Experimental Evidence for the Claim of Collatz Termination

Contents

Introduction 1
Computation 2
Run-Time 3
upStep Count 4
Discussion 5
Statistical Proofs 5
'Existence Proof' Using Statistics 6
Sizes of Infinite Sets 7
Testing Methodology 9
Interpreting the Test Results 10
371,642 digits 11
973,702 digits 12
3,716,420 digits 12
6,726,720 digits 13
Philosophical Position 13
Version History 14

Introduction

If a proof relies on experimental evidence, it is only fair to publish such evidence in an accessible manner. The raw numerical data and code is provided in a linked zip file.

Since the Collatz sequence has excited the interest of a large spectrum of people, it is only reasonable to present the results in as simple and accessible a way as is possible.

This paper is formatted for reading on a landscape monitor with internet access so that the hyperlinked references can be seamlessly and effortlessly consulted.

There are two key papers to be referenced:

REF1: *The Negative Collatz Sequence*, L.O.Green. v1.20, Aug 2022

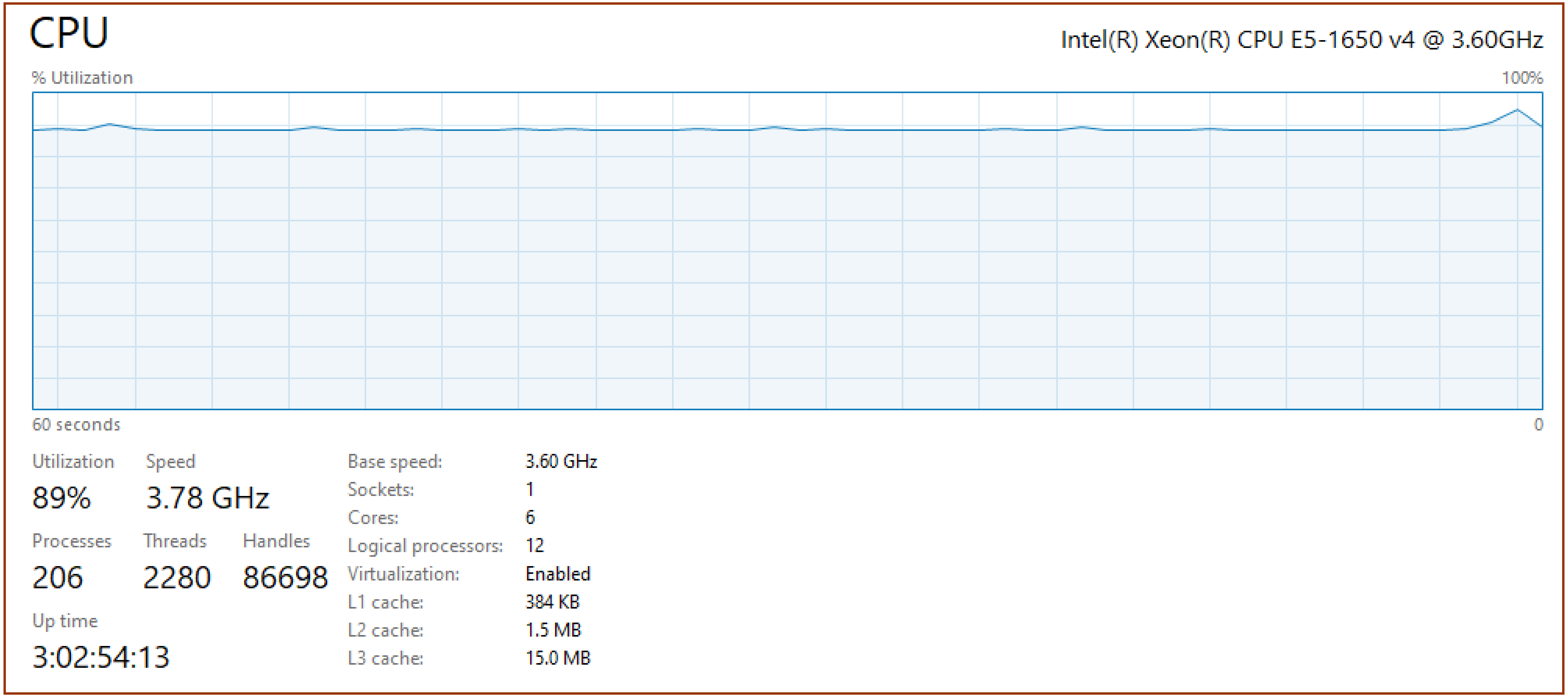
REF2: *The Inner Structure of the Collatz Iteration Sequence*, L.O.Green, v1.72, Aug 2022

In order for the Collatz sequence to terminate the algorithm has been adjusted as shown below:

- **START** from any positive integer
- **If** the value is even then divide it by two,
  else multiply it by 3 and add 1.
- Repeat the previous step if the resultant value is greater than one.
- **TERMINATE**

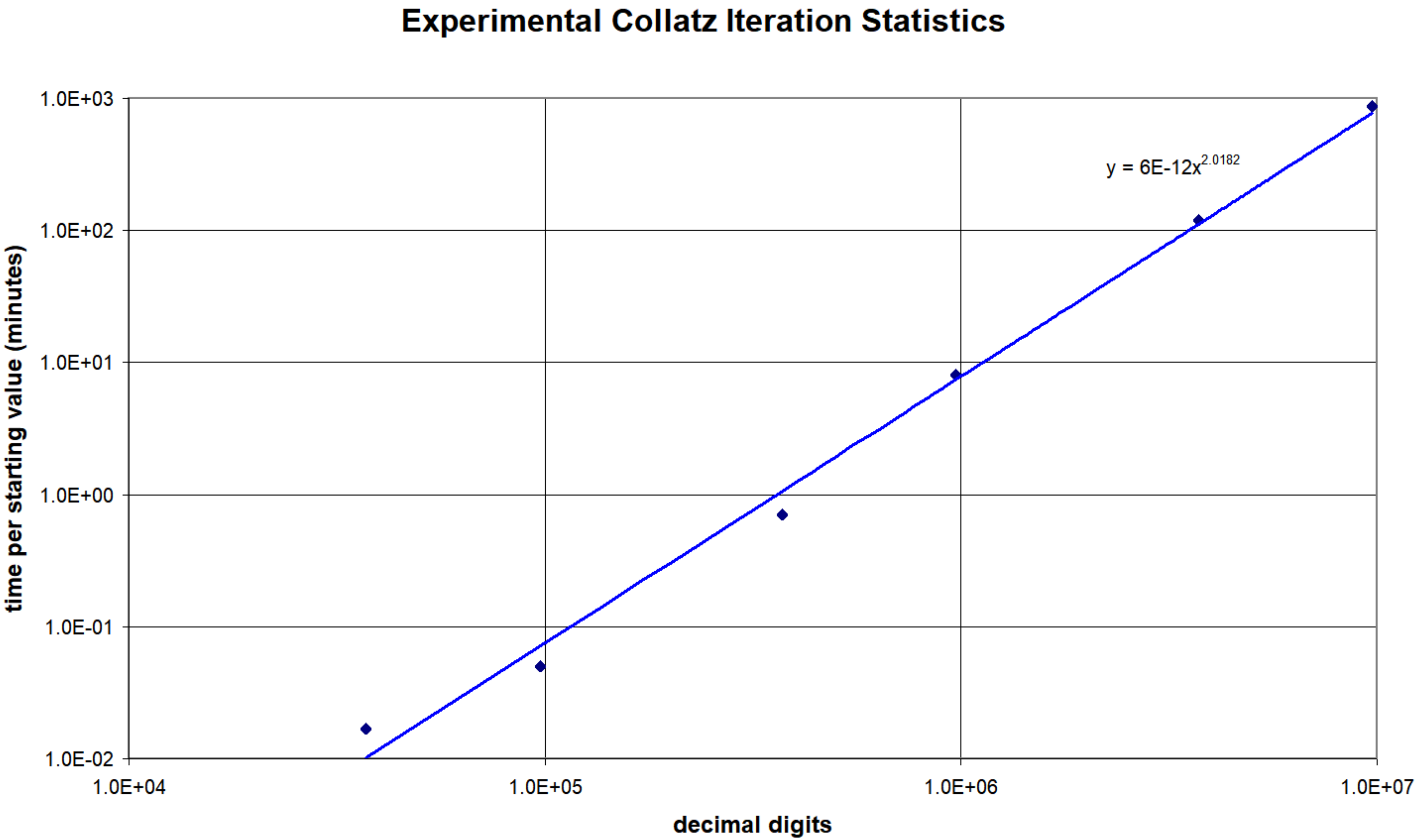
Computation

We will be handling numbers having in excess of 1 million decimal digits. In order to do this a large integer library is clearly needed. The chosen library was a WIN x64 build of MPIR. The workstation used for the testing is detailed below:



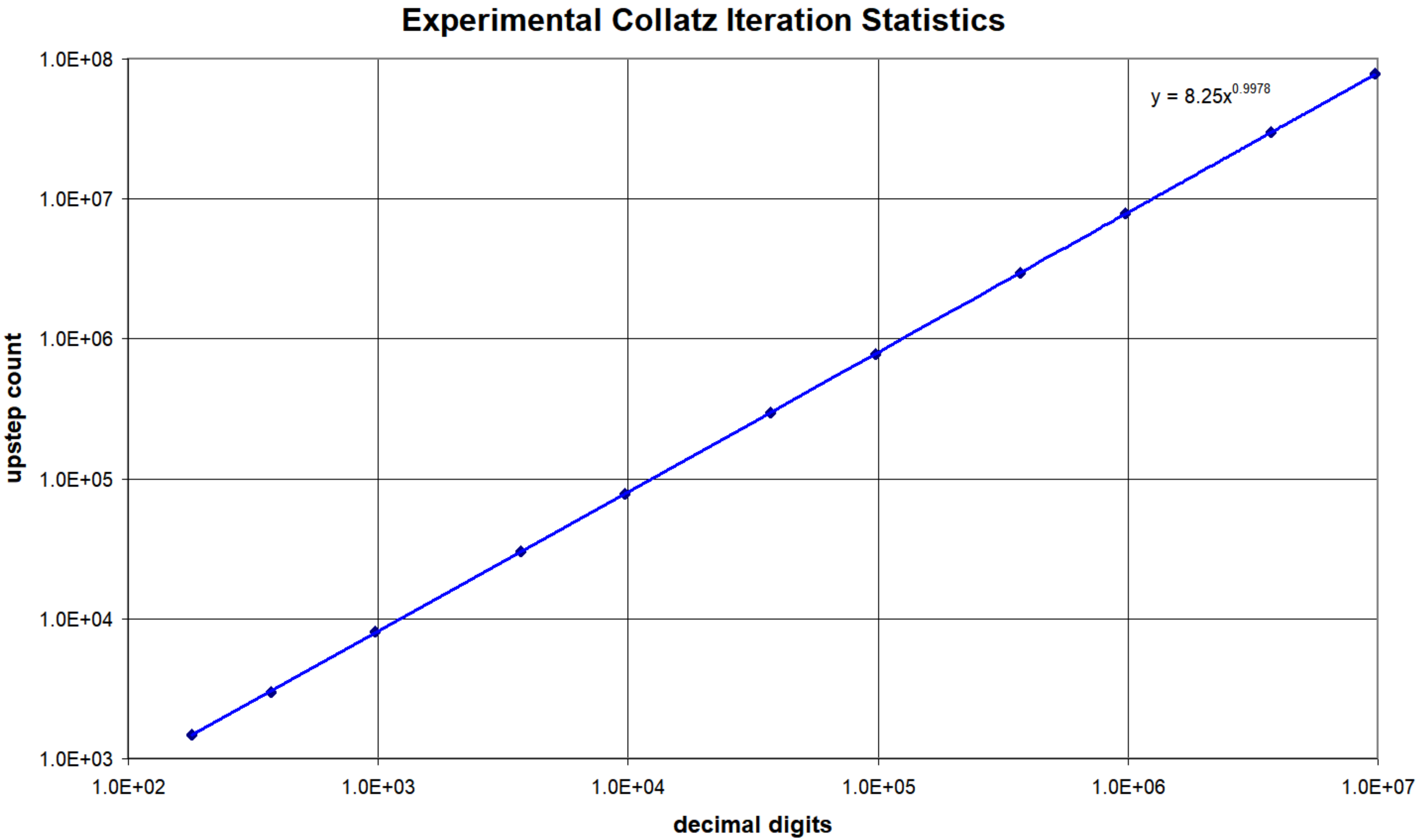
All timings presented refer to this machine.

Run-Time


Experimental Collatz Iteration Statistics
y = 6E-12x^2.0182
time per starting value (minutes)
1.0E+03
1.0E+02
1.0E+01
1.0E+00
1.0E-01
1.0E-02
1.0E+04
1.0E+05
1.0E+06
1.0E+07
decimal digits

upStep Count

Each (3× + 1) step is called an upStep, and we have counted these.



Discussion

It should be noted that the maximum x-extent of those graphs was around ten million decimal digits! At this level, a single starting value took around 14½ hours to terminate. This is the practical limit of what can reasonably be achieved with the available computational setup.

In the first revision of REF1 the upper test limit was 4E371641, which I felt was excessively and unnecessarily high. However, commenters on the YouTube summary of my claims regarded such a limit as not nearly adequate. Hence the desire to revisit that limit, and see just how high it can be pushed.

Statistical Proofs

The Rank Table (REF2, p.15) systematically lists all odd numbers and shows how many iteration down steps each takes before the next odd number is reached. A Rank 1 (R1) value has a single upstep (U) and a single downstep (D) before the next odd value is reached. Every other odd number is systematically an R1 value, in other words half of all odd numbers are R1 values (in the sense of natural density). Numbers which are not in R1 are either in R2 or a higher numbered rank.

R1 values go U then D so they are roughly 3/2 greater than their initial values. R2 values go UDD so they are roughly 3/4 of their original value. It is very reasonable to say that if we start at a random odd value, it is equally likely that we go up by 3/2 or down by 3/4.

All the odd numbers are systematically placed in the Rank Table. What happens next, that is after the first iteration steps, is not *necessarily* random. The Collatz sequence is entirely deterministic, so if you now say that there is again a 50% chance of hitting an R1 value that is not yet proven.

As an example, after the random first starting point, the outcome of the first iterations steps to the next odd value always produces a number of the form (6k ± 1). (REF2, p.19). Odd numbers, divisible by 3, are never found in the middle of an iteration chain. You can start from such a value, or you can start from an even value, divisible by 3, but you cannot otherwise jump in to such a value. Divisible-by-3 values are systematically non-random.

If we look *uncritically* at the iteration steps so far the calculation could go up or down with equal probability, making the resultant value (3/2)×(3/4) on average, which is 9/8 = 1.125. That is not tending towards termination. But the true picture is more complicated.

R1 is 1/2 of odd numbers with 3/2 = 1.5000 scale.
R2 is 1/4 of odd numbers with 3/4 = 0.7500 scale.
R3 is 1/8 of odd numbers with 3/8 = 0.3750 scale.
R4 is 1/16 of odd numbers with 3/16 = 0.1875 scale.
R5 is 1/32 of odd numbers with 3/32 = 0.0938 scale.
…

There is no limit to the Rank index, and the scale factor drops by a factor of two for each next Rank. The averaged scale factor, assuming the Ranks appear in an iteration in proportion to their abundance, is therefore an appropriately weighted product. Suppose we only go as far as R5. We have failed to consider the infinity of Ranks beyond 5, so we double the R5 population and say the resultant factor will be lower than our calculation suggests.

Of 32 scale factors, 16 are R1, 8 are R2, 4 are R3, 2 are R4 and 2 are R5, *on average*.

$$F = \left(\frac{3}{2}\right)^{16}\left(\frac{3}{4}\right)^{8}\left(\frac{3}{8}\right)^{4}\left(\frac{3}{16}\right)^{2}\left(\frac{3}{32}\right)^{2} \quad = \frac{3^{32}}{2^{62}} \quad = 0.0004$$

This is an *apparently* powerful argument that the Collatz sequence terminates. Unfortunately such a statistical argument neglects the basics of the solution to the Collatz iteration sequence.

The Collatz sequence has exactly three possible outcomes for any particular starting point:

1) it terminates
2) it wanders off towards infinity, and therefore never terminates.
3) it gets stuck in a loop, and therefore never terminates.

Any proof must at least consider all three possibilities.

The statistical argument above neglects the case of loops, which render any average trend meaningless.

‘Existence Proof’ Using Statistics

In effect what I am saying here is that *using* statistics does not automatically make a proof “statistical” in the sense of the previous section.

In the folk lore of the old wild west there is the notion of a dead man’s hand whilst playing cards. The combination of cards was so improbable that it was used as evidence of cheating. The holder of such a hand typically then acquired acute lead poisoning. The same idea applies to dice such that improbable outcomes are associated with loaded (biased) dice.

The basis of the argument presented in REF1 is along these same lines. If loops or wandering off to infinity properties exist within the Collatz sequence, they must leave evidence of their presence. Study of the Negative Collatz sequence ($^{-}$Collatz), which has two loops, shows that each of the three possible outcomes has the same probability of occurrence, namely 33%. It means that 66% of starting values do not terminate.

Suppose you had a pair of dice and you wanted to know if they had been ‘fixed’ to never land on 5 or 6. How many times would you consider it reasonable to throw the dice, never seeing a 5 or a 6, before you decided they were almost certainly fixed? This is analogous to the situation being tested.

Sizes of Infinite Sets

It was established (REF2) that it is not possible for there to be a single counter-example to the Collatz conjecture, there have to be infinitely many counter examples. At least one commenter on the YouTube summary stated that this conclusion was so obvious as to not need stating explicitly. I strongly disagree.

Consider the Level 0 set of REF2.

L0 = { 2, 4, 8, 16, 32, 64, 128, … }

Up to some large value 2^H , we only have H values in L0.

L0 is an infinite set which has almost no elements in it in the sense of natural density, the fractional amount tending to zero very rapidly. For example only 1 in 1,000,000 consecutive natural numbers are in L0 for H = 25. We clearly regard this as a pretty small infinity.

L1 is the next infinite set in the Structure Table.

L1 = { 5, 21, 85, 341, 1365, 5661, … }. These are spaced by a $(4\times + 1)$ rule, so there are always less than half as many as in the L0 set for a given upper limit. An even smaller infinity.

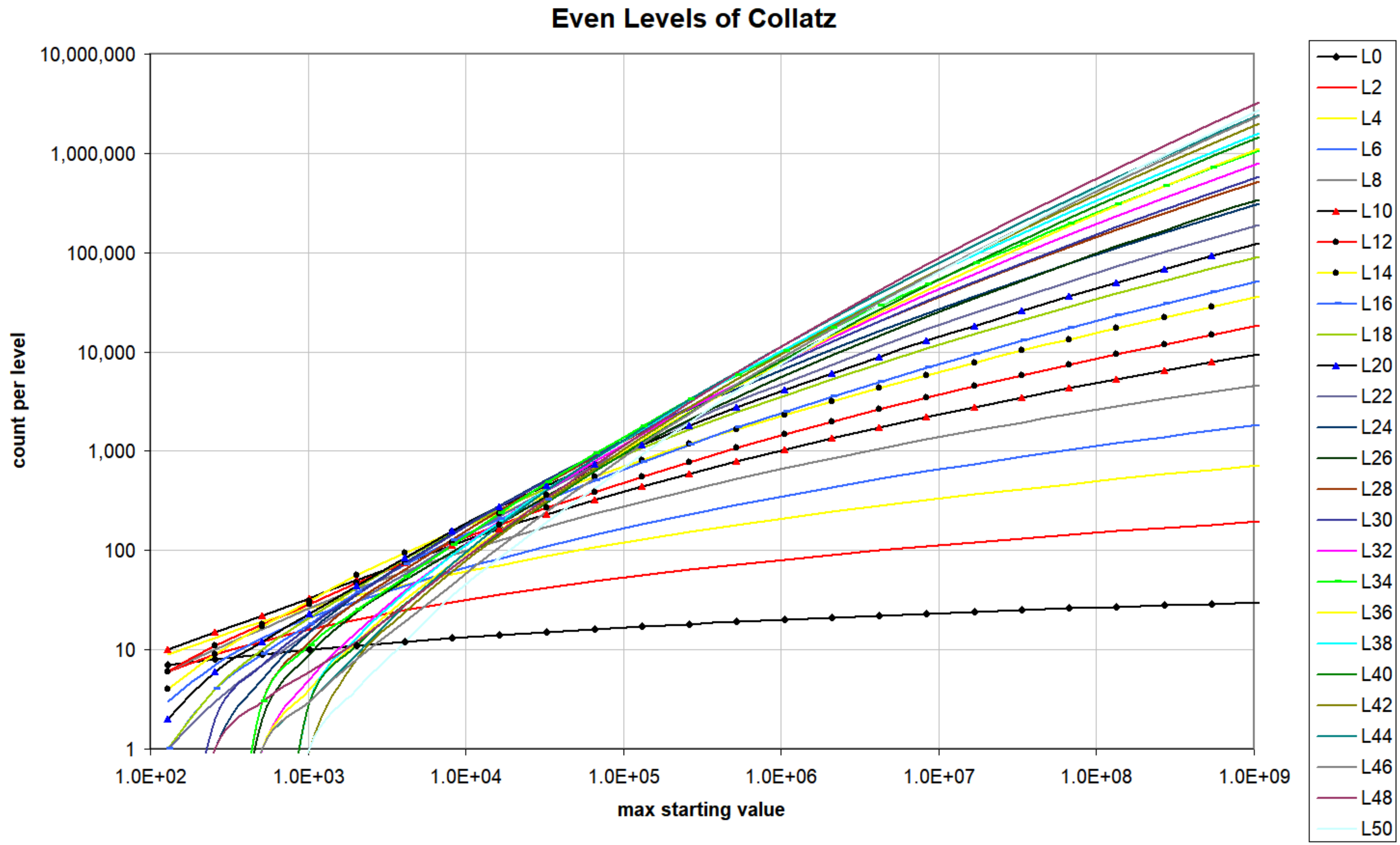
However, each value in L1 creates an infinite power-of-two multiple of that L1 value in L2, so now we have a product of (small) infinities.

L3 is smaller than L2 since only half of L2 values are accessible from L3. The point is that in some way the infinity of sets of infinite sets fills out to cover all the available numbers up to 1E20 (the brute-force tested known terminating values). This is not as improbable at it might at first seem since direct calculation of the sizes of the Levels sets shows that the higher levels grow faster than the lower Levels, at least as far as the computation was continued.

The plot overleaf shows that the rate of growth of the higher Levels is far greater than the growth of the lower Levels.

Consider all starting values up to 1 million, shown on the graph as 1.0E+06. There are 20 points in L0, 1029 in L10, 4142 in L20, 7729 in L30, and 8462 in L40.

Of course there are 1 million starting values up to 1 million, so most of them do not occur in the Levels up to L50 that we have plotted. This agrees with our intuition that larger numbers take longer to iterate down to 1 than smaller numbers.


Even Levels of Collatz
count per level
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
1.0E+02
1.0E+03
1.0E+04
1.0E+05
1.0E+06
1.0E+07
1.0E+08
1.0E+09
max starting value
L0
L2
L4
L6
L8
L10
L12
L14
L16
L18
L20
L22
L24
L26
L28
L30
L32
L34
L36
L38
L40
L42
L44
L46
L48
L50

The evidence from the ⁻Collatz outcome map (REF1, p11) suggests that there is a minimal gap between the lowest counter-examples and uniform outcome density for the three possible outcomes. However, for the negative sequence the low limit values are 5 for the smaller loop and 17 for the larger loop. The point is that you do not have to 'wait' to say 500 for the density of outcomes for the loops to gradually increase to their 33% levels. Visually, those densities seem to appear immediately.

We do see, however, that localised density clumping certainly occurs for larger values. For example around 3E301 (REF1, pp19-21) the outcome maps are decidedly 2-colour rather than 3-colour. Nevertheless, non-termination is still found with better than 50% probability per point.

An outcome map from the (positive) Collatz sequence would be boring as it would be all one colour for every test we have ever done.

Testing Methodology

We are now testing beyond any reasonable number limit. The numbers have almost no meaning in terms of written values. How do you even write a 100,000 decimal digit number? If a counter example is found, how does anyone get to verify it?

The test method employed is to create the large numbers from smaller random numbers which are easy to communicate.

```
int first = 5000 - 123 + ( rand() % (5000 + 122) );
int second= 101 + ( rand() % 191 );
int third = 102 + ( rand() % 895 );

double dStart = first * pow(10.0, second);
```

We then build the large starting value from a formula:

```
mpz_set_d(start, dStart);
mpz_add_ui(start, start, third);
mpz_setbit(start, twos);
i64 extra = p + i64(twos/101);
mpz_setbit(start, extra);
```

+= is the C-language notation to add on the new value to the existing value.

If start was 31, start += 3 would give 34.

p is the loop counter, stepping by +1 for each new starting value.

$$start = first \times 10^{second}$$

$$start += third$$

$$start += 2^{twos}$$

$$start += 2^{p+(twos/101)}$$

A text file name is created from the values of first and second, containing the ‘recipe’ to construct that starting value in case it should be a non-terminating value.

Non-terminating values are hard to find in this sense, we don’t know what they are! We can’t say it is when the iteration reaches a specific value, and we can’t reasonably look for a repeating number without storing millions of prior iteration results.

The simple method used is to count upsteps, and if they exceed 4× the twos count we flag that as a ***potential*** non-terminating value. Since we have never found one, the next step has so far not been taken.

Interpreting the Test Results

We use statistics to *interpret* the test results, rather than to prove a theorem. If a non-terminating value is ever found then the job is done. If such a value is never found, which is looking remarkably probable, we have to consider how useful from an evidentiary point of view, the non-existence of such a result is.

How many times would you need to toss a coin to be confident the coin had two tails and no heads, no other inspection being possible? 10 tails in a row should only happen once in 2^{10} times (=1024). 20 tails in a row is around 1 in a million. 30 tails, one in a billion.

At what ***threshold*** do you say enough is enough?

371,642 digits

It is convenient to enter the primary starting value as:

twos = 1234567

This gives rise to a number with 371,642 decimal digits. We run 10 instances of the program (on the 12 logical processor computer) to accelerate the testing. Each starting value takes around 69 seconds to iterate down to 1.

It should be noted that running only 6 instances of the program (1 per core) allows completion in around 43 seconds per starting value.

We have 10,000 starting values as our evidence.

The calculation is just like a school problem. An event happens randomly with 50% probability. What is the chance that it does **not** happen after 10,000 tries?

Events, N	Probability of happening once	Probability of not happening after N tries
10,000	50%	0.50^{10000} = 5E-3011
10,000	10%	0.90^{10000} = 3E-458
10,000	1%	0.99^{10000} = 2E-44

I therefore claim with greater than 99.999% confidence that no counter-example for the Collatz Conjecture exists below a number with 371,641 decimal digits.

Negative Collatz Results

(Using version 2 code).

It is convenient to enter the primary starting value as:

twos = 1234567

This gives rise to a number with 371,642 decimal digits.

From 200 tests we have 97 which terminate at 1, and 103 which loop.

We use a test on the upCount value to decide when we might have encountered a loop. The value “ratio” is the upCount divided by the variable twos, which sets the start value as a power of two. The negative Collatz sequence generates a considerably higher upCount value (and hence ratio) for a given starting value. Typically the ratio is 4.7 for the negative Collatz sequence (on terminating values) whereas for the standard Collatz sequence a ratio of 2.5 was more common.

973,702 digits

It is convenient to enter the primary starting value as:

twos = 3234567

This gives rise to a number with 973,702 decimal digits. We run 10 instances of the program (on the 12 logical processor computer) to accelerate the testing. Each starting value takes around 8 minutes to iterate down to 1.

We have 1,100 starting values as our evidence.

Events, N	Probability of happening once	Probability of not happening after N tries
1,100	50%	0.50^{1100} = 7E-332
1,100	10%	0.90^{1100} = 5E-51
1,100	1%	0.99^{1100} = 2E-5

I therefore claim with greater than 99.99% confidence that no counter-example for the Collatz Conjecture exists below a number with 973,701 decimal digits.

Negative Collatz Results

(Version 2 code) It is convenient to enter the primary starting value as: twos = 3234567
This gives rise to a number with 973,702 decimal digits.

124 tests with only 22 non-terminating values found (18%). Hence using 1% probability should give adequate confidence.

3,716,420 digits

It is convenient to enter the primary starting value as:

twos = 12345678

This gives rise to a number with 3,716,420 decimal digits. We run 10 instances of the program (on the 12 logical processor computer) to accelerate the testing. Each starting value takes around 2 hours to iterate down to 1.

We have 340 starting values as our evidence.

Events, N	Probability of happening once	Probability of not happening after N tries
340	50%	0.50^{340} = 5E-103
340	10%	0.90^{340} = 3E-16
340	1%	0.99^{340} = 0.033

I therefore claim with greater than 96% confidence that no counter-example for the Collatz Conjecture exists below a number with 3,716,419 decimal digits.

update v1.10:

With 410 iterations at the next level up we can reduce the uncertainty by now claiming 750 total events.

0.99^{750} = 5.3E-4   better than 99.9% confidence

6,726,720 digits

It is convenient to enter the primary starting value as:

twos = 22345678

This gives rise to a number with 6,726,720 decimal digits. We run 10 instances of the program (on the 12 logical processor computer) to accelerate the testing. Each starting value takes around 12 hours to iterate down to 1.

We have 410 starting values as our evidence.

Events, N	Probability of happening once	Probability of not happening after N tries
410	50%	0.50^{410} = 4E-124
410	10%	0.90^{410} = 2E-19
410	1%	0.99^{410} = 0.016

I therefore claim with greater than 98% confidence that no counter-example for the Collatz Conjecture exists below a number with 6,726,719 decimal digits.

You might reasonably be wondering where the 1% probability figure has come from. Frankly, it is pretty much plucked out of the air. If there are two possible outcomes for the $^{+}$Collatz sequence, we expect each to have a 50% probability on average. Using 1% adequately allows for localised clumping in some poorly defined way. We do see incidence of 1% probabilities on the $^{-}$Collatz outcome maps, but we hope to avoid these, on average, by using random values over a range in excess of 10^{190}.

Philosophical Position

We have considered numbers having well in excess of 6 million digits, and with high confidence established that no counter examples exist. If counter examples do exist beyond say 50,000,000 digits, it has to be wondered if this has any practical implications, given the complexity of the calculations involving such counter examples.

It might reasonably be argued that such counter examples are irrelevant in the world of computing, and we can therefore move on to other problems. In this case, *for all practical purposes*, we might reasonably say that the Collatz Conjecture is true based on the analytic and experimental evidence presented.

Version History

v1.10: 24 Aug 2022. Add Negative Collatz results to 2^1234567.
and 2^3234567.
Add a page for 2^22345678.
Update confidence on 2^12345678.
Add a page for 2^22345678 (6.7 million digits)

v1.00: 5 Aug 2022, first release.

http://lesliegreen.byethost3.com/publications.html