Numerical verification of the Collatz Conjecture

Andreas-Stephan Elsenhans

University of Sydney

July 2026

Introduction

A dynamical system

Let $T : \mathbb{N} \to \mathbb{N}$ be $T(n) := n/2$, for n even, and $T(n) := \frac{3n+1}{2}$, for n odd. We consider the dynamical system

$$n,\ T(n),\ T(T(n)) = T^2(n),\ T(T(T(n))) = T^3(n),\ T^4(n), \ldots .$$

The system stops as soon as 1 is reached.

Examples

3, 5, 8, 4, 2, 1

27, 41, 62, 31, 47, 71, 107, 161, . . . , 3644, . . . , 4616, . . . , 23, 35, 53, 80, 40, 20, 10, 5, 8, 4, 2, 1 (70 steps)

Conjecture

The above iteration will always reach 1, independently of the initial positive integer value.

Names

Collatz conjecture (Lothar Collatz, 1910 - 1990), $3n+1$ conjecture, Syracuse problem, Ulam conjecture, Thwaites problem, Hasse's algorithm, . . .

Known examples

Theorem

The Collatz conjecture is true for all $n \leq 5.6 \cdot 10^{13}$.

Proof: Computation by Gary Leavens and Mike Vermeulen in 1992. On average an interval of length $6 \cdot 10^9$ in one night on one workstation.

Known examples

Theorem
The Collatz conjecture is true for all $n \leq 5.6 \cdot 10^{13}$.

Proof: Computation by Gary Leavens and Mike Vermeulen in 1992. On average an interval of length $6 \cdot 10^9$ in one night on one workstation.

Theorem
The Collatz conjecture is true for all $n \leq 2^{71} \approx 2.36 \cdot 10^{21}$.

Proof: Computation by David Barina on GPUs in 2025. On average 5 seconds of GPU time for an interval of length $2^{40} \approx 10^{12}$.

Comparison: One can check all numbers $< 2^{40}$ in about 2 minutes on a modern CPU.

Known examples

Theorem
The Collatz conjecture is true for all $n \leq 5.6 \cdot 10^{13}$.

Proof: Computation by Gary Leavens and Mike Vermeulen in 1992. On average an interval of length $6 \cdot 10^9$ in one night on one workstation.

Theorem
The Collatz conjecture is true for all $n \leq 2^{71} \approx 2.36 \cdot 10^{21}$.

Proof: Computation by David Barina on GPUs in 2025. On average 5 seconds of GPU time for an interval of length $2^{40} \approx 10^{12}$.

Comparison: One can check all numbers $< 2^{40}$ in about 2 minutes on a modern CPU.

This talk
We do not have more GPU time. But can we verify the conjecture in other particular cases?

Analysing the dynamics

Lemma

The maps

$$2\mathbb{Z}/2^{k+1}\mathbb{Z} \to \mathbb{Z}/2^k\mathbb{Z}, n \mapsto \frac{n}{2}, \text{ and}$$
$$(1+2\mathbb{Z})/2^{k+1}\mathbb{Z} \to \mathbb{Z}/2^k\mathbb{Z}, n \mapsto \frac{3n+1}{2}$$

are bijections.

Analysing the dynamics

Lemma

The maps

$$2\mathbb{Z}/2^{k+1}\mathbb{Z} \to \mathbb{Z}/2^k\mathbb{Z}, n \mapsto \frac{n}{2}, \text{ and}$$
$$(1+2\mathbb{Z})/2^{k+1}\mathbb{Z} \to \mathbb{Z}/2^k\mathbb{Z}, n \mapsto \frac{3n+1}{2}$$

are bijections.

Corollary

The first k steps of the iteration depend only on the last k bits of n.

Corollary

Prescribe the first k steps of the iteration arbitrarily (i.e., either $n \to \frac{n}{2}$ or $n \to \frac{3n+1}{2}$). Then there is exactly one way to choose the trailing k bits of a number such that the iteration starts in the prescribed way.

Compression of the dynamics – standard polynomials

Corollary

For every residue class $\overline{A} \in \mathbb{Z}/2^k\mathbb{Z}$, there exists a unique linear polynomial $S_{k,A}(x) = \frac{3^e x + s}{2^k}$ such that

$$T^k(a) = S_{k,A}(a) \text{ for all } a \in \overline{A}\,.$$

These polynomials are called *standard polynomials*.

Remark

The exponent e occurs exactly $\binom{k}{e}$ times in the standard polynomials $S_{k,\star}$. The largest leading coefficient $(3/2)^k$ occurs only once.

Remark

The constant term of a standard polynomial is bounded by $(3/2)^k - 1$.

Statistics of the standard polynomials

Question

What can we say about the distribution of the coefficients of $S_{k,\star}$?

Leading term

The mean value of the logarithm of the leading coefficient is $\frac{k}{2}\log(3) - k\log(2) \approx -0.144k$, the standard deviation is $\sqrt{k}\log(3)/2$.

Consequence

The probability for $T^k(x) < x$ tends to 1 for $k \to \infty$.

Statistics of the standard polynomials

Question

What can we say about the distribution of the coefficients of $S_{k,\star}$?

Leading term

The mean value of the logarithm of the leading coefficient is $\frac{k}{2}\log(3) - k\log(2) \approx -0.144k$, the standard deviation is $\sqrt{k}\log(3)/2$.

Consequence

The probability for $T^k(x) < x$ tends to 1 for $k \to \infty$.

Definition

The smallest k such that $T^k(x) < x$ is called *stopping time*.
The smallest k such that $T^k(x) = 1$ is called *total stopping time*.

Refined analysis

Theorem
Almost all positive integers have a finite stopping time.

Proof: Riha Terras, 1976.

Theorem
For almost all $x \in \mathbb{N}$, there exists an $k \in \mathbb{N}$ such that $T^k(x) < x^{0.7924}$.

Proof: Refined analysis of the above. Ivan Korec, 1994.

Refined analysis

Theorem
Almost all positive integers have a finite stopping time.

Proof: Riha Terras, 1976.

Theorem
For almost all $x \in \mathbb{N}$, there exists an $k \in \mathbb{N}$ such that $T^k(x) < x^{0.7924}$.

Proof: Refined analysis of the above. Ivan Korec, 1994.

Theorem
For almost all $x \in \mathbb{N}$, there exists an $k \in \mathbb{N}$ such that $T^k(x) < \log(x)$.

Proof: Requires far more than the above. Terence Tao, 2022.

Open question:
Is the density of the integers that satisfy the conjecture positive?

Some irregularities

Observation

The result of $n \rightsquigarrow 3n+1$ is not divisible by 3. All subsequent terms are not divisible by 3.

Experiment

Starting with $123^{456}, \ldots, 123^{456} + 100\,000$. After 39 iterations, only 9679 distinct numbers remain. After 191 iterations, only 100 numbers remain. They are in the range $4.2 \cdot 10^{934} \ldots 1.38 \cdot 10^{955}$. (We started with $9.925 \cdot 10^{952}$.). From the initial $10^5 + 1$ elements 23152 reached the same.

Some irregularities

Observation

The result of $n \rightsquigarrow 3n+1$ is not divisible by 3. All subsequent terms are not divisible by 3.

Experiment

Starting with $123^{456}, \ldots, 123^{456} + 100\,000$. After 39 iterations, only 9679 distinct numbers remain. After 191 iterations, only 100 numbers remain. They are in the range $4.2 \cdot 10^{934} \ldots 1.38 \cdot 10^{955}$. (We started with $9.925 \cdot 10^{952}$.). From the initial $10^5 + 1$ elements 23152 reached the same.

Search

The total stopping times of $238! + 1, \ldots, 238! + 10^9$ are all equal.

Proof

Computation by Dmitry Kamenetsky and Martin Ehrenstein.

Computer experiments

Philosophy

The analysis above was based on standard polynomials. Thus, the computer experiments need to use them systematically.

$$S_{2,0} = \frac{X}{4},\ S_{2,1} = \frac{3X+1}{4},\ S_{2,2} = \frac{3X+2}{4},\ S_{2,3} = \frac{9X+5}{4}.$$

$$S_{5,3} = \frac{9X+5}{32},\ S_{5,7} = \frac{81X+73}{32},\ S_{5,11} = \frac{27X+23}{32},$$
$$S_{5,15} = \frac{81X+65}{32},\ S_{5,19} = \frac{27X+31}{32},\ S_{5,23} = \frac{27X+19}{32},$$
$$S_{5,27} = \frac{81X+85}{32},\ S_{5,31} = \frac{243X+211}{32}.$$

Interpretation

All integers $n \not\equiv 7, 15, 27, 31 \bmod 32$ have a finite stopping time. Only the remaining four classes modulo 32 have to be checked.

https://oeis.org/A076227

Number of surviving Collatz residues mod 2^k

1, 1, 1, 2, 3, 4, 8, 13, 19, 38, 64, 128, 226, 367, 734, 1295, 2114, 4228, 7495, 14990, 27328, 46611, 93222, 168807, 286581, 573162, 1037374, 1762293, 3524586, 6385637, 12771274, 23642078, 41347483, 82694966, 151917636, 263841377, 527682754, 967378591, 1934757182, 3611535862, 6402835000

Thus, only 6 402 835 000 elements in an interval of length 2^{40} need to be checked.

Remark

This is one of the tricks that enable checking of all integers up to 2^{71}.

Computation

Checking the conjecture up to 2^{40} by a recursive enumeration of the 6 402 835 000 classes takes about 2 minutes. The recursive enumeration takes about 35 seconds.

What about checking large numbers?

Lack of research

So far, checks of random large numbers have mostly been done by amateur mathematicians.

Complexity

Assuming that a random ℓ-bit number needs about $C\ell$ iterations to reach 1, the check would require $O(\ell^2)$ bit operations if the check is done by computing $n, T(n), T^2(n), \ldots$.

What about checking large numbers?

Lack of research

So far, checks of random large numbers have mostly been done by amateur mathematicians.

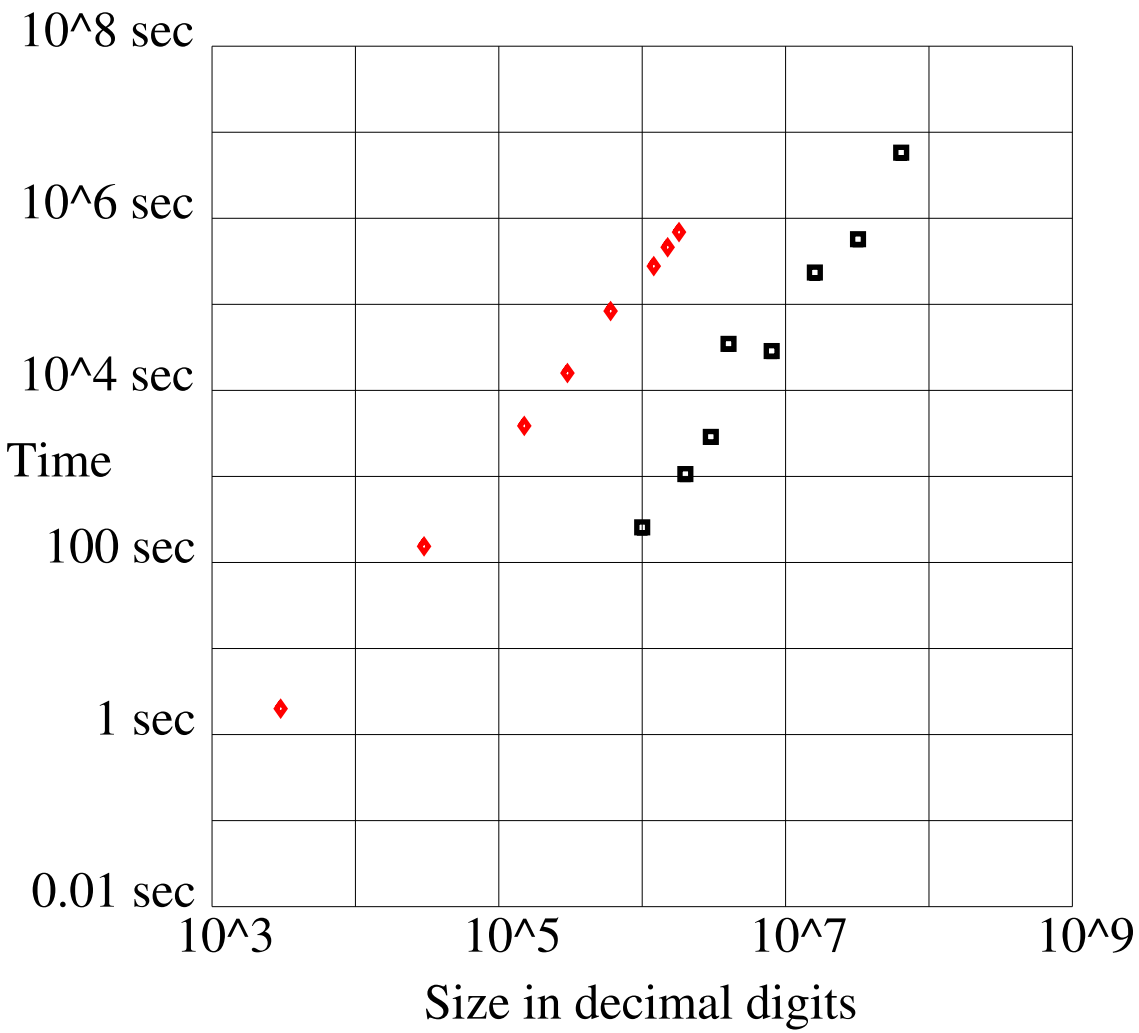
Complexity

Assuming that a random ℓ-bit number needs about $C\ell$ iterations to reach 1, the check would require $O(\ell^2)$ bit operations if the check is done by computing $n, T(n), T^2(n), \ldots$.

Naive improvements

Precompute all the 256 standard polynomials $S_{8,\star}$ and do 8 steps at a time. This gives a speedup by a constant factor. One could also try $S_{12,\star}$ or $S_{16,\star}$.

Run times posted on the web



Diamonds: W. Ren (2023), `https://eprint.iacr.org/2023/648.pdf`
Squares: `https://www.reddit.com/r/Collatz/comments/13g7cb0`

What do we have to compute?

Observation

We need the last bit of each number in the sequence to determine the next step.

Recursion formula

$$S_{k+\ell,a} = S_{k,S_{\ell,a}(a) \bmod 2^k} \circ S_{\ell,a \bmod 2^\ell}$$

Advantage

Compared to the naive approach, many steps in between work only with small numbers or are done modulo small powers of 2.

Implementation

Algorithm

Compute standard polynomials by concatenation of smaller ones.

```
function Collatz_rec(n, a)
  if n le 1 then
    return Collatz_direct(n, a);
  end if;
  t1 := n div 2;
  t2 := n - t1;
  p1 := Collatz_rec(t1, ModByPowerOf2(a,t1));
  r2 := MyEval(p1,ModByPowerOf2(a, n));
  p2 := Collatz_rec(t2, ModByPowerOf2(r2,t2));
  return MySubs(p2,p1);
end function;
```

Checking the conjecture

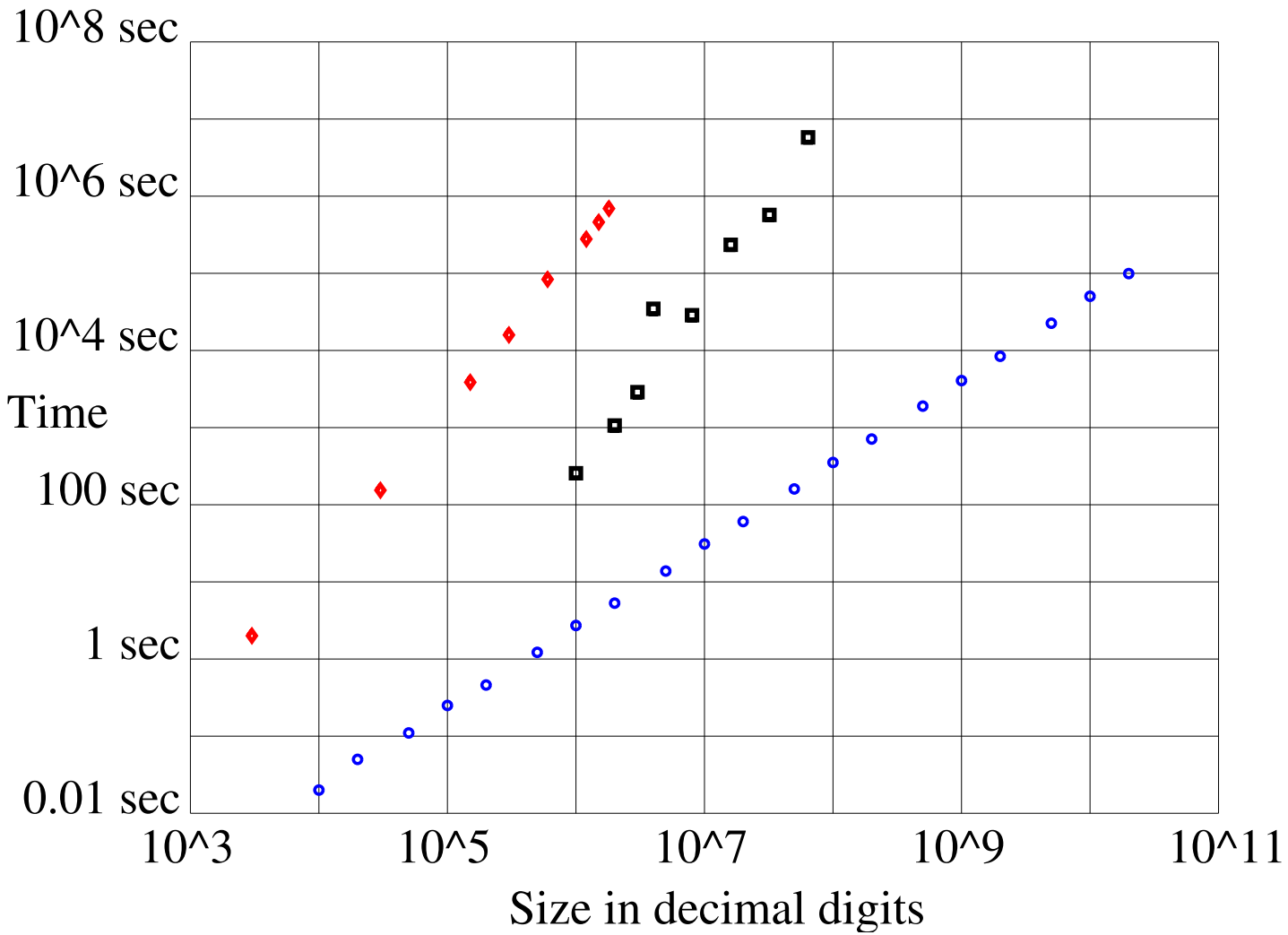
Algorithm

For an n-bit number, we compress $n/4$ steps in a macro-step by using one standard polynomial.

```
function total_stopping_time(a)
  cnt := 0;
  repeat
    StepSize := Max(1,Ilog2(a) div 4);
    cnt := cnt + StepSize;
    pol := Collatz_rec(StepSize, a);
    a := MyEval(pol, a);
  until a eq 1;
  return cnt;
end function;
```

Only the evaluation of the standard polynomial needs full-length integers.

Run time



Circles: New algorithm

Statistics

Number of decimal places	Number of examples	Average Number of steps to reach 1	Standard Deviation	Time
10 000	10000	160 085	1531	0.02s
100 000	10000	1 600 728	4838	0.25s
1 000 000	10000	16 007 868	15177	2.72s
10 000 000	10000	160 077 568	48567	31s
100 000 000	4000	1 600 785 302	152901	353s
1 000 000 000	400	16 007 826 613	516305	1.13h
10 000 000 000	1	160 079 821 246	—	13.2h

Put $\gamma := 1/\log(2/\sqrt{3})$, Then the average is about $\gamma \cdot \log(n_0)$ and the standard deviation is about $\gamma^{3/2} \cdot \sqrt{\log(n_0)} \cdot \log(3)/2$.

Modelling the dynamic as a random walk with drift predicts exactly this. I.e.,

$$\log(n) \rightsquigarrow \log(n) - \log(2) \quad \text{or} \quad \log(n) \rightsquigarrow \log(n) + \log(3) - \log(2)$$

with equal probabilities.

C versus magma

Possible optimisations

All operations with small integers can be done in `long ints`. The remaining operations are done using `GMP`.

AI tools

ChatGPT can convert `magma` code to `C` code. It recognises correctly when to use large integers. I asked to use `GMP`. Only one bug was introduced.
I added the above optimisation by hand.

Result

7 times faster as long as the data fit into the cache. 3.5 times faster on very long numbers.

Conclusion

The interpreter overhead is much smaller than expected.

Complexity analysis

Definition

$$G_c := \{n \in \mathbb{N} \mid T^k(n) = 1 \text{ for some } k \leq cB(n)\}$$

with $B(n)$ the bit-length of n.

Conjectural density

Assuming the total stopping time follows the observed distribution, the set G_5 has density 1.

Naive check

Checking the conjecture for $n \in G_c$ directly has complexity $O_c(B(n)^2)$.

Theorem

The complexity to compute $S_{k,a}$ is $O(k \log(k)^2 \log(\log(k)))$.
The new algorithm checks the conjecture for $n \in G_c$ in

$$O_c(B(n) \log(B(n))^2 \log(\log(B(n)))) .$$

Summary

Collatz conjecture
Is an open problem for which several partial results exist. Intensive computations have been done for a numerical verification.

New Algorithm
Using binary splitting, standard polynomials are computed efficiently. This leads to a fast verification of the conjecture for large numbers.

Statistical analysis
The data suggest that a 1-dimensional random walk with drift is a good model for predicting the total stopping time.

Summary

Collatz conjecture
Is an open problem for which several partial results exist. Intensive computations have been done for a numerical verification.

New Algorithm
Using binary splitting, standard polynomials are computed efficiently. This leads to a fast verification of the conjecture for large numbers.

Statistical analysis
The data suggest that a 1-dimensional random walk with drift is a good model for predicting the total stopping time.

Thank you!