Applied Mathematics on Science and Engineering, 3(2):1–10, 2026. DOI: 10.65036/amse321

Applied Mathematics on Science and Engineering

ISSN (Online): 3048-8362

Journal homepage: https://furthersci.org/journal-details.php?journal-id=49

Research Article

A Digit-Based Discrete Dynamical System Generated by a Figure-Changing Transformation

Ashok Bibhishanrao Munde [1*] **and Pramod Machindra Dhakane** [2]

[1]*Department of Mathematics, NKSPT's Arts, Science and Commerce College, Badnapur, Dist:Jalna (M.S.) India.*
[2]*Department of Mathematics, S.B.E.S. College of Science, Chhtrapati Sambhajinagar, (M.S.) India.*
[*]*Corresponding author: mashok.math@gmail.com*

Article Info

Keywords: *Digital root, Finite dynamical system, Periodic orbit, Functional graph, Cycle decomposition.*

MSC 2020: *11A63, 05C20 37B20, 37E99, 68R10.*

Received: *21.06.2026;*
Accepted: *25.07.2026;*
Published: *31.07.2026*

Abstract

Digit-based iterative processes often give rise to rich dynamical behavior despite being defined by simple arithmetic rules. In this paper, we investigate a figure-changing transformation acting on four-digit numbers, where each iteration is obtained by shifting digits and appending the digital root of a partial digit sum. We model this procedure as a discrete dynamical system on a finite state space and prove that every orbit eventually becomes periodic. Computational methods implemented in Python are used to perform an exhaustive analysis of all 10^4 four-digit states. The computation identifies exactly 168 distinct periodic cycles with cycle lengths 1, 2, 6, 8, 18, 24, and 72, and determines the corresponding basin structures. The proposed transformation illustrates how simple digit operations generate highly structured finite dynamical systems, providing a framework for further investigations of arithmetic transformations and their dynamical properties.

1. Introduction

Digit manipulation problems have long attracted interest in number theory and recreational mathematics due to their simplicity and surprisingly rich dynamical behavior. Classical examples include digit sum functions, digital roots, and iterative routines such as Kaprekar's transformation, where repeated application of simple digit rules leads to fixed points or cycles. These processes provide natural examples of discrete dynamical systems defined on finite sets, often exhibiting predictable long-term behavior following nontrivial transient dynamics.

Digit-based transformations are closely connected with modular arithmetic, particularly through the digital root function, which is invariant modulo nine. The mathematical properties of digit sums and digital roots have been extensively studied in classical number theory and recreational mathematics literature, primarily in connection with divisibility tests and congruence relations [1–3]. Such additive digit functions serve as foundational tools in understanding numerical patterns arising from digit operations.

From a dynamical perspective, several authors have investigated iterative digit functions with emphasis on cycle formation, attractors, and convergence behavior. It is well understood that repeated application of deterministic rules on finite sets inevitably results in periodic orbits or fixed points, situating these problems naturally within the framework of discrete dynamical systems [4, 5]. Graph-theoretic representations, where integers are modeled as vertices and transformations as directed edges, have proven especially effective for visualizing cycles and attractor basins in such systems [5, 6].

Among digit-based iterative processes, Kaprekar-type routines have received particular attention. The classical Kaprekar transformation on four-digit numbers and its convergence to constant values or cycles has been rigorously analyzed and extended to higher digit lengths [7–9]. These studies predominantly focus on digit rearrangement mechanisms and their terminal behavior.

In contrast, additive digit functions such as digit sums and digital roots have largely been investigated independently of positional digit dynamics, mainly in relation to modular arithmetic and number theoretic properties [1, 2]. As a result, rearrangement-based transformations and additive digit operations are typically treated as separate mechanisms in the literature. There is comparatively little work that systematically investigates digit transformations combining positional digit shifting with additive digital-root insertion within a unified iterative framework. In particular, the cycle structure, long-term dynamics, and graph-theoretic properties of such hybrid digit transformations have not been explicitly characterized.

Recent years have witnessed renewed interest in digit-based dynamical systems and their computational properties. Devlin and Zeng [10] determined the maximum number of iterations required for convergence to the Kaprekar fixed point among all four-digit numbers in arbitrary bases. Building upon structural aspects of finite mappings, Dor'e *et al.* [11] investigated the decomposition and factorisation of transients in functional graphs. Dahl [12] subsequently carried out an exhaustive computational analysis of Kaprekar maps for several digit lengths, emphasizing attractor basins, entropy measures, and global functional graph structures. Related structural properties of finite discrete-time dynamical systems were further explored by Dor'e *et al.* [13] using rooted-tree representations. More recently, Chen *et al.* [14] completely characterized four-digit Kaprekar dynamics in odd bases by determining terminal cycle lengths and their distribution. Complementary computational techniques for estimating basins of attraction in finite dynamical systems were developed by Datseris and Wagemakers [15]. Collectively, these studies demonstrate that simple digit transformations can exhibit highly organized global dynamics and motivate the investigation of new arithmetic transformations.

Motivated by this gap, the present work introduces and studies a specific figure-changing transformation acting on four-digit numbers that integrates both positional digit shifting and additive digital root operations. The transformation is modeled as a discrete dynamical system on a finite state space, enabling rigorous analysis of its asymptotic behavior. We establish the inevitability of cycle formation and examine the resulting cycle structures through theoretical reasoning supported by exhaustive computational verification using Python. Furthermore, the transformation is interpreted as a functional directed graph, providing clear visualization of cycles and attractor basins. This unified approach bridges additive digit functions, positional digit dynamics, and graph-theoretic methods, thereby addressing the identified gap in the existing literature.

2. Preliminaries and Definitions

In this section, we introduce the basic definitions, notation, and preliminary concepts required for the formal description and analysis of the proposed figure-changing transformation. These notions provide the mathematical foundation for the dynamical and computational results presented in subsequent sections.

2.1. Digit Representation

Throughout this paper, a four-digit number is represented as an ordered tuple of digits

$$N = abcd,$$

where $a,b,c,d \in \{0,1,\dots,9\}$ and a is allowed to be zero unless otherwise stated. This representation is interpreted both as a numerical value and as a digit sequence, depending on context. Such a digit-wise representation is standard in the study of digit-based transformations and allows convenient formulation of iterative rules.

Definition 2.1 (**Digital Root Function**). *The digital root of a nonnegative integer n, denoted by* $\mathrm{dr}(n)$*, is defined as the iterative sum of its decimal digits until a single-digit number is obtained. Formally, the digital root can be expressed as*

$$\mathrm{dr}(n) = \begin{cases} 0, & \text{if } n = 0, \\ 1 + (n-1) \bmod 9, & \text{if } n \geq 1. \end{cases}$$

The digital root function is determined by the residue class modulo nine and maps integers to the finite set $\{0,1,\dots,9\}$. This property plays a central role in ensuring that the transformation introduced in this work is well-defined and closed on the space of four-digit numbers.

Definition 2.2 (**Finite State Space**). *Let*

$$\mathrm{S} = \{0,1,\dots,9\}^4$$

denote the set of all ordered quadruples of digits. The cardinality of S is 10^4, and hence S forms a finite state space.

Any deterministic transformation acting on S naturally induces a discrete-time dynamical system. By standard results from finite dynamical systems, every orbit generated by repeated iteration of such a transformation must eventually enter a periodic cycle.

2.2. Iterative Processes and Orbits

Given a function $T : \mathrm{S} \to \mathrm{S}$, the sequence

$$N, T(N), T^2(N), \dots$$

is referred to as the orbit of N under T. If there exists a positive integer k such that $T^k(N) = N$, then N is said to belong to a cycle of length k. These notions will be used to characterize the long-term behavior of the figure-changing transformation introduced in the next section.

The definitions and notation established here enable a rigorous formulation of the transformation rule and provide the framework for theoretical analysis, computational experimentation, and graphical interpretation presented in the subsequent sections.

3. Figure-Changing Transformation

In this section, we introduce the figure-changing transformation that forms the central object of investigation in this study. The transformation is defined using elementary digit operations and acts deterministically on the finite state space of four-digit numbers introduced in Section 2.

Let $N = abcd \in \mathrm{S}$, where $a,b,c,d \in \{0,1,\ldots,9\}$. Define a mapping $T : \mathrm{S} \to \mathrm{S}$ by

$$T(abcd) = bcd\ \mathrm{dr}(a+b+c),$$

where $\mathrm{dr}(\cdot)$ denotes the digital root function defined earlier. Thus, the transformation is obtained by shifting the first three digits one position to the left and appending the digital root of their sum as the final digit.

Since the output is again a four-digit sequence, the mapping T is well-defined on S and generates a discrete-time dynamical system under repeated iteration.

3.1. Iterative Application

Starting from an initial value $N_0 \in \mathrm{S}$, repeated application of T produces a sequence

$$N_0, N_1 = T(N_0), N_2 = T(N_1), \ldots,$$

which will be referred to as the orbit of N_0 under the transformation. The structure of these orbits and their eventual periodic behavior will be analyzed in subsequent sections.

3.2. Illustrative Example

Consider the four-digit number $N_0 = 4729$. The sum of its first three digits is $4+7+2 = 13$, whose digital root is $\mathrm{dr}(13) = 4$. Hence,

$$T(4729) = 7294.$$

Further iterations are obtained by repeating the same procedure, generating a sequence of four-digit numbers through successive digit shifts and digital root insertions. This example illustrates the iterative construction of the transformation and its iterative nature.

3.3. Remark

The figure-changing transformation combines positional digit shifting with additive digit reduction via the digital root. While the shift operation preserves partial positional structure, the digital root introduces modular arithmetic effects that influence subsequent iterations. This interaction produces nontrivial dynamical behavior and motivates the theoretical and computational analysis developed in the following sections.

4. Cycle Formation and Theoretical Observations

In Section 3, a deterministic figure-changing transformation

$$T : \mathrm{S} \to \mathrm{S}$$

was defined on the finite state space S consisting of all four-digit ordered tuples. Each iteration produces a new element of the same state space by combining digit shifting and digital-root based modification. We now investigate the long-term behavior of repeated iteration of this transformation and establish theoretical properties governing orbit evolution and periodicity.

4.1. Existence of Cycles

The transformation acts on a finite and discrete state space and therefore generates well-defined orbits under iteration. Such systems naturally exhibit eventual periodic behavior, a fundamental property of finite discrete dynamical systems [5].

Theorem 4.1. *For any initial value $N_0 \in \mathrm{S}$, the sequence generated by repeated iteration of the transformation T eventually enters a periodic cycle.*

Proof. The state space S contains exactly 10^4 ordered quadruples of digits. The transformation T defined in Section 3 assigns to each element of S a unique image in the same set. Hence the sequence generated from an initial value N_0,

$$N_0, T(N_0), T^2(N_0), \ldots,$$

remains entirely within S.

Since S is finite, at most 10^4 distinct values can appear in this sequence. Therefore, by the pigeonhole principle, two iterates must eventually coincide. That is, there exist integers $m > n \geq 0$ such that

$$T^m(N_0) = T^n(N_0).$$

Applying the transformation repeatedly to both sides yields

$$T^{m+k}(N_0) = T^{n+k}(N_0)$$

for all integers $k \geq 0$. Letting $p = m - n$, we obtain

$$T^{n+p}(N_0) = T^{n}(N_0),$$

which shows that the sequence becomes periodic after finitely many steps. Thus every orbit eventually enters a cycle. □

4.2. Orbit Decomposition

The above result implies that the state space can be partitioned into disjoint dynamical components. Each component consists of a periodic cycle together with all initial values whose trajectories eventually enter that cycle. Consequently, every element of S belongs to exactly one such component and converges to a unique periodic orbit under repeated application of T.

Definition 4.2. *An element $N \in$ S is called a periodic point of period k if*

$$T^{k}(N) = N$$

for some positive integer k. The smallest such integer k is called the period (or cycle length) of N.

Definition 4.3. *If an element $N \in$ S satisfies $T(N) = N$, then N is called a fixed point of the transformation.*

These definitions enable classification of all elements of the state space according to their long-term dynamical behavior.

4.3. Dependence on Digit Structure

Unlike classical digit-rearrangement processes, the transformation introduced in Section 3 combines positional shift with arithmetic reduction through the digital root of a partial digit sum. Hence the evolution of an orbit depends simultaneously on digit position and modular relationships among digits. This dual dependence produces structured yet nontrivial periodic behavior, where distinct initial values may converge to the same attractor while others form independent cycles.

4.4. Bounds on Cycle Lengths

Because the transformation operates on a finite state space, every cycle must be finite. In particular, no cycle can contain more elements than the total number of states. Thus for any initial value $N \in$ S, there exists a positive integer $k \leq 10^4$ such that

$$T^{k}(N) = N.$$

While this theoretical bound follows immediately from finiteness, the actual cycle lengths observed are typically much smaller. Determining the precise cycle lengths, identifying fixed points, and classifying attractor structures require systematic computational exploration. These aspects will be investigated in the subsequent computational section, where numerical experimentation and graphical representation are employed to analyze the complete dynamical structure of the transformation.

5. Computational and Graph-Theoretic Analysis

To verify the theoretical results established in Section 4, the proposed transformation was implemented in Python. The computational procedure generates the complete orbit corresponding to a given initial value, detects the occurrence of a repeated state, and determines the associated cycle length. The resulting orbit is then interpreted as a functional directed graph, providing a graphical representation of the underlying discrete dynamical system. Consider the initial value

$$N_0 = 4729.$$

Successive application of the transformation defined in Section 3 produces the orbit shown in Table 1.

Iter.	Number	Iter.	Number
N_0	4729	N_{12}	4931
N_1	7294	N_{13}	9317
N_2	2949	N_{14}	3174
N_3	9496	N_{15}	1742
N_4	4964	N_{16}	7423
N_5	9641	N_{17}	4234
N_6	6411	N_{18}	2349
N_7	4112	N_{19}	3499
N_8	1126	N_{20}	4997
N_9	1264	N_{21}	9974
N_{10}	2649	N_{22}	9747
N_{11}	6493	N_{23}	7472

Table 1: Complete orbit of 4729 forming a periodic cycle of length 24

Table 1 shows that the orbit generated from the initial value 4729 consists of twenty-four distinct states before returning to its starting point. Hence the orbit is a periodic cycle of length 24 with no transient phase. This computational result is consistent with Theorem 4.1 and provides explicit numerical verification of the theoretical analysis.

5.1. Verification by Python Computation

The orbit was generated using the Python program given in Appendix A. The program records successive iterates, detects the first repeated state, and computes the corresponding cycle length automatically. The computational summary is presented in Table 2.

Table 2: Summary of the computational results.

Initial value	4729
Total iterations until repetition	24
Cycle starts at	0
Cycle length	24
Repeated state	4729
Transient states	0
Orbit type	Periodic cycle

The computational output confirms that the repeated state coincides with the initial value. Consequently, the orbit begins its periodic behavior immediately, producing a directed cycle of length 24 without any transient states.

5.2. Graphical Representation of the Cycle

The periodic orbit listed in Table 1 is represented by the directed functional digraph shown in Figure 1. Each vertex denotes a four-digit state, while each directed edge represents one application of the transformation.

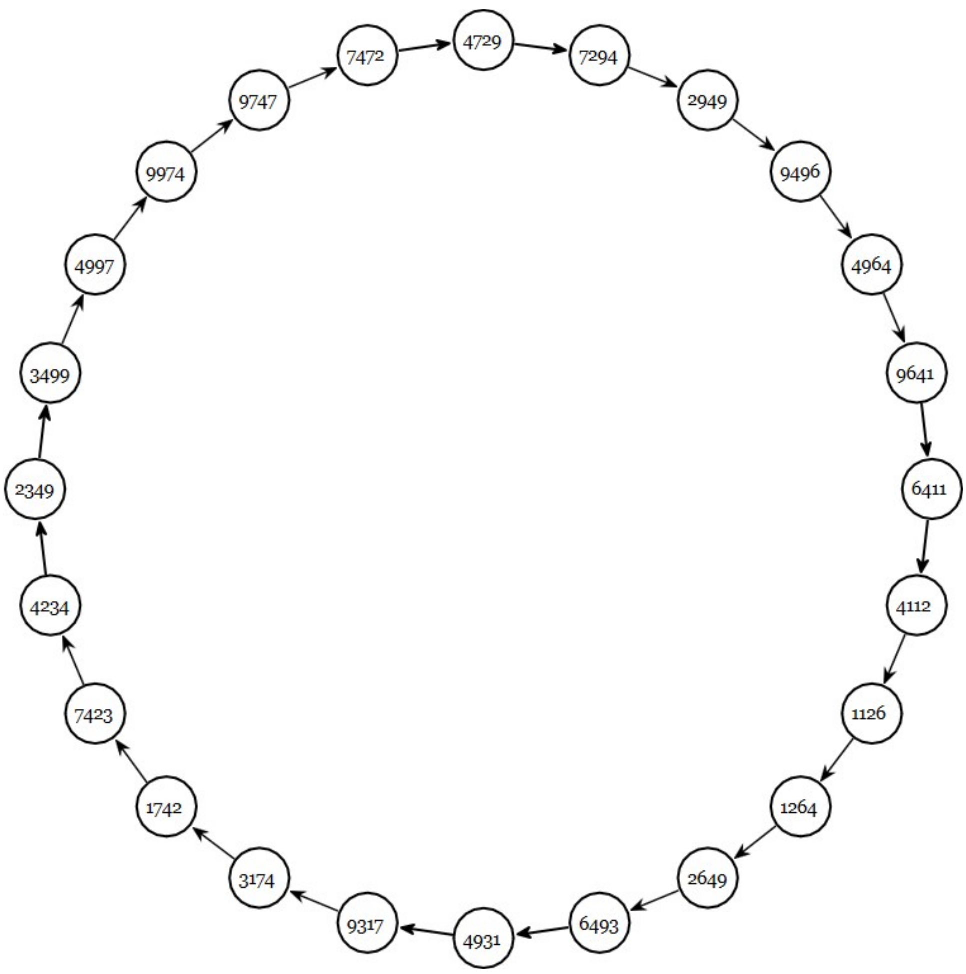

Figure 1: Functional digraph of the periodic orbit generated from the initial value 4729. Each directed edge represents one application of the transformation T. The orbit returns to its initial state after 24 iterations, forming a directed cycle of length 24.

Figure 1 provides a graphical illustration of the representative periodic orbit generated by the proposed transformation. The directed cycle visually confirms the computational results presented in Table 1 and the periodicity established in Theorem 4.1. Although the figure illustrates only one periodic orbit, it serves as a representative example of the dynamical behaviour produced by the transformation. In the next section, the computational investigation is extended to the entire state space in order to determine all periodic components, classify their cycle lengths, and analyse their basin structures.

6. Global Dynamical Structure of the State Space

The numerical experiment presented in Section 5 verifies the behavior of a single orbit generated from the initial value 4729. To understand the dynamics of the transformation completely, it is necessary to investigate the entire finite state space consisting of all four-digit numbers.

Since the transformation defined in Section 3 is deterministic and maps the finite set

$$\mathrm{S} = \{0000, 0001, \ldots, 9999\}$$

into itself, every orbit must eventually enter a periodic cycle. An exhaustive computational analysis was therefore performed by applying the transformation to every element of S. The computation identifies every distinct periodic cycle together with its corresponding basin of attraction.

The complete computation was carried out using Python. The implementation iteratively follows each orbit until a repeated state is encountered. Previously explored trajectories are not recomputed, thereby ensuring that every periodic component is identified exactly once. The Python program used for this global computation is provided in Appendix B.

6.1. Global Cycle Decomposition

The exhaustive computation described above reveals that the transformation partitions the entire state space into a finite collection of mutually disjoint periodic components. Each four-digit number belongs to exactly one orbit and therefore eventually reaches a unique periodic cycle. Consequently, the state space admits a complete decomposition into basins of attraction, each associated with a single periodic cycle.

The computational analysis establishes the following result.

Theorem 6.1. *Let $T : \mathrm{S} \to \mathrm{S}$ be the figure-changing transformation defined in Section 3, where $|\mathrm{S}| = 10^4$. Then the directed functional graph associated with T decomposes into exactly* 168 *pairwise disjoint periodic cycles. Furthermore, every periodic orbit has one of the following lengths:*

$$1,\ 2,\ 6,\ 8,\ 18,\ 24,\ \textit{or}\ 72.$$

No other cycle lengths occur.

Proof: *(By exhaustive computation).* Since the state space is finite and the transformation is deterministic, every orbit eventually becomes periodic by Theorem 4.1. Therefore, every connected component of the functional graph contains exactly one directed cycle together with its associated basin of attraction.

To determine all periodic components, every element of the state space was examined computationally. For each initial value, the transformation was iterated until a previously visited state appeared. Whenever a new periodic cycle was detected, it was recorded, while previously classified states were excluded from further computation.

The exhaustive search over all 10^4 states identified exactly 168 distinct periodic cycles. Moreover, every observed cycle has length belonging to the set

$$\{1, 2, 6, 8, 18, 24, 72\},$$

and no additional cycle lengths were found. Since the computation covers the entire finite state space, the classification is complete. □

6.2. Distribution of Periodic Cycle Lengths

The exhaustive computation described in the previous subsection not only identifies every periodic cycle but also determines the frequency of occurrence of each admissible cycle length. The resulting distribution is summarized in Table 3.

Table 3: Distribution of periodic cycle lengths in the complete state space.

Cycle length	Number of cycles	Percentage (%)
1	2	1.19
2	4	2.38
6	3	1.79
8	9	5.36
18	3	1.79
24	87	51.79
72	60	35.71
Total	168	100.00

Table 3 shows that the transformation admits only seven distinct periodic lengths throughout the entire state space. Among the 168 periodic cycles, cycles of lengths 24 and 72 dominate the global dynamics, accounting for more than 87% of all periodic components. In contrast, fixed points and shorter cycles occur only rarely.

The exhaustive computational analysis shows that the transformation admits only the seven cycle lengths listed in Table 3. No additional periodic length occurs within the complete state space of 10^4 four-digit numbers.

This classification demonstrates that the proposed transformation does not generate periodic orbits of arbitrary lengths. Instead, the combined action of positional digit shifting and digital-root insertion constrains the dynamics to a small set of admissible periods. Consequently, the global dynamics exhibit a highly structured periodic organization despite the simple definition of the transformation.

The predominance of cycles of lengths 24 and 72 further suggests that these periods arise naturally from the arithmetic constraints imposed by the transformation. Conversely, cycles of lengths 1, 2, 6, 8, and 18 represent exceptional configurations occupying only a small portion of the periodic structure.

6.3. Basin Structure of the State Space

The results presented in the previous subsection describe the distribution of the *periodic cycles*. However, the number of cycles alone does not fully characterize the global dynamics of the transformation. An equally important aspect is the distribution of the initial states among these periodic components.

For a periodic cycle C, the *basin of attraction* of C is defined as the set of all initial four-digit numbers whose forward orbits eventually enter C. Since every orbit becomes periodic by Theorem 4.1, the state space is partitioned into disjoint basins of attraction, each associated with exactly one periodic cycle.

Using the exhaustive computational procedure described in Appendix B, the basin size corresponding to every periodic cycle was determined. Table 4 summarizes the total number of states belonging to the basins of cycles having the same period.

Table 4: Distribution of the state space according to basin size.

Cycle Length	Number of Cycles	States in Basins	State Space (%)
1	2	16	0.16
2	4	8	0.08
6	3	28	0.28
8	9	118	1.18
18	3	78	0.78
24	87	3170	31.70
72	60	6582	65.82
Total	168	10000	100.00

Table 4 reveals a marked contrast between the distribution of periodic cycles and the distribution of the state space. Although cycles of length 24 are the most numerous, accounting for 87 of the 168 periodic cycles, they attract only 31.70% of the entire state space. In contrast, the 60 cycles of length 72 attract 6582 initial states, representing 65.82% of all four-digit numbers.

The remaining cycle lengths collectively account for less than 3% of the state space. Consequently, the long-term behavior of a randomly selected initial value is overwhelmingly governed by the periodic cycles of lengths 24 and 72.

This basin structure demonstrates that the global dynamics are highly non-uniform. While several periodic lengths occur, the overwhelming majority of initial states are attracted to relatively few classes of periodic behavior. The coexistence of numerous periodic components together with highly unequal basin sizes is a characteristic feature of the proposed figure-changing transformation.

7. Conclusion

A new figure-changing transformation based on positional digit shifting and digital-root insertion has been introduced and investigated on the finite state space of four-digit numbers. Theoretical analysis establishes that every orbit generated by the transformation is eventually periodic. An exhaustive computational investigation of the entire state space consisting of 10^4 four-digit numbers was carried out to determine the global dynamical behaviour of the transformation.

The computation shows that the state space decomposes into exactly 168 distinct periodic components with admissible cycle lengths

$$\{1,2,6,8,18,24,72\},$$

the maximum observed period being 72. The accompanying basin analysis provides a quantitative description of the domains of attraction associated with these periodic components, thereby completing the global dynamical characterization of the transformation.

The results demonstrate that a simple digit transformation governed by the digital-root operation can generate highly structured finite dynamical behaviour with rich arithmetic and graph-theoretic properties.

Future work may consider analogous transformations on higher-digit numbers, alternative digit operations, and related arithmetic functions in order to obtain deeper theoretical characterizations of their periodic structures and associated finite dynamical systems.

Article Information

Author Contributions: A.B.M. - Conceptualization, Methodology, Data curation, Formal analysis, Supervision; P.M.D. - Writing – original draft, Writing – review & editing.

Funding / Financial Support: The authors received no external funding.

Conflict of Interest: Authors have declared that no competing interests exist.

Ethical Approval: Not Applicable.

Informed Consent: Not Applicable.

Data Availability Statement: Data included in research paper materials.

Clinical Trial Registration: Not Applicable.

Reporting Guidelines Statement: Not Applicable.

Disclaimer (Artificial Intelligence): The author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.), and text-to-image generators have been used during writing or editing of manuscripts.

References

[1] I. Niven and H. S. Zuckerman. *An Introduction to the Theory of Numbers*. John Wiley & Sons, New York, 1960.

[2] R. K. Guy. *Unsolved Problems in Number Theory*. Springer, New York, 3^{rd} edition, 2004.

[3] G. Weiss. Digit sums and digital roots in number theory. *Mathematics Magazine*, 91:101–112, 2018.

[4] R. L. Devaney. *An Introduction to Chaotic Dynamical Systems*. Addison-Wesley, Redwood City, 2^{nd} edition, 1989.

[5] S. Elaydi. *Discrete Chaos: With Applications in Science and Engineering*. Chapman & Hall/CRC, Boca Raton, 2005.

[6] F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

[7] D. R. Kaprekar. A new class of numbers. *Scripta Mathematica*, 21:303–306, 1955.

[8] T. Sreenivasan. Kaprekar's constant and its generalizations. *Mathematics Magazine*, 62:303–306, 1989.

[9] J. S. Madachy. *Mathematics on Vacation*. Charles Scribner's Sons, New York, 1979.

[10] P. Devlin and T. Zeng. Maximum distances in the four-digit Kaprekar process. *arXiv preprint arXiv:2010.11756*, 2020.

[11] F. Dor´e, E. Formenti, A. E. Porreca, and S. Riva. Decomposition and factorisation of transients in functional graphs. *Theoretical Computer Science*, 1008:114668, 2024.

[12] C. D. Dahl. Coarse-grained drift fields and attractor-basin entropy in Kaprekar's routine. *Entropy*, 28(1):92, 2026.

[13] F. Dor´e, K. Perrot, A. E. Porreca, S. Riva, and M. Rolland. Infinite trees for division and roots over finite discrete-time dynamical systems. *Natural Computing*, 2026.

[14] E. Chen, K. Ono, R. E. Schwartz, and D. S. Thakur. Four-digit Kaprekar dynamics in odd bases. *arXiv preprint arXiv:2606.20439*, 2026.

[15] G. Datseris and A. Wagemakers. Effortless estimation of basins of attraction. *Chaos*, 31:023115, 2021.

Appendix :

A. Python Code for Orbit Generation and Cycle Detection

The following Python program generates the orbit of a four-digit number under the transformation defined in Section 3, detects cycle formation, and determines the cycle length.

```
def digital_root(n):
 if n == 0:
 return 0
 return 1 + (n - 1) % 9

def transform(num):
 s = str(num).zfill(4)
 d1, d2, d3, d4 = map(int, s)

 dr = digital_root(d1 + d2 + d3)
 new_num = int(f"{d2}{d3}{d4}{dr}")
 return new_num, dr

def generate_orbit(initial):
 visited = {}
 orbit = [initial]
 dr_list = ["--"]

 current = initial
 step = 0

 while True:
 if current in visited:
 start = visited[current]
 cycle_length = step - start
 break

 visited[current] = step

 next_num, dr = transform(current)
 orbit.append(next_num)
 dr_list.append(dr)

 current = next_num
```

```
 step += 1

 return orbit, dr_list, start, cycle_length

def print_orbit_table(initial):
 orbit, dr_list, start, length = generate_orbit(initial)

 print("Orbit Table:\n")
 print("Iter Number DR")
 print("----------------------")

 for i in range(len(orbit)):
 print(i, orbit[i], dr_list[i])

 print("\nCycle starts at:", start)
 print("Cycle length =", length)

# Run example
print_orbit_table(4729)
```

Listing 1: Python program for orbit generation and cycle detection

Representative Program Output

The program reports

```
Cycle starts at: 0
Cycle length = 24
```

The complete orbit generated by the program is reported in Table 1.

B. Python Program for Global Cycle Computation

The following Python program performs an exhaustive exploration of the complete state space consisting of all four-digit numbers. It identifies every distinct periodic cycle, determines the distribution of cycle lengths, and reports representative cycles.

```
from collections import Counter

# ----------------------------------------------------
# Digital Root Function
# ----------------------------------------------------
def digital_root(n):
 if n == 0:
 return 0
 return 1 + (n - 1) % 9


# ----------------------------------------------------
# Figure-Changing Transformation
# ----------------------------------------------------
def transform(num):

 s = str(num).zfill(4)

 a, b, c, d = map(int, s)

 dr = digital_root(a + b + c)

 return int(f"{b}{c}{d}{dr}")


# ----------------------------------------------------
# Global Cycle Computation
# ----------------------------------------------------
visited_global = set()

cycles = []

for start in range(10000):

 if start in visited_global:
 continue

 local_index = {}

 orbit = []

 current = start
```

```
 while current not in local_index and current not in visited_global:

 local_index[current] = len(orbit)

 orbit.append(current)

 current = transform(current)

 # New cycle found
 if current in local_index:

 cycle = orbit[local_index[current]:]

 cycles.append(cycle)

 visited_global.update(orbit)


# ----------------------------------------------------
# Summary
# ----------------------------------------------------
print()

print("=" * 60)
print("GLOBAL ANALYSIS OF THE TRANSFORMATION")
print("=" * 60)

print()
print("Total states :", 10000)
print("Distinct cycles discovered :", len(cycles))
print()

distribution = Counter(len(c) for c in cycles)

print("Cycle Length Distribution")
print("-------------------------")

for L in sorted(distribution):
 print(f"Length {L:3d} : {distribution[L]} cycles")

print()

largest = max(len(c) for c in cycles)
smallest = min(len(c) for c in cycles)

print("Maximum cycle length :", largest)
print("Minimum cycle length :", smallest)

print()

print("=" * 60)
print("Sample cycles")
print("=" * 60)
print()

for i, cycle in enumerate(cycles[:10], start=1):

 print(f"Cycle {i}")

 print(cycle)

 print()
```

Listing 2: Python program for global cycle computation