by step a (Here, a is the power of the first reflection element). Then that rotation element is the answer of $sr^a \cdot sr^b$.

3. Rotation × Reflection

The product of rotation, r^a followed by reflection, r^b is given by:

$$r^a \cdot sr^b = (r^a s)r^b = sr^{-a} \cdot r^b = sr^{b-a} = sr^{(b-a) \bmod n}$$

We can identify the product of a rotation element followed by a reflection element, $r^a \cdot sr^b$ using the Cayley graph. We have to start from the reflection element sr^b and traverse through the reflection elements in the clockwise direction by step a (Here, a is the power of rotation element). Then that reflection element is the answer of $r^a \cdot sr^b$.

4. Reflection × Rotation

The product of reflection, sr^b. followed by rotation, r^b is given by:

$$sr^a \cdot r^b = sr^{(a+b) \bmod n}$$

We can identify the product of a reflection element followed by a rotation element, $sr^a \cdot r^b$ using the Cayley graph. We have to start from the reflection element sr^b and traverse through the reflection elements in the counterclockwise direction by step b (Here b is the power of the rotation element). Then that reflection element is the answer of $sr^a \cdot r^b$.

The order of any reflection element sr^k is 2. The order of a rotation element r^k is given by:

$$|r^k| = \frac{n}{\gcd(k,n)}$$

where $k \in \{0, 1, \ldots, n-1\}$. To find the order of a rotation element $r^k \in D_{2n}$ using the Cayley diagram, start at the identity element 1, and repeatedly move along the rotation elements in counter clock wise direction by steps of size k (i.e., follow every k^{th} arrow as if multiplying by r^k). Count how many such steps are needed to return back to the identity element 1. This count is the order of the element r^k).

The inverse of any reflection element sr^k). is itself. The inverse of a rotation element r^k is r^{n-k}. We can find the inverse of r^k by looking at the Cayley graph of the group. To find the inverse of r^k, starting from 1 we traverse through rotation elements in the clockwise direction k times. Then that k^{th} element is the inverse of r^k.

There are three types of subgroups in dihedral groups: cyclic subgroups, reflection subgroups (a special case of mixed subgroups), and mixed subgroups.

Cyclic subgroups consist solely of rotations and can be identified by the Cayley graph. For any divisor d of n, there exists a subgroup of D_{2n} that contains rotations generated by r^d:

$$< r^d > = \{1, r^d, r^{2d}, \ldots, r^{d(\frac{n}{d}-1)}\}$$

The order of this subgroup is $\frac{n}{d}$. If d is a divisor of n, then starting from 1, and traversing the elements of the outer circle (rotation elements) in the counterclockwise direction with increments of the size d, yields the elements of a subgroup of order $\frac{n}{d}$.

For any reflection sr^k of D_{2n}, the set $\{1, sr^k\}$ forms a subgroup of order 2 (reflection subgroups).

Mixed subgroups are generated by a reflection element and a rotation element. For any divisor d of n, $\langle r^{\frac{n}{d}}, sr^i \rangle$ is a subgroup of order 2d, where $i \in \{0, 1, \ldots, n-1\}$. To find the elements of the mixed subgroup $\langle r^{\frac{n}{d}}, sr^i \rangle$ of order 2d, we can find the d rotation elements by starting from 1, and traversing the elements of the outer circle (rotation elements) in the counterclockwise direction with increments of the size $\frac{n}{d}$. As for the d reflection elements, starting from sr^i and traverse the elements of the inner circle (reflection elements) in the counterclockwise direction with increments $\frac{n}{d}$ yields those d reflection elements.

Symmetric Groups S_n

The symmetric group on *n* elements, denoted by S_n, is the group of all bijective functions (permutations) from the set $\{1, 2, \ldots, n\}$ to itself. That is,

$$S_n = \{\sigma : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\} \mid \sigma \text{ is a bijection}\},$$

with the group operation being a composition of functions. The identity element is the identity permutation, and the inverse of a permutation is its inverse as a function.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Symmetric Group of order 2! which is denoted by S_2 contains only two elements: identity $e = (1)(2)$ and transposition $(1\ 2)$.

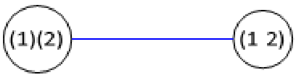

Figure 4: Cayley Graph of S_2.

S_3 consists of all permutations of 3 elements, with $|S_3| = 6$. Its elements include:

- Identity: $e = (1\ 2\ 3)$
- Transpositions (2-cycles): $(1\ 2), (1\ 3), (2\ 3)$
- 3-cycles: $(1\ 2\ 3), (1\ 3\ 2)$

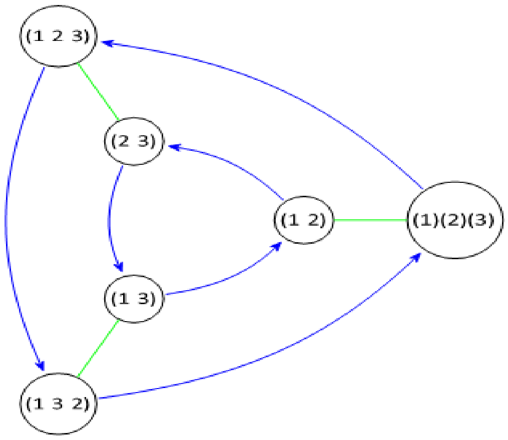

Figure 5: Cayley Graph of S_3.

The Cayley graph of S_3 is constructed using the generating set $\{(1\ 2\ 3), (1\ 2)\}$. The symmetric group S_3 and the dihedral group D_6 are isomorphic as abstract groups. Both groups consist of six elements and share the same multiplication table up to relabeling of elements. This means there exists a bijective map between the elements of S_3 and D_6 that respects the group operation. In particular:

- S_3 is the group of all permutations of 3 elements.
- D_6 is the group of symmetries of a regular triangle, including 3 rotations (including identity) and 3 reflections

There exists an isomorphism between the two groups:

$$S_3 \cong D_6$$

As a consequence, their Cayley graphs are isomorphic under a suitable choice of generators.

If we take the generators of S_3 to be a transposition and a 3-cycle, for example:

$$S_3 = \langle (1\ 2), (1\ 2\ 3) \rangle,$$

and the generators of D_6 to be a reflection s and a rotation r, then the corresponding Cayley graphs are structurally identical.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Since S_3 and D_6 are isomorphic, the same methods used to analyze the group structure of D_6, such as determining the order of elements, finding their inverses, performing group operation among two elements and identifying subgroups, can also be applied directly to S_3.

S_4 consists of all permutations of 4 elements, with $|S_4| = 24$. Its elements include:

- Identity: e.
- Transpositions (order 2): $(1\ 2), (1\ 3), (1\ 4), (2\ 3), (2\ 4), (3\ 4)$
- 3-cycles (order 3): $(1\ 2\ 3), (1\ 3\ 2), (1\ 2\ 4), (1\ 4\ 2), (1\ 3\ 4), (1\ 4\ 3), (2\ 3\ 4), (2\ 4\ 3)$.
- 4-cycles (order 4): $(1\ 2\ 3\ 4), (1\ 2\ 4\ 3), (1\ 3\ 2\ 4), (1\ 3\ 4\ 2), (1\ 4\ 2\ 3), (1\ 4\ 3\ 2)$.
- Double transpositions (order 2): $(1\ 2)(3\ 4), (1\ 3)(2\ 4), (1\ 4)(2\ 3)$.

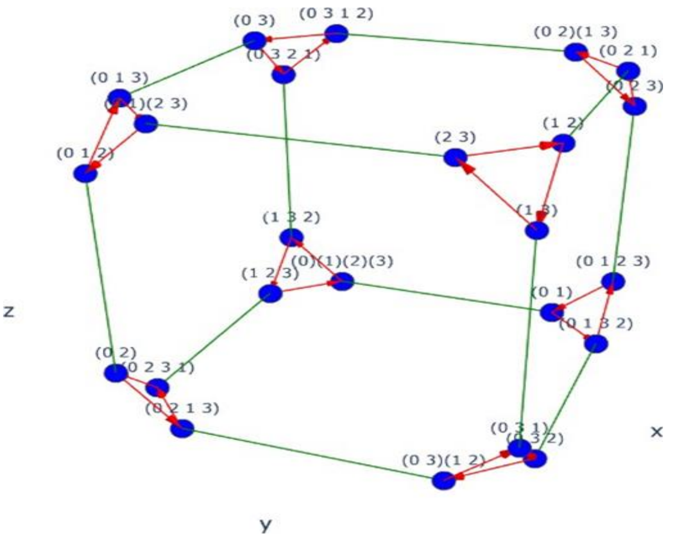

Figure 6: Cayley Graph of S_4 - Truncated Cube Version

The Cayley graph of the symmetric group S_4 has been constructed using a layout based on the geometry of a truncated cube, which is one of several possible layouts suited for visualizing this group.

For this visualization, the generators used are $(1\ 3\ 2)$ and $(0\ 1)$.

$$S_4 = \langle (1\ 3\ 2), (0\ 1) \rangle$$

- The permutation $(0\ 1)$ is represented by green edges.
- The permutation $(1\ 3\ 2)$ is represented by red edges.

In the Cayley diagram of a group, each vertex represents a group element, and each directed edge corresponds to the application of a generator. Given a set of generators, an edge sequence for a vertex (i.e., a group element) describes a specific sequence of generators that must be applied, starting from the identity element, to reach that vertex.

More formally, for a group G with a generating set $\{g_1, g_2, \ldots, g_k\}$, the edge sequence for an element $h \in G$ is an ordered sequence $(g_{i_1}, g_{i_2}, \ldots, g_{i_m})$ such that:

$$g_{i_1} \cdot g_{i_2} \cdot \cdots \cdot g_{i_m} = h$$

and each g_{i_j} is one of the generators of G.

The order of an element $h \in G$ is the smallest positive integer k such that

$$h^k = e,$$

where e is the identity. To find the order of an element $h \in G$ using the Cayley diagram:

1. Determine an edge sequence of h, i.e., a sequence of generators that leads from the identity to h. Denote it as $s = (s_1, s_2, \ldots, s_m)$, where each $s_i \in \{g, r\}$

2. Starting at the identity vertex e, follow the edges labeled by the sequence s to reach h.

3. Repeat the same edge sequence starting from the current vertex.

4. Continue repeating the edge sequence until you return to the identity element.

5. The number of repetitions needed to return to e is the order of the element h.

To find the inverse of a group element $h \in S_4$ using the Cayley diagram, we use the following graphical method based on the concept of edge sequences:

1. Determine the edge sequence of the element h. This is the ordered list of generator applications (e.g., colors or labels like g_1, g_2) used to reach h from the identity vertex.

2. Reverse the edge sequence to get the path corresponding to the inverse. If the original edge sequence is $s = (s_1, s_2, \ldots, s_k)$, the reversed edge sequence is $(s_k, s_{k-1}, \ldots, s_1)$.

3. Important**:** When following the reversed edge sequence, you must assume the directions of all edges in the Cayley graph are also reversed. This means that if there is an edge from g_1 to g_2 labeled s_i, then in the reverse process we consider it as an edge from g_2 to g_1.

4. Start from the identity element e and follow the reversed edge sequence through the reversed graph.

5. The endpoint reached after traversing the full reversed sequence is the inverse h^{-1}.

In the symmetric group S_4, the group operation is function composition of permutations, applied from right to left. That is, given two elements $g, h \in S_4$, their composition g ⍰ h means applying h first, then g.

To perform the group operation g ⍰ h (i.e., apply h after g) for any two elements in S_4, we use a graphical method based on the edge sequences in the Cayley diagram:

1. Select Elements: Choose two elements $g, h \in S_4$. We want to compute g ⍰ h, which corresponds to first applying h, then g.

2. Obtain Edge Sequence of h: Extract the edge sequence of the element h. This is the sequence of generators (with color and direction) used to reach h from the identity element in the Cayley diagram.

3. Traverse from g Using h's Edge Sequence**:** Starting from the vertex corresponding to g, follow the edge sequence of h, using the same generator order, directions, and colors as in the original graph. The endpoint of this traversal is the composition g ⍰ h.

We determine the elements of a subgroup $H \leq S_4$ by tracing paths in the Cayley diagram according to the subgroup's generators. The method differs slightly depending on whether the subgroup has one or two generators.

1. Subgroups with One Generator: These subgroups are cyclic and behave like $\langle g \rangle = \{e, g, g^2, \ldots\}$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Step 1: Start from the identity element e.

Step 2: Identify the generator g and its edge sequence (i.e., the path from e to g using the graph's colored edges).

Step 3: From g, follow the same edge sequence to reach g^2, and from there again to get g^3, and so on.

Step 4: Continue this process until the identity is reached again.

The set of all such vertices visited forms the cyclic subgroup.

2. Subgroups with Two Generators: Most subgroups of S_4 are generated by two elements. In such cases, at least one generator is of order 2; denote this one by g_2 and the other generator by g_1.

- **Special Cases:**

I. If the generators are $g_1 = (1\ 3\ 2)$ and $g_2 = (0\ 1)$, the subgroup is the group S_4.

II. If the generators are $g_1 = (1\ 3\ 2)$ and $g_2 = (0\ 2)(1\ 3)$, the subgroup is the alternating group A_4, consisting of all even permutations (i.e., permutations that can be written as a product of an even number of transpositions).

- **General Case:**

Step 1: Begin with the identity e, and include both generators g_1 and g_2 in the initial set of subgroup elements.

Step 2: Determine the edge sequence of g_1, i.e., the ordered set of generator labeled edges used to reach g_1 from the identity in the Cayley diagram.

Step 3: Starting from g_1, follow the same edge sequence to obtain a new element. Add this element to the subgroup set.

Step 4: Repeat this process: from each newly obtained element, apply the same edge sequence of g_1 again, until the resulting element is already in the subgroup set. Stop this traversal once a repeat is found.

Step 5: Now, perform the same type of traversal starting from g_2:

I. From g_2, follow the edge sequence of g_1 to obtain a new element.

II. From that new element, again apply the edge sequence of g_1, and continue this process as before.

III. Each new element found is added to the subgroup set, until an already discovered element is reached, at which point that path is terminated.

Step 6: The final collection of distinct elements obtained through these steps constitutes the subgroup generated by $\langle g_1, g_2 \rangle$.

Software Implementation

As part of this research, three Python-based web applications were developed using the Flask framework to visualize and explore the structure of the cyclic group C_n, dihedral group D_{2n}, and symmetric group S_4. These interactive applications enable users to engage with the algebraic properties of C_n, D_{2n} and S_4 through real-time visualizations and computations presented in a user-friendly web interface. All operations are carried out by custom Python algorithms written specifically for this study. The application runs every computation in real time: user input is processed by the Flask server, which invokes the Python back-end logic to compute results and immediately returns the output via the interface.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Cyclic group visualization tool

The tool is grounded in key concepts from cyclic group theory. It allows users to specify a value for n, generate the corresponding group C_n, and perform various operations to understand its structure and subgroups better.

The application supports the following functionalities in real time:

- Visualization of the Cayley diagram for C_n arranged in a circular layout using a standard generator.
- Execution of the group operation (modular addition) between any two elements in C_n.
- Computation of the inverse and the order of a selected element.
- Identification of the elements in the subgroup generated by a given element.

To demonstrate the capabilities of the application, the following screenshots of its core functionalities in action were included. These visual examples illustrate how users can interact with the tool to explore the algebraic structure of cyclic groups dynamically.

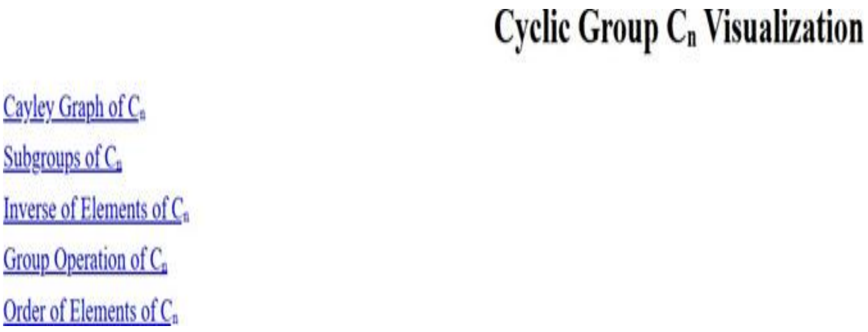

Figure 7: Main page of C_n visualization web tool

Cayley Diagram Visualization

The application generates the Cayley diagram of C_n when n is given.

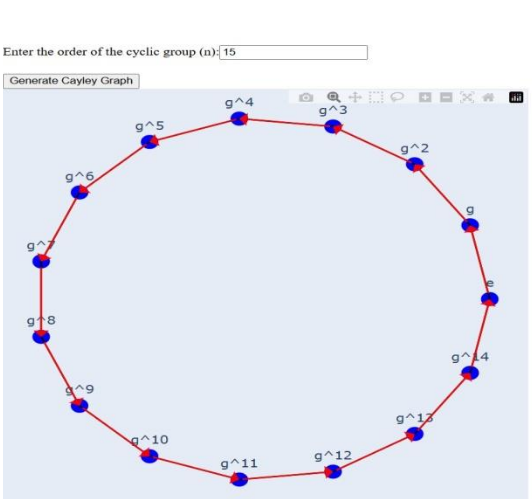

Figure 8: Cayley diagram of C_{15}

Performing Group Operation

Users can select two elements from C_n and compute their sum modulo n, visualizing the result and its position in the cycle.

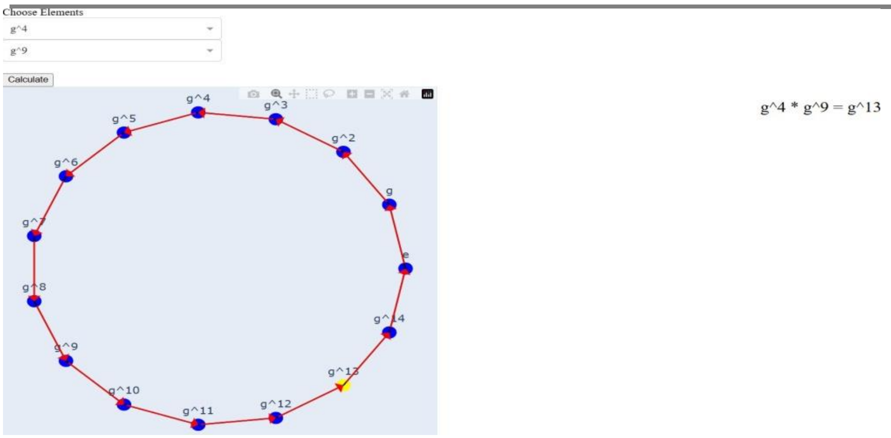

Figure 9: Example of performing $g^4 \cdot g^9 = g^{13}$ in C_{15} using the group operation tool.

Finding the Inverse of an Element

The application allows users to select an element and automatically computes its inverse under modular addition.

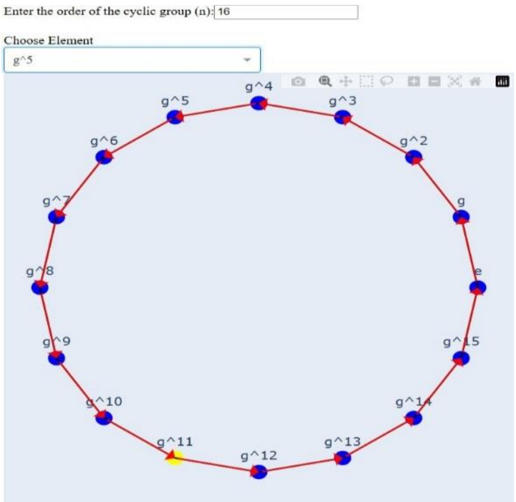

Figure 10: Displaying the inverse of element g^5 in C_{16}, which is g^{11}

Determining the order of Elements

The application allows users to select an element and automatically computes its order.

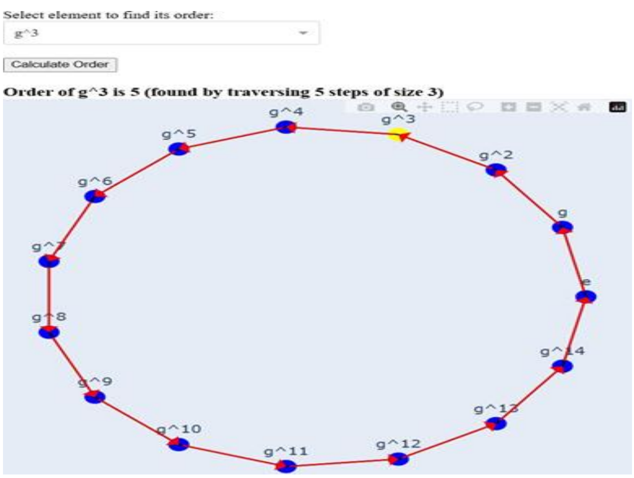

Figure 11: order of g^3 in C_{15} which is 5

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Identifying Subgroup Elements

The application highlights all elements of the subgroup generated by a selected element in C_n.

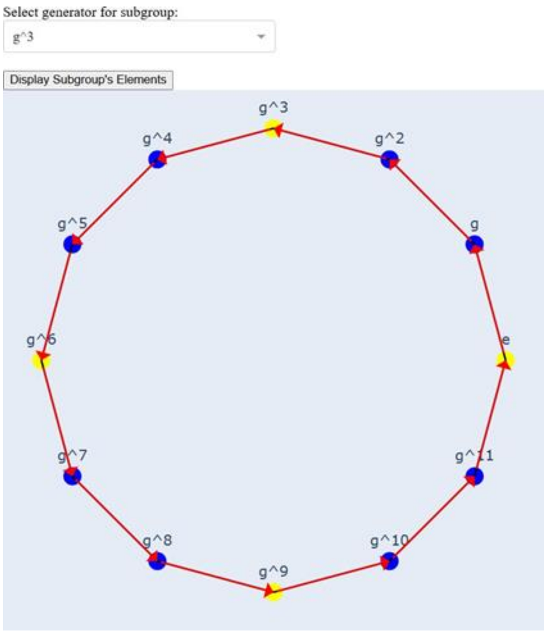

Figure 12: Elements of subgroup generated by element g^3 in C_{12}.

Dihedral group visualization tool

The tool is grounded in fundamental group theoretic concepts and offers a practical platform for experimentation. Users can specify a value for n, generate the corresponding group D_{2n}, and perform a range of operations to analyze the group's structure and subgroups.

The application offers the following real-time functionalities:

- Visualization of the Cayley diagram for D_{2n}, constructed using a chosen set of generators (typically one rotation and one reflection).
- Execution of the group operation between any two elements in D_{2n}.
- Computation of the inverse and the order of a selected group element.
- Identification of all elements in the subgroup generated by a given set of elements.

The following screenshots showcase the interactive functionalities of the web tool related to D_{2n}.

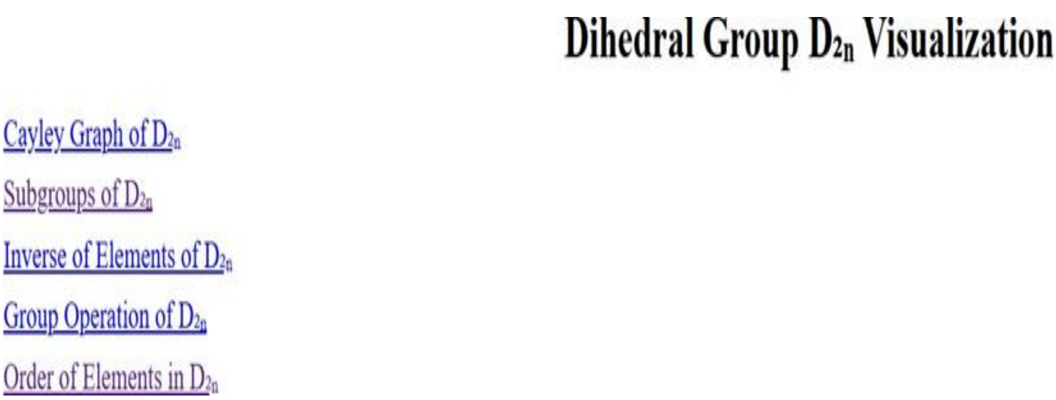

Figure 13: Main page of D_{2n} visualization web tool

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Cayley Diagram Visualization

The application generates the Cayley diagram for D_{2n}, using two generators: a rotation r and a reflection s.

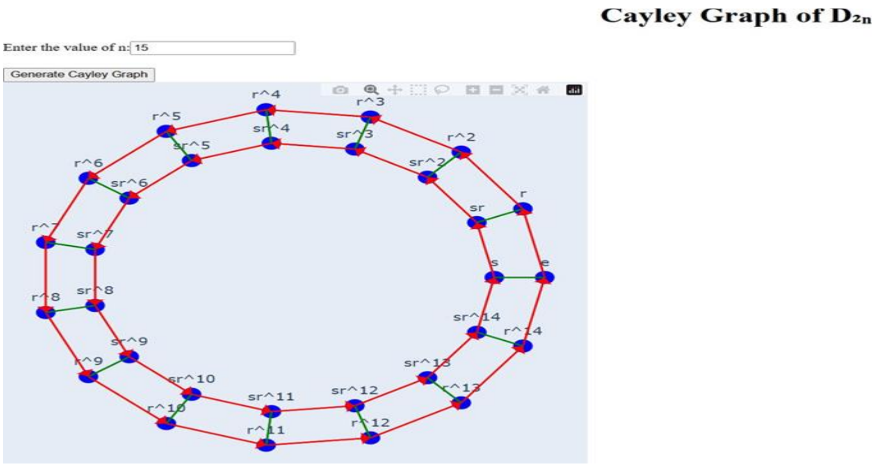

Figure 14: Cayley diagram of D_{30}

Performing Group Operation

Users can choose two elements of D_{2n} and compute their product. The application shows both the operation and the resulting group element.

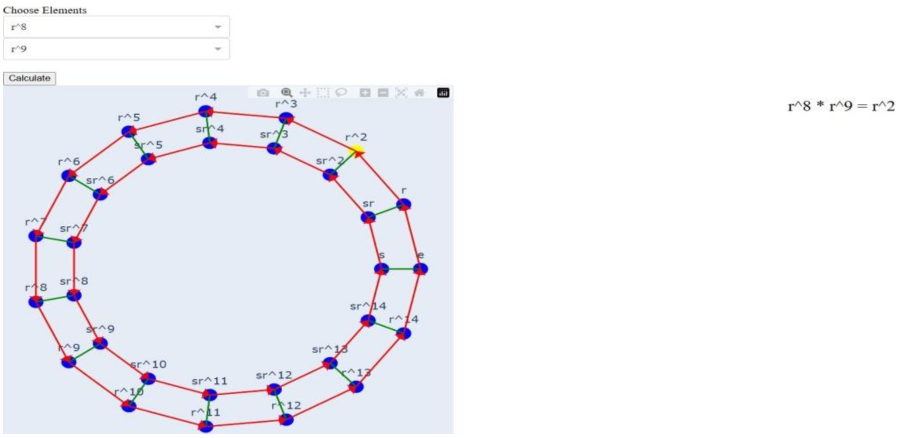

Figure 15: Group operation in D_{30}: computing $r^8 \cdot r^9 = r^2$.

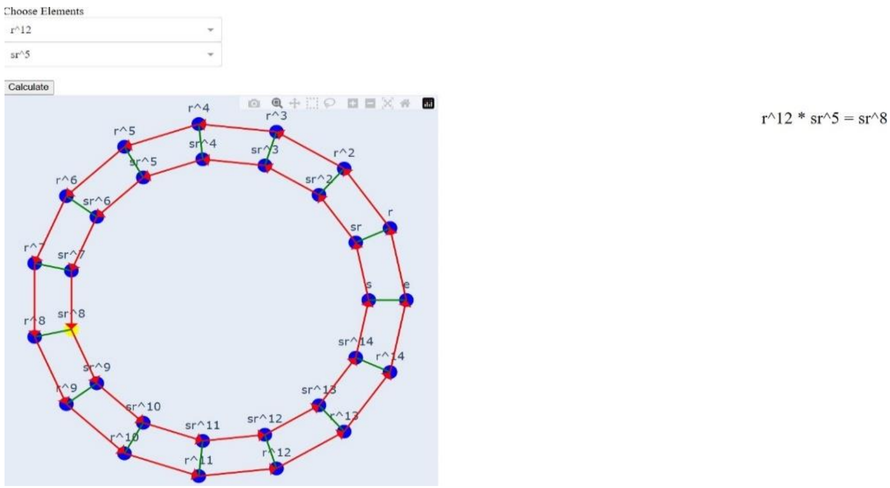

Figure 16: Group operation in D_{30}: computing $r^{12} \cdot sr^5 = sr^8$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

Finding the Inverse of an Element

The tool computes the inverse of any selected element in D_{2n}, whether it is a rotation or a reflection, and displays the result based on the group's multiplication rules.

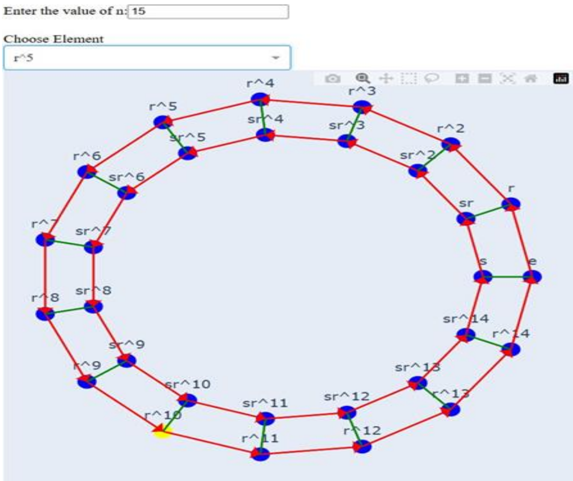

Figure 17: Inverse of r^5 in D_{30} which is r^{10}

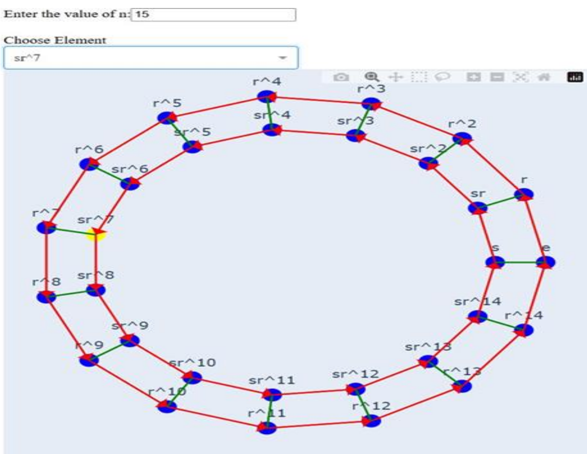

Figure 18: Inverse of sr^7 in D_{30} which is sr^7 itself

Determining Element Orders

Users can select an element and view its order.

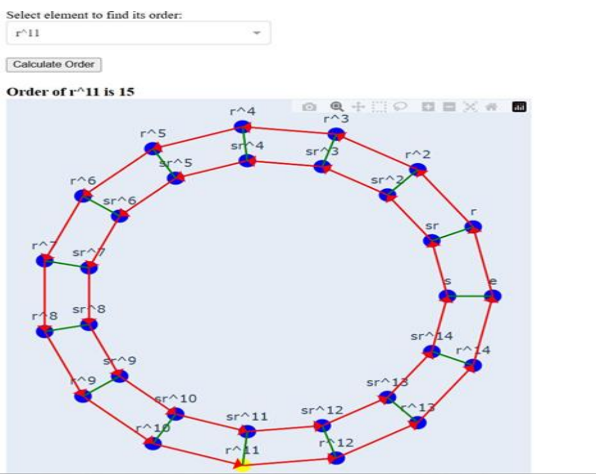

Figure 19: Order of element r^{11} in D_{30} is 15

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

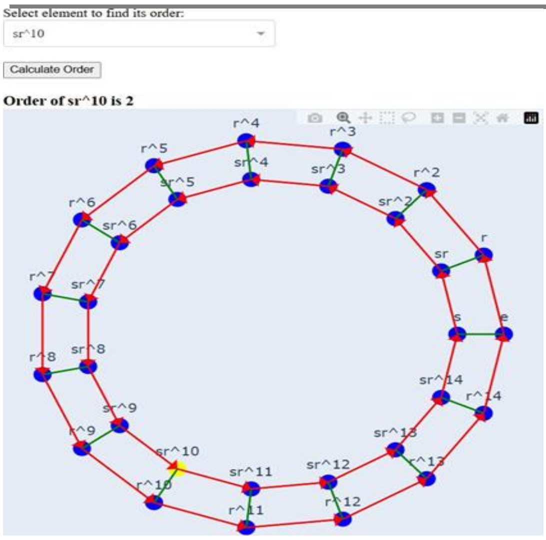

Figure 20: Order of element sr^{10} in D_{30} is 2

Identifying Subgroup Elements

The application can optionally highlight elements of a subgroup generated by the chosen elements.

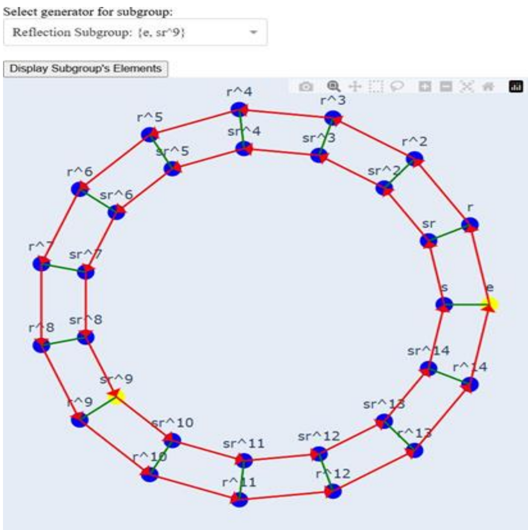

Figure 21: Subgroup generated by e and sr^9 in D_{30}

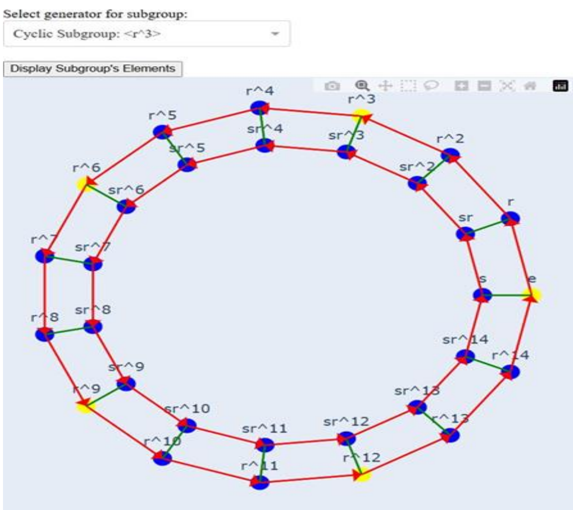

Figure 22: Subgroup generated by r^3 in D_{30}

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

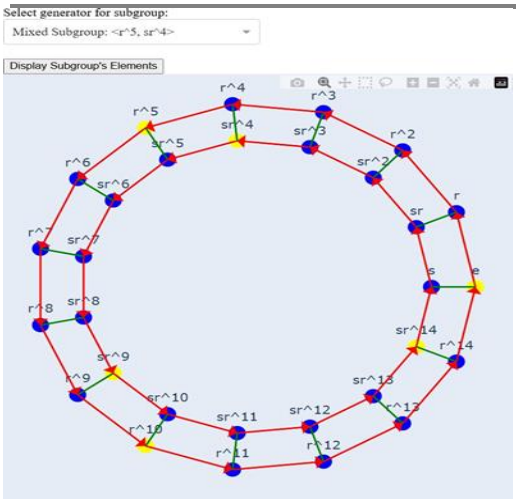

Figure 23: Subgroup generated by r^5 and sr^4 in D_{30}

Symmetric group S_4 visualization tool

The Web interface supports the following functionalities in real time:

- Visualization of the Cayley diagram for S_4 (truncated cube version).
- Performing the group operation between any two selected elements.
- Calculating the inverse and the order of a selected element.
- Identifying subgroup elements generated by given elements. (The generating set has to be chosen by the user from a drop-down list)

The following figures in this section showcase the interactive functionalities of the web tool related to.

Figure 24: Main page of S_4 visualization web tool

Cayley Diagram Visualization

The tool generates the Cayley diagram of S_4 using a given set of generators (0 1) and (1 3 2).

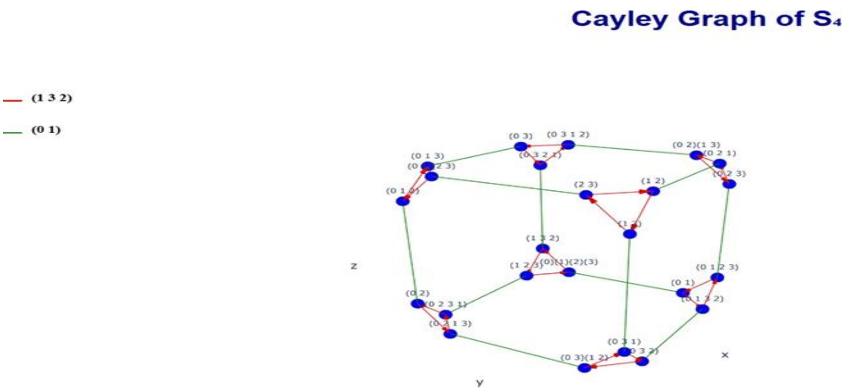

Figure 25: Cayley diagram of S_4 with generators (0 1) and (1 3 2).

Performing Group Operation

Users can select two permutations and compute their composition. The result is shown in standard cycle notation, and the order of operation is clearly indicated (i.e., function composition from right to left).

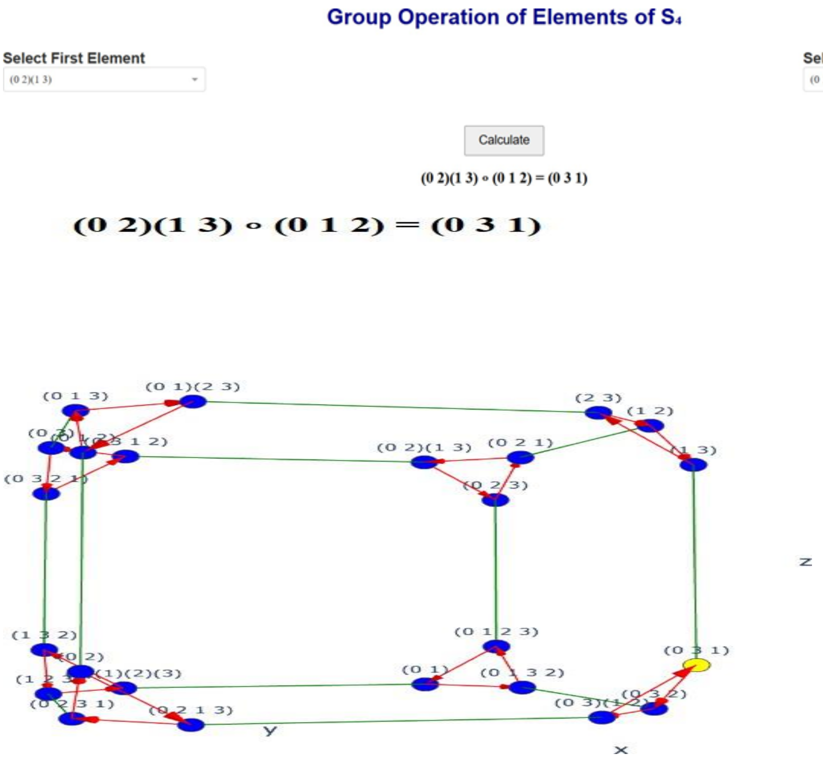

Figure 26: Example of group operation: composing (0 2)(1 3) and (0 1 2) to obtain (0 3 1)

Finding the Inverse of an Element

The application allows users to select a permutation and displays (Highlights) its inverse. Since every permutation is invertible, this feature supports the conceptual understanding of how inverses behave in S_4.

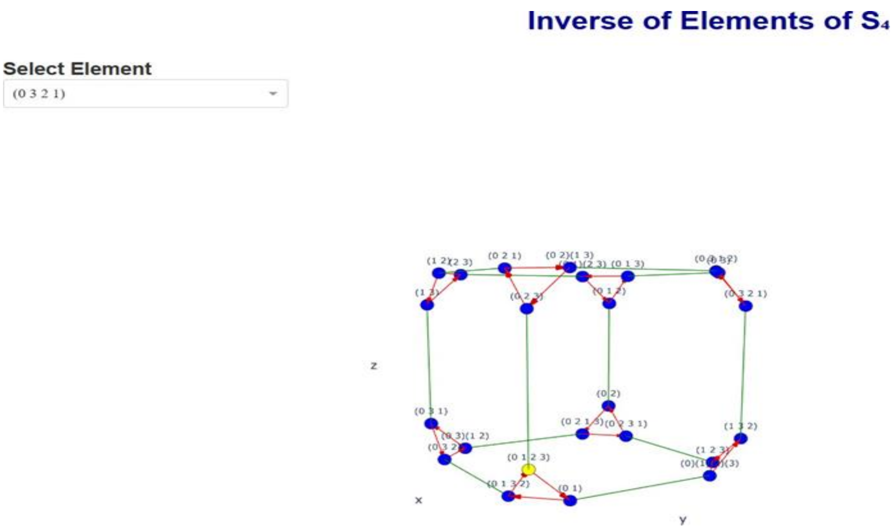

Figure 27: Inverse of permutation (0 3 2 1) is (0 1 2 3).

Determining Element Orders

The tool calculates the order of a permutation which is defined as the smallest positive integer k such that applying the permutation k times results in the identity.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume X Issue VII July 2025

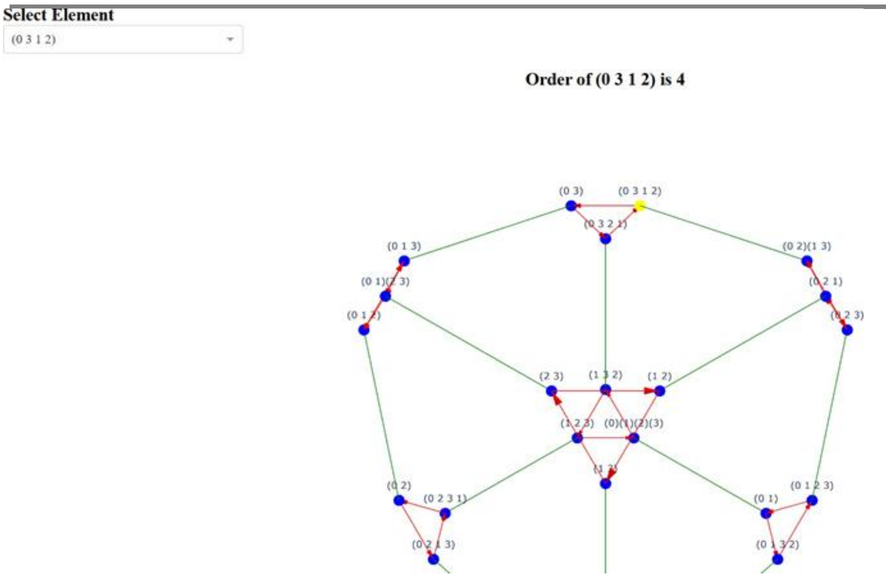

Figure 28: Order of (0 3 1 2) is 4

Identifying Subgroup Elements

The application can highlight the elements of a subgroup generated by a selected element or a pair of elements.

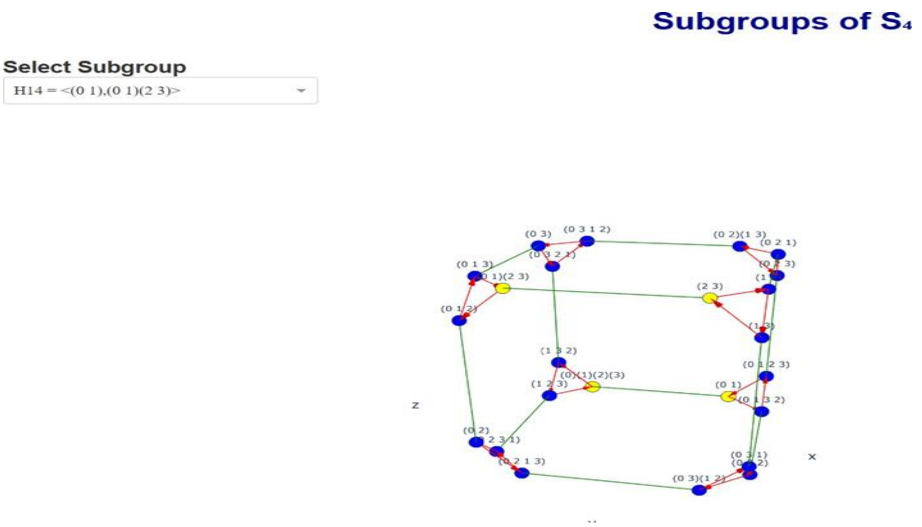

Figure 29: Subgroup generated by (0 1) and (0 1)(2 3)

DISCUSSION

In order to create links between abstract group theory and intuitive graphical representations, this manuscript has methodically examined geometric visualization techniques for basic algebraic structures. Basically, two visualization methods were studied, which are circle representation (for cyclic groups) and Cayley graphs (for cyclic groups, dihedral groups, and symmetric groups). For each group, the basic properties of those groups were studied algebraically and visually, and algorithms were created to find element orders, inverses of elements, perform group operations, and find elements of subgroups using visualizations.

While tools such as GAP (The GAP Group, 2023), SageMath (The Sage Developers, 2023), and Group Explorer (Carter, n.d.) provide valuable features for exploring group theory, they each exhibit limitations when it comes to integrating visual and computational analysis in an interactive and accessible way. GAP (The GAP Group, 2023) and SageMath (The Sage Developers, 2023) offer powerful computational capabilities, including symbolic manipulation of group elements, subgroup construction, and order computations. However, their visual interfaces are minimal or require external packages (e.g., GRAPE or XGAP), and they do not support real-time interaction with Cayley diagrams. Group Explorer, on the other hand, emphasizes visualization and provides an interactive interface to explore structures such as subgroups and cosets, but it lacks the ability to compute or display group properties like element orders, inverses, or perform group operations algorithmically within the visualization.

In contrast, the Python-Flask-based web applications, developed using algorithms created in this study, offers a seamless combination of Cayley diagram visualization and algorithm-driven interaction. Users can compute element orders, inverses, subgroup memberships, and perform group operations directly through graphical interaction with the Cayley diagram, without performing any symbolic calculations manually. The tool supports dynamic visualization of cyclic groups C_n, dihedral groups D_{2n}, and the symmetric group S_4, with scalability for arbitrary positive integers n in C_n and D_{2n}. Although high values of n result in increasingly dense diagrams due to screen space limitations, this is a visual constraint that can be optimized in future versions. This level of interactive analysis is currently unavailable in existing group theory tools, positioning this system as a novel and accessible resource for both research and education.

Table 1: Comparative Feature Analysis of Existing and Implemented Group Theory Tools

Feature/ Tool	GAP / SageMath	Group Explorer	This study (Flask Web App)
Group operations (symbolic)	Yes	Limited (via tables)	Visual & algorithmic
Element order computation	Yes (manual/ scripted)	No	Visual, automatic
Inverse finding	Yes	No	Visual, automatic
Subgroup detection	Yes (manual)	Highlight only	Automatic via diagram
Cayley diagram visualization	Minimal / via packages	Yes	Yes
Interactivity with diagram	No	Click-based, limited analysis	Full interaction & computation
Support for C_n, D_{2n}, and S_4	Symbolically only	Fixed examples	Dynamic, parameterized n
Handles arbitrary n	Yes	No	Yes (visual limitation noted)
Built for education & exploration	Limited	Yes	Yes

This work opens several promising avenues for further investigation across theoretical, computational, and educational domains. The visualization framework developed in this study naturally extends to more complex group structures and innovative applications. The current visualization techniques for symmetric groups up to S_4 suggest potential generalizations to higher order symmetric groups S_5 and S_6, though significant challenges emerge due to their increased complexity (120 and 720 elements, respectively). These challenges include developing higher dimensional representations that maintain interpretability, creating efficient algorithms for pattern recognition in complex graphs, and optimizing computational performance for interactive exploration. Beyond symmetric groups, the methodology could be adapted to other important non-abelian structures such as quaternion groups, matrix groups, and semidirect products, each requiring specialized visualization approaches to capture their unique algebraic properties. Additionally, the multiplicative groups of integers modulo n, $(Z/nZ)^{\times}$, present an interesting test case for visualizing number theoretic structures with applications to cryptography and prime number distribution.

There are tremendous opportunities to improve group visualization using emerging technology. By facilitating interactive theorem demonstration, collaborative learning environments, and three-dimensional manipulation of intricate structures, immersive Virtual Reality (VR) and Augmented Reality (AR) interfaces have the potential to completely change group exploration. Through automated pattern detection in Cayley graphs, predictive modeling of group attributes, and generative systems for producing ideal visual representations, machine learning techniques may offer new avenues for group structure analysis. Another exciting avenue is provided by topological approaches, specifically through the use of metric space embeddings that maintain algebraic interactions, geometric group theory techniques for comprehending large scale patterns, and persistent homology to investigate group structures. These innovative techniques might offer fresh perspectives on the overall organization of complicated groups.

The visualization tools developed in this research have significant potential for educational innovation. A systematic program of curriculum development could produce visualization-based lesson plans, interactive textbooks, and novel assessment tools that leverage graphical representations. Special attention should be given to accessibility enhancements, including the creation of tactile representations for visually impaired students, multimodal interfaces that accommodate different learning styles, and language independent learning materials that make abstract algebra more universally accessible.

CONCLUSION

This study highlights the value of geometric visualization in enhancing the understanding of abstract algebraic structures. By employing circle representations and Cayley graph constructions for cyclic groups, dihedral groups and symmetric group S_4, supported by interactive computational tools, we bridge the gap between formal algebraic theory and intuitive insight. The proposed framework not only contributes to theoretical development but also offers practical relevance in areas such as mathematics education, molecular symmetry, cryptography and computational algebra. Future work may expand these methods into broader domains of pure mathematics, applied computation, and interdisciplinary learning. Our combination of geometric intuition with algebraic thinking provides a promising basis for future work at the interface of computer science, algebra, and visualization.

REFERENCES

1. Arcavi, A. (2003). The role of visual representations in the learning of mathematics, Educational Studies in Mathematics, 52(3), pp. 215-241.
2. Carter, N. (2009). Visual Group Theory. Mathematical Association of America.
3. Carter, N. (n.d.). Group Explorer. https://nathancarter.github.io/group-explorer/GroupExplorer.html
4. Cayley, A. (1878). Desiderata and suggestions: No. 2. The theory of groups: graphical representation. *American Journal of Mathematics*, *1*(2), 174-176.
5. Dummit, D.S., & Foote, R.M. (2004). Abstract Algebra (3rd ed.). Hoboken, John Wiley & Sons.
6. Gauss, C. F. (1966). *Disquisitiones Arithmeticae* (A. A. Clarke, Trans.). Springer-Verlag. (Original work published 1801)
7. Malik, D.S., Mordeson, J.N., & Sen, M.K. (1997). Fundamental of Abstract Algebra. McGraw-Hill.
8. The GAP Group. (2023). GAP - Groups, Algorithms, and Programming (Version 4.12.2) [Computer software]. https://www.gap-system.org
9. The Sage Developers. (2023). SageMath (Version 10.2) [Computer software]. https://www.sagemath.org