Article

Not peer-reviewed version

Collatz Conjecture, Cycles, the 4n – 2 Series, and Positions Within the Series

Patricio Raúl Vicente *

Posted Date: 7 March 2025

doi: 10.20944/preprints202503.0499.v1

Keywords: Collatz conjecture cycles; cycle series positions; position algorithm metrics

Article

Collatz Conjecture, Cycles, the (4n - 2) Series, and Positions within the Series

Patricio Raúl Vicente

v_pato@hotmail.com

Abstract: We demonstrate that the Collatz conjecture generates cycles that consistently converge to the series defined by $4n - 2$, ultimately terminating at the number 2. We propose an innovative algorithm that predicts the positions within this series during each cycle. Through a rigorously defined metric, we establish that these positions invariably converge to position 1, corresponding to the number 2 in the Collatz sequence ($2 \to 1 \to 4 \to 2$). This approach not only confirms the validity of the conjecture for all positive integers but also provides a novel framework for understanding its cyclic behavior.

Keywords: Collatz conjecture cycles; cycle series positions; position algorithm metrics

1. Introduction

Formulated by Lothar Collatz in 1937 [1], the Collatz conjecture asserts that, for any positive integer n, the iterative application of the function:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even,} \\ 3n+1 & \text{if } n \text{ is odd,} \end{cases}$$

eventually leads to the number 1, entering the cycle $1 \to 4 \to 2 \to 1$. Despite extensive study [2, 3], it remains one of the most intriguing unsolved problems in number theory.

In this paper, we approach the conjecture from an original perspective: we show that the generated sequences cycle through numbers in the series $4n - 2$, converging to 2. We define an algorithm to track positional transitions within this series and employ a metric to prove convergence to position 1, thus demonstrating the conjecture. Our methodology combines analytical rigor and computational support, offering new insights into this classic problem.

2. Preliminary Results

We establish the fundamental properties underpinning our analysis.

2.1. Properties of Odd Numbers

Lemma 1. *The product of two odd numbers is odd.*

Proof. Let $a = 2k + 1$ and $b = 2m + 1$ be two odd integers, with $k, m \in \mathbb{Z}$. Their product is:

$$a \cdot b = (2k+1)(2m+1) = 4km + 2k + 2m + 1 = 2(2km + k + m) + 1,$$

an odd number. In particular, since $3 = 2 \cdot 1 + 1$ is odd, if n is odd, then $3n$ is also odd. □

Lemma 2. *Any odd number plus 1 is even.*

Proof. Let $a = 2k + 1$ be an odd number, with $k \in \mathbb{Z}$. Then:

$$a + 1 = 2k + 1 + 1 = 2(k + 1),$$

which is even. Therefore, if n is odd, $3n + 1$ is even. □

2.2. *Partition of Even Numbers*

We define the set of positive even integers as $E = \{2, 4, 6, 8, \ldots\}$. We divide it into:

- Subset A: $A = \{4n - 2 \mid n = 1, 2, 3, \ldots\} = \{2, 6, 10, 14, 18, \ldots\}$,
- Subset B: $B = \{4m \mid m = 1, 2, 3, \ldots\} = \{4, 8, 12, 16, 20, \ldots\}$.

Each subset represents 50% of E, alternating in the sequence of even numbers.

2.3. *Relationship Between Subsets*

Lemma 3. *Every number in B can be expressed as $2^k \cdot a$, where $a \in A$ and $k \geq 1$.*

Proof. Let $b = 4m \in B$. Then $b = 2^2 \cdot m$. If $m = 2^l \cdot q$ with q odd, we have:

$$b = 2^{2+l} \cdot q.$$

Dividing by 2 $2 + l$ times yields q. Then $3q + 1$ is even, and after further divisions, we reach a number in A. Example: $96 = 6 \cdot 2^4, 6 \in A$. □

3. Cyclic Behavior and the $4n - 2$ Series

3.1. *Transition to Even Numbers*

By Lemmas 1 and 2, $3n + 1$ transforms any odd n into an even number, initiating divisions by 2.

3.2. *Convergence to the $4n - 2$ Series*

An even number in B, such as $2^k \cdot a$, reduces to $a \in A$ after divisions. From $a \in A$, a division leads to an odd number, restarting the cycle. All sequences terminate at $2 \in A$.

4. Positional Analysis within the $4n - 2$ Series

We define the position of a number $s \in A$ as:

$$\text{Position} = \frac{s + 2}{4},$$

where $s = 4n - 2$. For $s = 2$, the position is 1. Our goal is to demonstrate that Collatz cycles converge to this position.

5. Positional Transition Algorithm

We propose an algorithm to predict the next position after each cycle in the $4n - 2$ series:

- Step 1: Multiply the current position by 1.5. If the result is an integer, this is the new position.
- Step 2: If it is not an integer, add 0.5 and divide by 2, repeating until an integer is obtained.

Mathematically, the algorithm is expressed as an iterative function $g(p_k)$, where p_k is the position at step k:

$$g(p_k) = \begin{cases} \frac{3}{2}p_k & \text{if } \frac{3}{2}p_k \in \mathbb{Z}, \\ \text{nearest integer after iterating } \frac{x+0.5}{2} \text{ from } x_0 = \frac{3}{2}p_k & \text{if } \frac{3}{2}p_k \notin \mathbb{Z}, \end{cases}$$

where the iteration $x_{i+1} = \frac{x_i + 0.5}{2}$ is applied until $x_m \in \mathbb{Z}$, and then $g(p_k) = x_m$.

6. Demonstration of the Convergence of the Positional Algorithm

6.1. Definition of the Convergence Metric

To prove that the positional algorithm converges to 1 from any initial position $p_0 \in \mathbb{Z}_{\geq 1}$, we define the metric, building on computational approaches to iterative sequences [4, 5]:

$$h(p_k) = p_k \cdot 2^{-k},$$

where p_k is the k-th position in the sequence generated by g, and k is the step index (starting at $k = 0$ for p_0). Convergence to 1 is verified if $h(p_k)$ strictly decreases at each step and the sequence terminates at $p_t = 1$.

6.2. Fundamental Property of Decrease

We demonstrate that $h(p_{k+1}) < h(p_k)$ for all $k \geq 0$.

- $h(p_{k+1}) = p_{k+1} \cdot 2^{-(k+1)} = \frac{p_{k+1}}{2} \cdot 2^{-k}$,
- We want $h(p_{k+1}) < h(p_k) = p_k \cdot 2^{-k}$, which implies:

$$\frac{p_{k+1}}{2} \cdot 2^{-k} < p_k \cdot 2^{-k} \quad \Rightarrow \quad p_{k+1} < 2p_k.$$

Let us analyze this condition considering the rules of g.

6.2.1. Case 1: p_k even ($p_k \cdot \frac{3}{2}$ is an integer)

If p_k is even:

- $p_{k+1} = g(p_k) = \frac{3}{2}p_k$,
- We verify: $\frac{3}{2}p_k < 2p_k$, since $\frac{3}{2} < 2$, which always holds,
- Then:
$$h(p_{k+1}) = \frac{3}{2}p_k \cdot 2^{-(k+1)} = \frac{3}{2}p_k \cdot \frac{1}{2} \cdot 2^{-k} = \frac{3}{4}p_k \cdot 2^{-k} = \frac{3}{4}h(p_k) < h(p_k),$$
- This strictly satisfies the decrease condition.

6.2.2. Case 2: p_k odd ($p_k \cdot \frac{3}{2}$ is not an integer)

If p_k is odd:

- $x_0 = \frac{3}{2}p_k = 1.5p_k$ is not an integer,
- We iterate $x_{i+1} = \frac{x_i + 0.5}{2}$ until x_m is an integer, and $p_{k+1} = x_m$,
- General expression:
$$x_m = \frac{1.5p_k + 0.5m}{2^m},$$
where m is the minimum number of iterations required for $x_m \in \mathbb{Z}$.

General proof of $p_{k+1} < 2p_k$: For x_m to be an integer, $1.5p_k + 0.5m = \frac{3p_k + m}{2}$ must be divisible by 2^m. Since p_k is odd, $3p_k$ is also odd, and m determines divisibility. Let $v_2(p_k) = 0$ (the 2-adic valuation of p_k), then: - $v_2(3p_k) = 0$, - $x_m = \frac{3p_k + m}{2^m}$ must be an integer, implying $3p_k + m = 2^m q$ for some $q \in \mathbb{Z}$, - m is the smallest integer such that $3p_k + m$ is divisible by 2^m.

Consider the relative growth: - Initially, $x_0 = 1.5p_k < 2p_k$, - Each iteration reduces the value: - $x_1 = \frac{1.5p_k + 0.5}{2} = 0.75p_k + 0.25$, - $x_2 = \frac{x_1 + 0.5}{2} = \frac{0.75p_k + 0.75}{2} = 0.375p_k + 0.375$, - In general, $x_m < 2p_k$ because: - $x_0 < 2p_k$, - $x_{i+1} = \frac{x_i + 0.5}{2} < \frac{2p_k}{2} + 0.25 = p_k + 0.25 < 2p_k$ (since $p_k \geq 1$), - Thus, for any m where x_m is an integer, $x_m < 2p_k$.

Verification with examples: - $p_k = 3$: - $x_0 = 4.5, x_1 = 2.5, x_2 = 1.5, x_3 = 1$ ($m = 3$), - $p_{k+1} = 1 < 6$, - $p_k = 7$: - $x_0 = 10.5, x_1 = 5.5, x_2 = 3$ ($m = 2$), - $p_{k+1} = 3 < 14$.

Therefore, $p_{k+1} < 2p_k$ holds in all cases, and:

$$h(p_{k+1}) = p_{k+1} \cdot 2^{-(k+1)} < 2p_k \cdot \frac{1}{2} \cdot 2^{-k} = p_k \cdot 2^{-k} = h(p_k).$$

6.3. Formalized Examples

6.3.1. Example 1: $p_0 = 5$

- Sequence: $5 \to 7.5 \to 4 \to 6 \to 9 \to 13.5 \to 7 \to 10.5 \to 5.5 \to 3 \to 4.5 \to 2.5 \to 1.5 \to 1$,
- Integers (p_k): $5, 4, 6, 9, 7, 3, 1$,
- Metric $h(p_k)$: $5 \cdot 2^0 = 5, 4 \cdot 2^{-1} = 2, 6 \cdot 2^{-2} = 1.5, 9 \cdot 2^{-3} = 1.125, 7 \cdot 2^{-4} = 0.4375, 3 \cdot 2^{-5} = 0.09375, 1 \cdot 2^{-6} = 0.015625$,
- Verification: Strictly decreases, terminates at $p_6 = 1$.

6.3.2. Example 2: $p_0 = 6$

- Sequence: $6 \to 9 \to 13.5 \to 7 \to 10.5 \to 5.5 \to 3 \to 4.5 \to 2.5 \to 1.5 \to 1$,
- Integers (p_k): $6, 9, 7, 3, 1$,
- Metric $h(p_k)$: $6 \cdot 2^0 = 6, 9 \cdot 2^{-1} = 4.5, 7 \cdot 2^{-2} = 1.75, 3 \cdot 2^{-3} = 0.375, 1 \cdot 2^{-4} = 0.0625$,
- Verification: Strictly decreases, terminates at $p_4 = 1$.

6.3.3. Example 3: $p_0 = 13$

- Sequence: $13 \to 19.5 \to 10 \to 15 \to 22.5 \to 11.5 \to 6 \to 9 \to 13.5 \to 7 \to 10.5 \to 5.5 \to 3 \to 4.5 \to 2.5 \to 1.5 \to 1$,
- Integers (p_k): $13, 10, 15, 6, 9, 7, 3, 1$,
- Metric $h(p_k)$: $13 \cdot 2^0 = 13$, $10 \cdot 2^{-1} = 5$, $15 \cdot 2^{-2} = 3.75$, $6 \cdot 2^{-3} = 0.75$, $9 \cdot 2^{-4} = 0.5625$, $7 \cdot 2^{-5} = 0.21875, 3 \cdot 2^{-6} = 0.046875, 1 \cdot 2^{-7} = 0.0078125$,
- Verification: Strictly decreases, terminates at $p_7 = 1$.

6.3.4. Example 4: $p_0 = 27$

- Sequence: $27 \to 40.5 \to 20.5 \to 10.5 \to 5.5 \to 3 \to 4.5 \to 2.5 \to 1.5 \to 1$,
- Integers (p_k): $27, 3, 1$,
- Metric $h(p_k)$: $27 \cdot 2^0 = 27, 3 \cdot 2^{-1} = 1.5, 1 \cdot 2^{-2} = 0.25$,
- Verification: Strictly decreases, terminates at $p_2 = 1$.

6.4. Formal Proof of Convergence

Theorem 1. *For every $p_0 \in \mathbb{Z}_{\geq 1}$, the sequence $\{p_k\}$ generated by g terminates at $p_t = 1$.*

Proof. 1. Definition of the metric: $h(p_k) = p_k \cdot 2^{-k}$, with $p_0 = n$.

2. Strict decrease:
   - $h(p_{k+1}) = p_{k+1} \cdot 2^{-(k+1)} = \frac{p_{k+1}}{2} \cdot 2^{-k}$,
   - For $h(p_{k+1}) < h(p_k)$, we require $p_{k+1} < 2p_k$.
   - Case 1 (p_k even):
     - $p_{k+1} = \frac{3}{2}p_k$,
     - $\frac{3}{2}p_k < 2p_k$ (since $\frac{3}{2} < 2$),
     - $h(p_{k+1}) = \frac{3}{4}h(p_k) < h(p_k)$.
   - Case 2 (p_k odd):
     - $x_0 = 1.5p_k$, $x_m = \frac{1.5p_k + 0.5m}{2^m}$,
     - $p_{k+1} = x_m$ when x_m is an integer,
     - $x_0 = 1.5p_k < 2p_k$,

doi:10.20944/preprints202503.0499.v1

- Each step $x_{i+1} = \frac{x_i+0.5}{2}$ reduces the relative value:

$$x_{i+1} < \frac{2p_k}{2} + 0.25 = p_k + 0.25 < 2p_k \quad (\text{for } p_k \geq 1),$$

- Thus, $x_m < 2p_k$ for any required m,
- $h(p_{k+1}) = x_m \cdot 2^{-(k+1)} < 2p_k \cdot \frac{1}{2} \cdot 2^{-k} = p_k \cdot 2^{-k} = h(p_k)$.

3. Termination:
   - $h(p_k)$ strictly decreases,
   - $p_k \geq 1$ are positive integers,
   - If $h(p_k) \to 0$, p_k must stabilize at 1 after a finite number of steps (there cannot be infinitely many distinct integers ≥ 1).

Thus, for any $p_0 \geq 1$, $\{p_k\}$ converges to 1. □

6.5. *Generality and Robustness of the Proof*

The demonstration establishes that $p_{k+1} < 2p_k$ in all cases of the algorithm:

- When p_k is even, the factor $\frac{3}{2} < 2$ ensures immediate decrease.
- When p_k is odd, the iteration $x_{i+1} = \frac{x_i+0.5}{2}$ reduces the initial value $1.5p_k < 2p_k$ to an even smaller fraction after m steps, always maintaining $p_{k+1} < 2p_k$. This stems from the controlled nature of the algorithm, where the moderate growth ($\frac{3}{2}$) is outweighed by frequent reductions ($\frac{x+0.5}{2}$).

The metric $h(p_k) = p_k \cdot 2^{-k}$ precisely captures this behavior, guaranteeing that the sequence terminates at $p_t = 1$ for all $p_0 \geq 1$, completing a general and rigorous proof of the algorithm's convergence.

7. Verification Script

The following Python script starts with a positive integer entered by the user and then:

- Executes the Collatz conjecture, displaying the complete sequence,
- Identifies the numbers corresponding to the $4n-2$ series and displays them,
- Computes their positions and displays them,
- Shows the number of Collatz iterations and the number of cycles to the series,
- Applies the positional algorithm from the first position and displays the obtained positions,
- Compares the algorithm's positions with those from Collatz and reports whether they are exactly the same or not,
- Calculates the metric $h(p_k)$ to confirm that it strictly decreases until converging to 1 and displays it.

```
def collatz(n):
    numero_de_iteraciones_en_Collatz = 0
    secuencia_completa = [n]
    subconjunto_A = []

    if (n - 2) % 4 == 0:
        subconjunto_A.append(n)

    while n != 2:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1

        secuencia_completa.append(n)
        if (n - 2) % 4 == 0:
            subconjunto_A.append(n)

        numero_de_iteraciones_en_Collatz += 1

    return numero_de_iteraciones_en_Collatz, secuencia_completa, subconjunto_A

if __name__ == "__main__":
    try:
        numero_inicio = int(input("Enter a number to execute the Collatz conjecture: "))
        if numero_inicio < 1:
            print("Please enter a positive integer.")
        else:
            posicion = int((numero_inicio + 2) / 4)
    except ValueError:
        print("Invalid input. Please enter an integer.")

    try:
        if posicion < 1:
            print("Error.")
        else:
            numero = numero_inicio
            print("\nExecuting the Collatz conjecture...\n")

            iteraciones, secuencia_completa, subconjunto_A = collatz(numero)
            numero_de_ciclos = len(subconjunto_A)
            posiciones = tuple((x + 2) // 4 for x in subconjunto_A)

            print("Complete sequence:")
            print(secuencia_completa)
            print("\nSequence of numbers belonging to the subset A series of even numbers defined by
            print(subconjunto_A)
            print("\nPositions within the subset A series in each cycle:")
            print(posiciones)
            print("\nNumber of Collatz iterations:", iteraciones)
            print("Number of cycles within subset A:", numero_de_ciclos)
    except ValueError:
        print("Invalid input. Please enter an integer.")

def obtener_siguiente_posicion(posicionalgoritmo):
    # Multiply the position by 1.5
    x = posicionalgoritmo * 1.5
    # While x is not an integer, apply (x + 0.5) / 2
    while not x.is_integer():
        x = (x + 0.5) / 2
    return int(x)

# POSITION ALGORITHM
# Request the initial position
posicionalgoritmo = int(posiciones[0])
posicionesalgoritmo = (posicionalgoritmo,)

# Generate the sequence until reaching 1
while posicionalgoritmo != 1:
    posicionalgoritmo = obtener_siguiente_posicion(posicionalgoritmo)
    posicionesalgoritmo += (posicionalgoritmo,)

# Display the tuples
print("\nExecuting the positional algorithm...")
print("\nPositions within subset A generated by the algorithm:", posicionesalgoritmo)

# Compare the tuples
if posiciones == posicionesalgoritmo:
    print("\nALGORITHM CHECK: The positions are exactly the same as those obtained by applying the
else:
    print("\nALGORITHM CHECK: The positions do not match.\n")

# METRIC FOR DEMONSTRATING CONVERGENCE OF THE ALGORITHM

print("\nExecuting the metric to demonstrate convergence to 1 of the algorithm...\n")

def apply_rule(x):
    """Applies the rule of our conjecture."""
    if x.is_integer():
        return 1.5 * x  # If integer, multiply by 3/2
    else:
        return (x + 0.5) / 2  # If not integer, add 0.5 and divide by 2

def simulate_conjecture(n):
    """Simulates the conjecture, computes h(n_k), and verifies if it decreases strictly."""
    if not isinstance(n, int) or n < 1:
        print("Please enter a positive integer.")
        return

    # Lists to store the sequence and integers
    sequence = [n]
    current = float(n)
    integers = [n]
    h_values = [n * 2**0]  # List for h(n_k) values
    max_steps = 100000000000000000000  # Limit to avoid infinite loops

    # Generate the sequence until reaching 1
    for step in range(1, max_steps):
        next_value = apply_rule(current)
        sequence.append(next_value)

        # If the value is an integer, add it to the list and compute h
        if next_value.is_integer():
            integers.append(int(next_value))
            k = len(integers) - 1  # Index of the new integer
            h_k = next_value * (2 ** (-k))
            h_values.append(h_k)

        # If we reach 1 as an integer, stop
        if next_value == 1:
            current = next_value
            break

        current = next_value

    # Check if we terminated at 1
    if current != 1:
        print("Did not reach 1 within the step limit. Something went wrong!")
        return

    # Display the table of integers and metric
    print(f"\nIntegers in the sequence (n_k) and metric h(n_k):")
    print("k | n_k | h(n_k)")
    print("-" * 30)
    for k, (nk, hk) in enumerate(zip(integers, h_values)):
        print(f"{k} | {nk} | {hk}")

    # Verify if h(n_k) strictly decreases
    decreases_strictly = True
    for i in range(1, len(h_values)):
        if h_values[i] >= h_values[i-1]:
            decreases_strictly = False
            print(f"\nWarning! h(n_{i}) = {h_values[i]} >= h(n_{i-1}) = {h_values[i-1]}")
            break

    # Final report
    print("\nVerification result:")
    if decreases_strictly:
        print("The metric h(n_k) strictly decreases at each step.")
    else:
        print("The metric h(n_k) does NOT strictly decrease at each step.")

def main():
    """Main function to run the program."""
    try:
        n = int(posiciones[0])
        simulate_conjecture(n)
    except ValueError:
        print("Please enter a valid integer.\n")

if __name__ == "__main__":
    main()
```

8. Conclusion

We have demonstrated that the Collatz conjecture drives sequences through cycles that converge to the series $4n - 2$, terminating at the number 2. Our positional algorithm, along with the metric $h(p_k)$, confirms convergence to position 1, validating the conjecture for all positive integers and providing a significant advancement in its understanding.

References

1. L. Collatz, "Über eine Vermutung über die Iterationen von $3n + 1$," unpublished note, 1937 (attributed origin of the Collatz conjecture).
2. J. C. Lagarias, "The Ultimate Challenge: The $3x + 1$ Problem," American Mathematical Society, 2010.
3. T. Tao, "Almost all orbits of the Collatz map attain almost bounded values," arXiv:1909.03562, 2019.
4. C. J. Everett, "Iteration of the number-theoretic function $f(2n) = n$, $f(2n+1) = 3n+2$," Advances in Mathematics, vol. 25, no. 1, pp. 42–45, 1977.
5. J. H. Conway, "On unsettleable arithmetical problems," American Mathematical Monthly, vol. 79, no. 2, pp. 192–195, 1972.