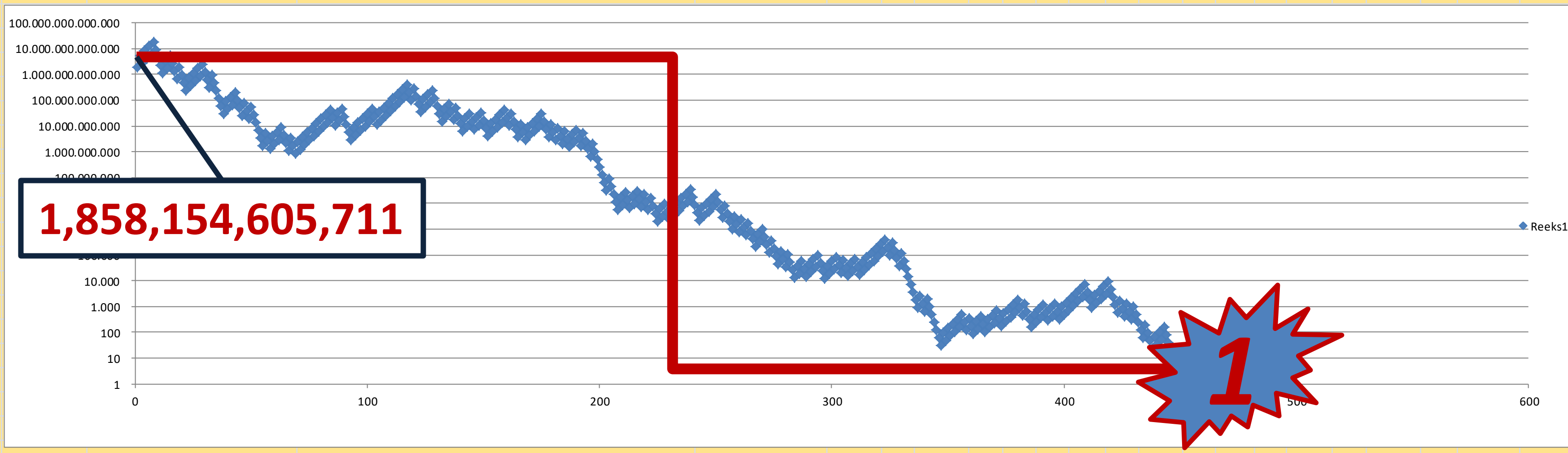
Why the Collatz string always end at 1

The 'Tail x Body + Head' approach
Caspar L.P.M. Pompe MSCe_200517

Number X_1: 1.858.154.605.711

10883

Number X(n+1) := Xn x A + a, in case Xn is Odd: A= 3
a = 1
Number X(n+1) := Xn / B in case Xn is Even: B= 2



Take a random number X_1. You may enter any number up to ten trillion in the box above.
When this number is Even, we divide it by 2 so we get number X_2.
When this number is Odd, we will multiply it by 3 and add 1 to the product to get X_2.
We repeat this procedure many times over. For example the string of numbers starting with 1,858,154,605,711 ends after 452 steps at $X_{453}=1$.
Why is this always the case – with any number?
This question is formulated by Mr Lothar Collatz in 1927!

Table of contents

Trajectory of this paper

This paper leads you through my research to crack the Collatz problem. First I show you a bit of my research to get on grips with this seemingly simple question. Collatz trajectories of all numbers under 128 have been analysed painstakingly. A simple question which proved hard to answer!

I discovered that the string of numbers behave like a bouncing ball on an increasingly sloping floor. In the end the ball finds itself in the deepest hole of Collatz space: the 1. Why does this ball one time jump upwards and then falls directly on the floor – and why does it another time jump to the left, while it is not certain if it will then hit the floor, or a bit later? This can all be explained by analysing the last two binary numbers in which the decimal numbers are translated. At a certain moment – back in 2017 - I devised the Tail x Body + Head method to write Collatz numbers in an alternative notation. This method is explained in two forms, the straight TxB+H and the Cohort method. The latter resulting in two colourfull graphs.

Finally two mechanisms are described by which the numbers on average decline in a finite process. Sometimes the string temporarily develops upwards, but this can be explained. It is also observed that at the end of an elaboration the string hits for example value 103. Then the string flares up to almost 10,000. But in the end the string falls down to 1. Why? You'll see on the last pages.

I hope the Collatz community likes this approach. Of course I have the impression that I have cracked the problem. But I gladly leave it to mathematicians to determine whether my TxB+H approach is really the path to the salvation of addicted Collatz researchers.

If you like to receive my Collatz calculator or wish to comment, please mail to pompe124@planet.nl.

Collatz addiction

Early 2017 I was grasped by the question of Lothar Collatz, who formulated in 1927 the 3.n+1 conjecture. As many mathematicians have warned; this is an addictive topic. Recently (september 2019) mathematician Terence Tao published his proof that is true ‘for ‘almost’ all numbers’. Tao states that the problem is not yet solved completely, but he has made a major advance in cracking the puzzle.

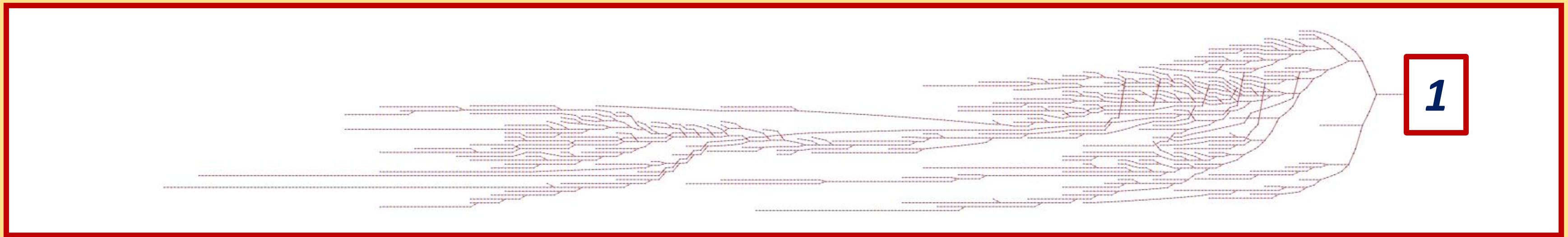


Figure 1. Trajectories of strings look like sea weed

Mathematicians like to study the behaviour of numbers. So, the Collatz string of number is often depicted in graphs like the one above. Many strings of numbers end at 1- the graph of those connected strings looks like seaweed. In the Netherlands this problem is sometimes referred to as ‘the hail-storm problem’, because the graph of the string of numbers looks like a hail storm.

After some time of literature research I came to grips with the problem. In the beginning I tried to approach the problem with several Modules. E.g., with a Module 4 one divides number X into pieces of 4. This approach is often encountered in publications. Later I shifted to a binary method, the TxB+H format.

I hope that this ‘Tail x Body + Head’ approach is usefull to the Collatz community.

Module 20?

I continued my research with modules 20, because the last 20 of the number determines – after division – the last digit of X_n. This exercise learned insight into the trajectories of strings of numbers up to 20. But it doesn't lead to a workable method towards proof.

	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
X.3+1	4		10				22		28		34		40		46		52		58	
X / 2		1		2		3						6						9		
X.3+1																				
X / 2	2		5	1			11		14		17	3	20		23		26		29	
X.3+1			16								52				70				88	
X / 2	1							1	7				10				13			
X.3+1																				
X / 2			8								26				35	1			44	
X.3+1															106					
X / 2			4								13								22	
X.3+1																				
X / 2			2												53				11	
X.3+1															160					
X / 2			1																	
X.3+1																				
X / 2															80					
X.3+1																				
X / 2															40					

Figure 2. Trajectories of strings 1 to 20

Numbers up to 128 all lead to 1

Numbers up to 128 have been worked out. Interesting to see that certain strings link to preceding ones, like in the sea-weed graphs. The string of 103 flares up to a value of 9232, before it recedes via 106 to 53 and 160 and further down to 10, 5, 16 and 1.

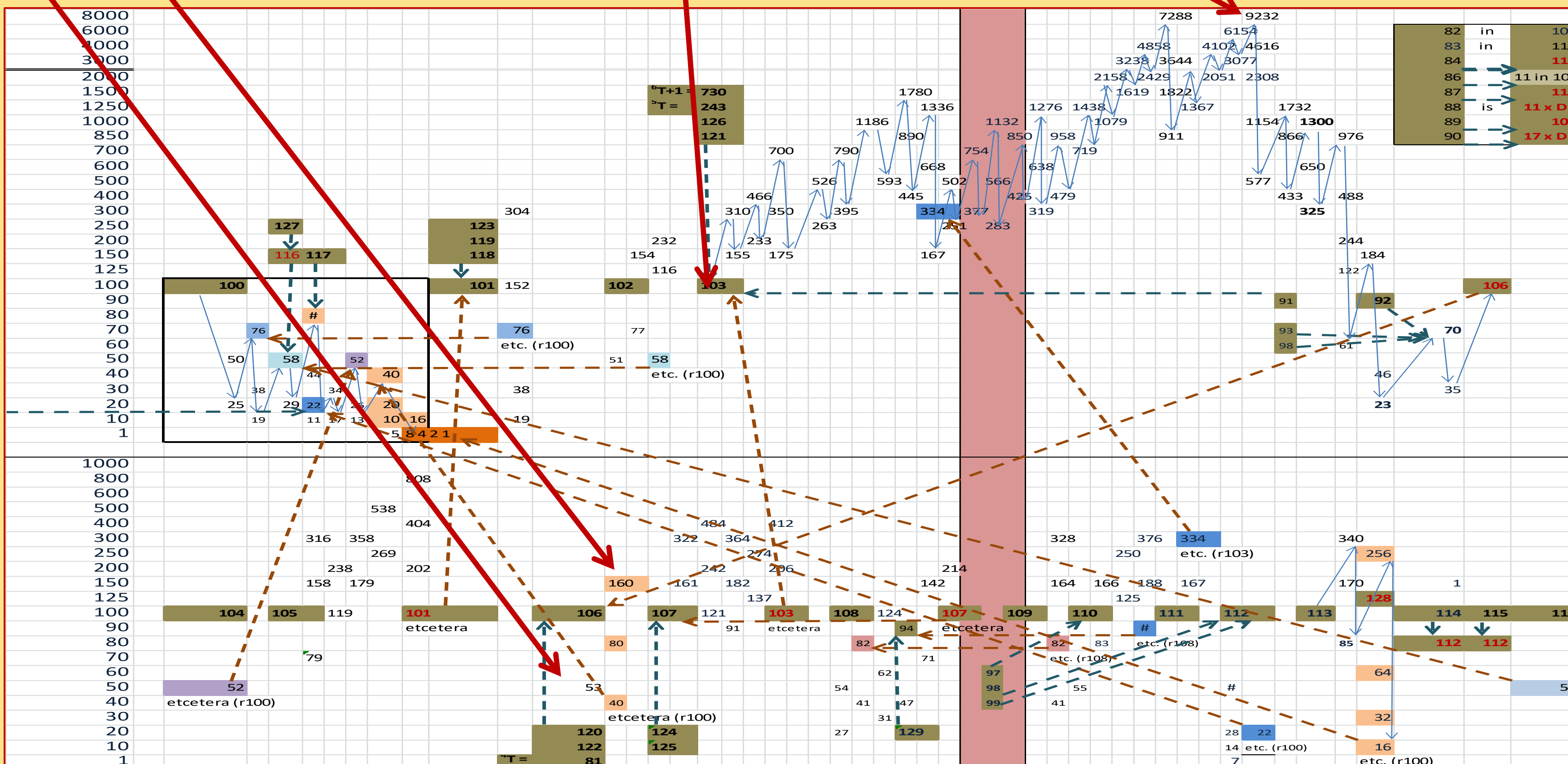


Figure 3. Trajectories of strings 1 to 128

It is good to notice that all numbers up to 128 = 2^7 lead to 1. We need this insight later on.

Logartihmic scale to straighten things up

My research continued with building a ‘Collatz calculator’. This calculator can tackle strings with a length of about 1,000 numbers. Which seems to be sufficient for numbers X_1 of the order of one trillion (10^{12}). Applying a logarithmic scale of the Y-axis of the graph, the dots of the numbers form series of straight line segments. With this calculator one can also study the behaviour of 5.x+2; 7.x-1 and other combinations.

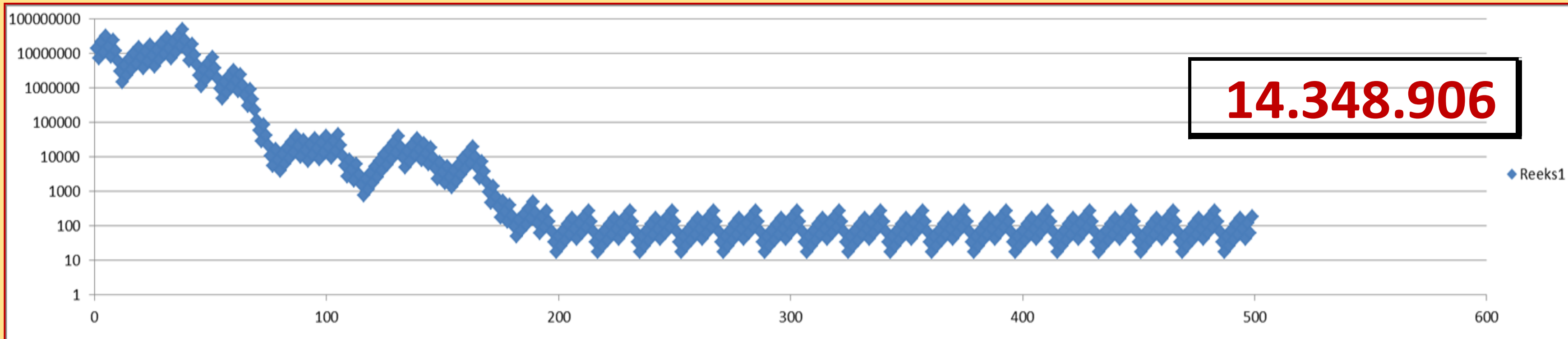


Figure 4. Graph of 3.x – 1 . Cycle at 7.

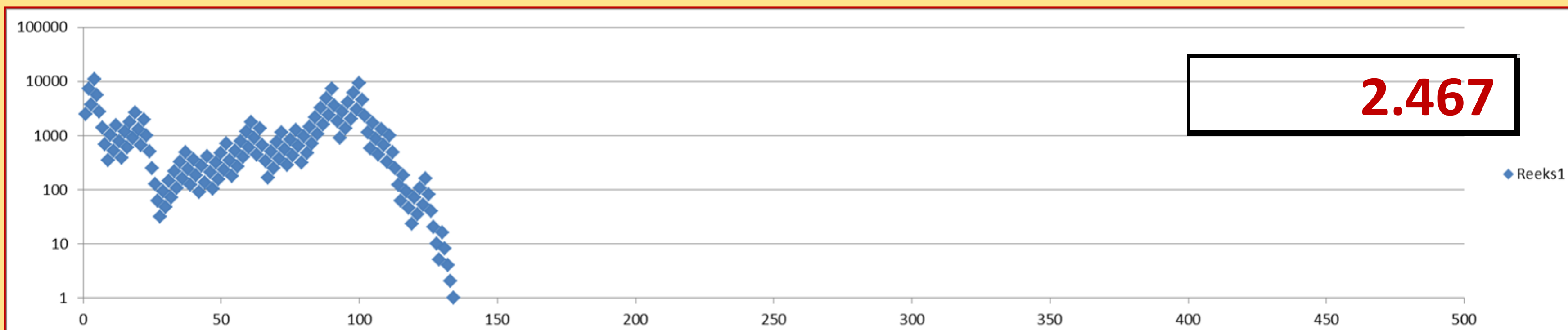


Figure 5. Graph of 3.x + 1 . String ends in a Cycle starting at 1…4-2-1-4-2-1

Last digits determine the beginning of the string

You will notice that the form of the end of String 2,467,775 looks similar to String 2,467. The beginning of the string seems to be determined by the last digits of the number.

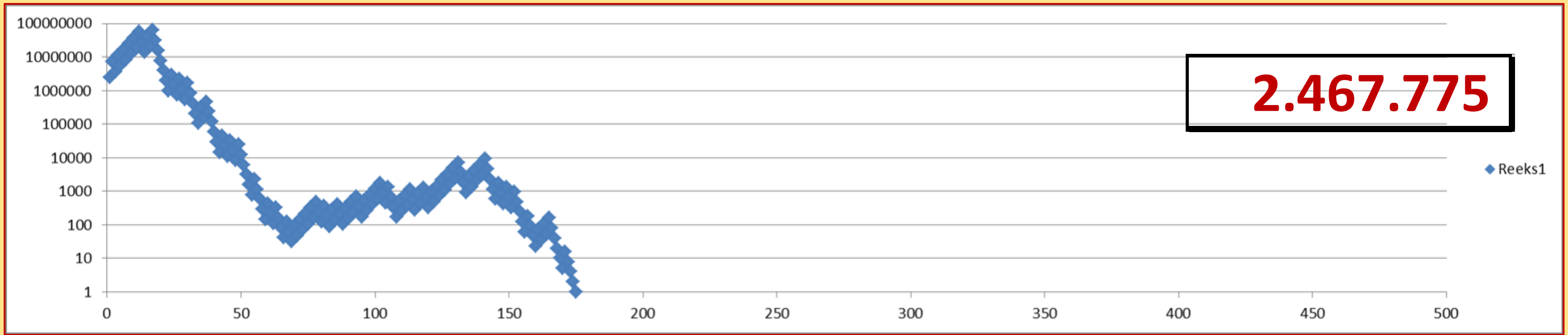


Figure 6. Last digits of number X seem to determine the form of the beginning of the string

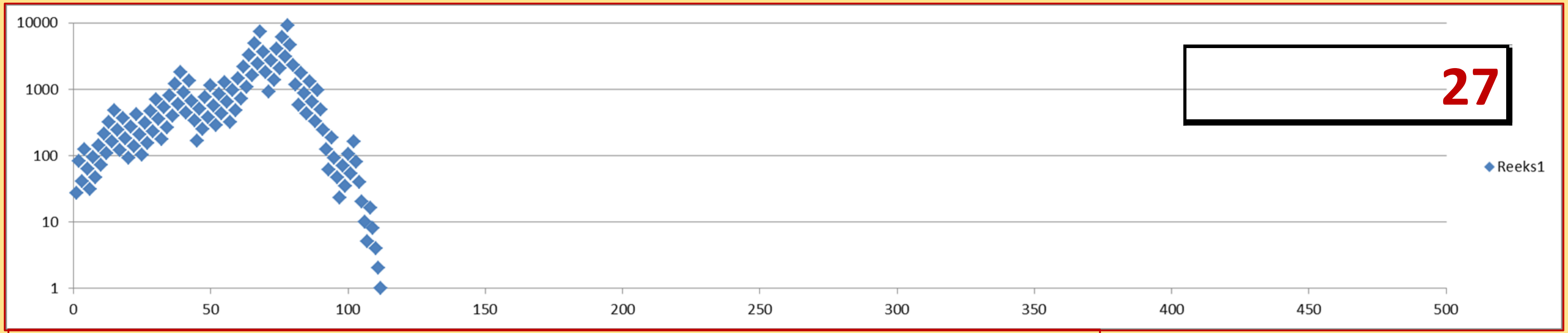


Figure 7. A low number such as 27 can result in a long string (112! numbers)

Numbers like 47 and 55 also result in an impressive development of the string. Often the ends of the graph look similar. Strings of numbers around 100 are linked to preceding ones. Hence the similarity of the ends of their graphs.

Numbers D, T, N and M

In my research I have tested many numbers. Looking for special numbers that result in long strings. I tried to construct a number that results in a never ending string. Alas to no avail!

Important numbers in this approach are the numbers D, T, N and M.
D are all numbers that are a power of 2 or [1000…00] such as 1, 2, 4, 64, 1024 (d=0,1,2,6 and 10).

$$\mathbf{D = 2^d}$$

The most anoyable number is [111…11] . In the process the string of this number increases continually. The number [111…11] is the result of [100..00] – [1]! This number is named N.

$$\mathbf{N = D - 1}.$$

T are all numbers that are a power of 3 such as 1,3,9, 273 (m = 0, 1, 2 and 5).

$$\mathbf{T = 3^m}$$

An important find of the TxB+H method is that when N is processed, the maximum result is 2 x (3^d-1). Since after multiplication the number is divided directly, we find that the result is 2 x 3^d-2 = 2 x (T -1) after (2xm) steps. The number of steps equals 2xm, because after a multiplication the product always is an Even number. Multiplication and division come in pairs. We reach the last number of the process after dividing 2 x (T -1). We name this number M. Thus the string has a length of (1 + 2xm+1) numbers.

$$\mathbf{M = T - 1}$$

On the next page you will see how it works out. And – luckily the digital [M] contains many [0]'s. We'll look into this phenomenon later on.

Graphs of N and N[0]N

A [0] in between substrings of N [111.11] influences the length of the string greatly.

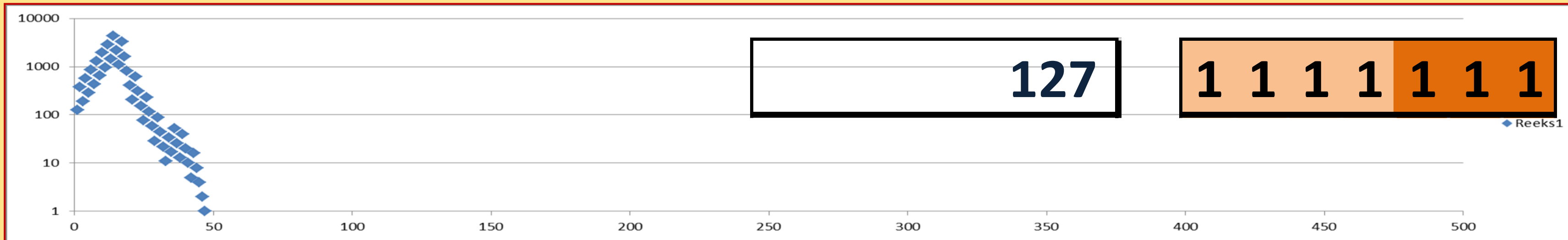


Figure 8. Elaboration of an unbroken string of [1]'s results in a continuous up-swing of the string

Insert a [0] in between the [1]'s and the up-swing resulting from [1111111] is broken after [111] is 'digested'.

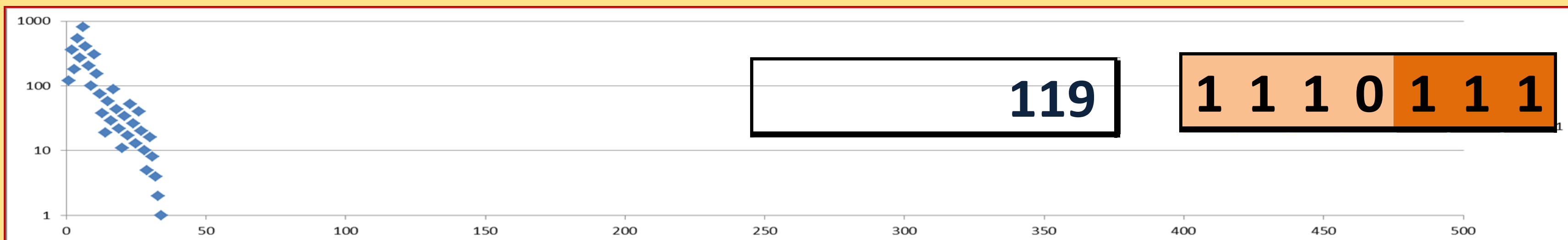


Figure 9. A [0] in between [1]'s shortens the string substantially

[10101010….0101] -> [10000…000]

Other interesting numbers are numbers like [101] (5). 3 x 5 + 1 = 16. and [1010101] = 85 → 256!
How come? 3 x 5 + 1 = 2 x 5 + 1 x 5 + 1 ….now look wat happens with the binary numbers!
The [1]'s first interlock and result in N. Then [1] is added and all [1]'s disappear and the number gets one digit longer to become D.

[1010101]
[10101010]
[11111111]
[1]
[100000000]

Binary Number Cruncher

Binary numbers can be ‘digested’ gradually. The numbers are ‘eaten’ away by the ‘Binary Number Crucher’ (BNC). Let’s take number 26 to demonstrate the BNC at work. In the process you’ll also see the basis of the Tail x Body + Head notation, that I will discuss next.

Total number		$T = 3^m$	Body	Body		Head		Head
26	divide	1	26	1 1 0 1 0				
13	mutate	1	13	1 1 0 1	BNC			
	multiply	1	12	1 1 0 0		1		1
40	divide	3	12	1 1 0 0		1 0 0	3 x 1 + 1 =	4
20	divide	3	6	1 1 0	BNC	1 0		2
10	divide	3	3	1 1	BNC	1		1
	mutate	3	2	1 0		1 0 0	3 x 1 + 1 =	4
5	muteren	3	1	1		1 0		2
erbij.	multiply	3	0	0	BNC	1 0 1	3 x 1 + 2 =	5
16	divide	1	16	1 0 0 0 0	BNC			
8	divide	1	8	1 0 0 0	BNC			
4	divide	1	4	1 0 0	BNC			
2	divide	1	2	1 0	BNC			
1		1	1	1	BNC			

Figure 10. Binary Number Cruncher at work

Number 26 equals [11010]. The last digit being a [0], the number is Even and is divided by 2. The effect is that the last digit [0] is ‘eaten’ by the BNC. The ‘Body’ gets shorter at each division! Then we got a problem. The last digit of the Body becomes a [1]. The solution is Mutation. The last [1] is shifted to the right, thus forming the beginning of a new ‘Head’ number. Note that the Body’s last digit is now [0]. Then we can multiply: 3 x Body + 3 x Head + 1. The Head turns from [1] to [100] (4). Both Body and Head are Even, so the whole TxB+H is Even. We can divide twice – eating away the last 2 [0]’s of both Body and Head. Then both Body and Head become Odd. 3 x [1] of the Body is shifted to the right. The Head turns into [100] again. We can now divide the numbers again. Both numbers get one digit shorter. The number mutates once more. The Body becomes [0], the Head becomes [101]. This Head turns into a new Body of 3 x [101] + 1 = [10000] (16) after multiplication ……8…4…2..**1.** Nice!

[1011] = 11

Tail x Body + Head format

You have just seen a simple form of the 'Tail x Body + Head' structure, that I have devised to tackle the Collatz problem. Any number can be written in a T x B + H format.
When we start with a primary number, T= 3^m and B is the initial number. The Head number is still 0. We have not yet executed a multiplication, so m=0 and T_1=1 (3^o).

Only the Head (H) placed on the right side is processed by 3 x H + 1 when TxB+H is Odd.
The m becomes 1. (T^1) := 3. The process: 1 x B_n + H_n := 3 x B_{n+1} + 3 x H_n + 1 = T^m x B_{n+1} + H_{n+1}.
When TxB+H is Odd, we have two options.

- Body is Odd → Mutate number by shifting T^mx[1] to the right and multiply.
- Body is Even and Head is Odd → multiply directly.

Note that 'm' is the counter of the multiplications, while 'n' is the counter of numbers in the string.

When TxB+H is Even, both TxB and H are divided by 2.
Both Body and Head lose a [0] at the end of the number.

So during the process we have four options:

1. Both Body and Head are Even, Both Body and Head lose a [0].
2. Body = Even, Head = Odd → T^m ;= $T^{(m+1)}$) and H_{n+1}:= 3xH_n +1.
3. Body = Odd, Head = Even → mutation: $B_{(mut)}$:= B-[1]; $H_{(mut)}$ = T^m + H_n→ multiplication: T^m ;= T^{m+1} ; The Head H_{n+1} becomes 3x $H_{(mut)}$ + 1 and the Body B_{n+1} is the mutated $B_{(mut)}$ which is Even.
4. Both Body and Head are Odd → T^m ;= T^{m+1}; mutation: $B_{(mut)}$:= B-[1]; and Head H_{n+1}:= H_n + T^m (Odd).

Yes, quite complicated. You easily make mistakes. A good programmer can certainly automate this process.

[1100] = 12

Several methods of working with Tail x Body + Head

Several formats of the TxB+H method can be applied. The special number N (see sheet Numbers D, T, N and M) has been elaborated with an ever increasing Head term. In that format mutations of the Body (T^mx[1]) are added to an increasingly big Head. In the end T^m x B = 0, because B has become [0]. The Head has become almost as big as T^m itself. This new Head becomes the next Body to be processed. This Body turns out to be 2 x (T^m-1) = 2 M. What a beautiful outcome!

One may also use a certain Module D, e.g. 128 = 2^7. In that case the initial number X is divided by D. The Body is (integer (X/128)) and the Head is the remaining number (H < 128). At the end game of the string one must reduce the Module to a smaller value. I have used 128 in the beginning of my research. It works fine, seems to be faster, but in the end the length of the string remains the same as with other formats of process notation.

Yet another way is that each time a T^m x [1] is transferred from Body to the right it forms a new Head. Then the structure changes a bit. The structure becomes T^m x B + T^o x B_o + T^p x B_p + T^q x B_q + H, in which only the Head is processed with 3H+1 at multiplication. The B's are former Heads that automatically shift to the left at the moment of a new mutation. A new Head will be created and it's predecessor becomes a new B. Such a new B can have two forms.

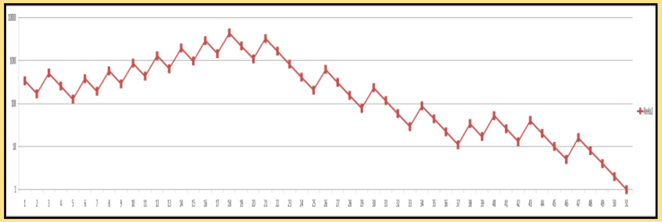
1. B_{new} can be the sum of an even number of shifted Tx[1]'s, which – as Head – has been divided one or more times until TxB+H has turned Odd.
2. B_{new} can be the sum of an odd number of shifted Tx[1]'s and as Head has been multiplied to (3xΣT+1) and subsequently be divided once or more.

Can you still follow me? Well, this is the Cohort method, which I will show next.

The Cohort method

You know that image of an army of angry Vikings that emerge on the crest of a hill? First one row of men, but gradually their front widens. And behind these warriors more rows of warriors emerge. I get this image when thinking of the TailxBody+Head approach. It starts with the Body – warriors in order of height from left to right (e.g. Number 83 = [1010011] = [1000000+10000+10 +1] – a front of 4 men wide. The tallest is 7 feet tall, the next is 5 feet etc.). At the first multiplication another two rows appear behind them. Same distribution of height. Thus forming a Cohort that increases in length at every multiplication. But – don't be afraid – something strange happens. The Cohorts are reduced in height at each division. At every mutation of an -Odd- Cohort it gets one man narrower! On the right a new front appears. But…Gradually all Cohorts disappear to…1!

338



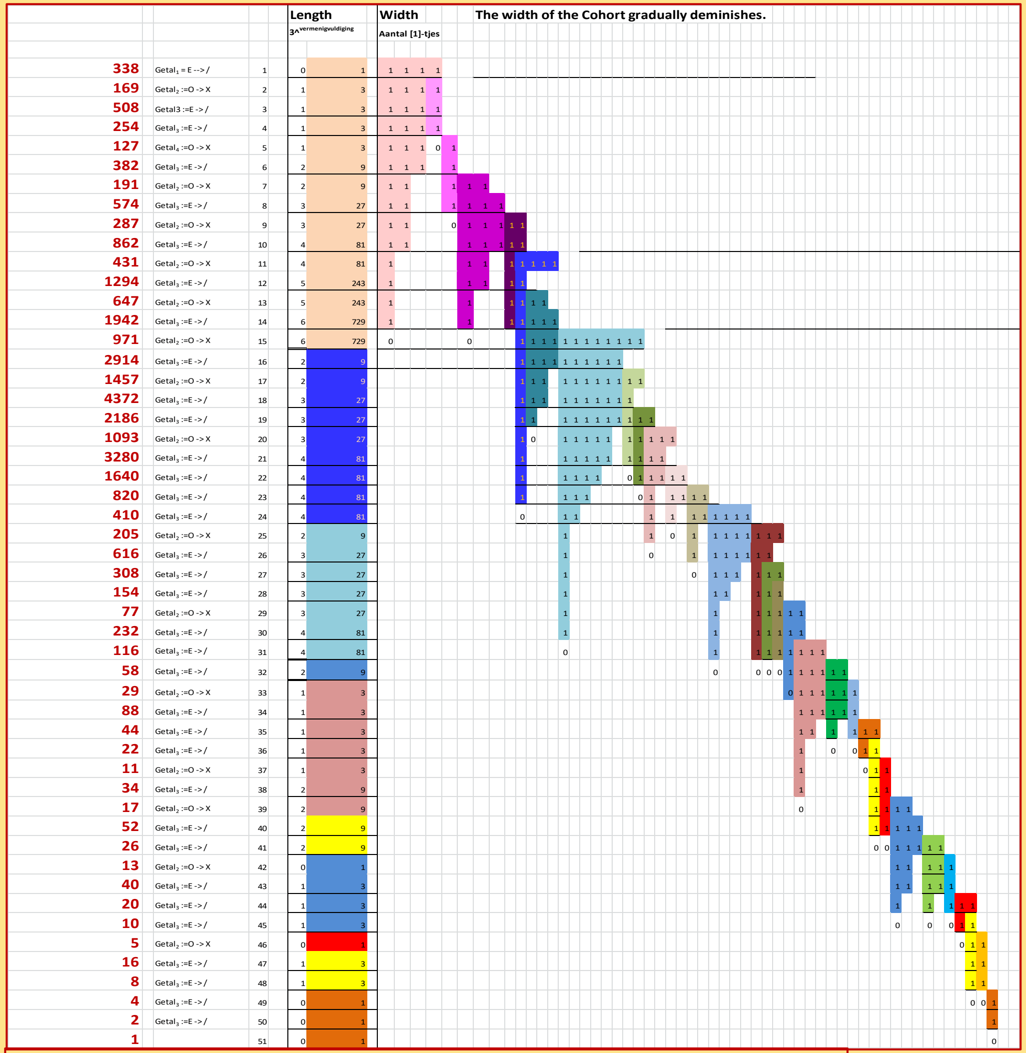


Figure 11. Inevitably the Cohorts are reduced to [1]

[1110] = 14

Cohorts inevitably shrink to [1]!

This image shows the ‘height’ of the Cohorts. The height of the Cohort equals the number of digits of the binary number. At each division the height gets lower. When you will see only one short warrior [1] on the right side of the Cohort – Beware! Behind this one warrior you’ll see T other warriors. Together they will form a new Cohort on the right side (the new Head). These (initially small) Cohorts will gradually shrink to [1]. These are (3xΣT+1) Cohorts or ΣT/D Cohorts. Eventually the primary Cohort has only one short warrior left. This [1] and his Tail forms a new secondary Cohort. That secondary Cohort will also shrink to [1]xT. At the same time other Cohorts have reached their last [1]xT’s. Together these [1]xT’s create a tertiairy Cohort (ΣT). But alas! Inevitably all Cohorts are reduced to the final [1]! We will look into these cohorts later.

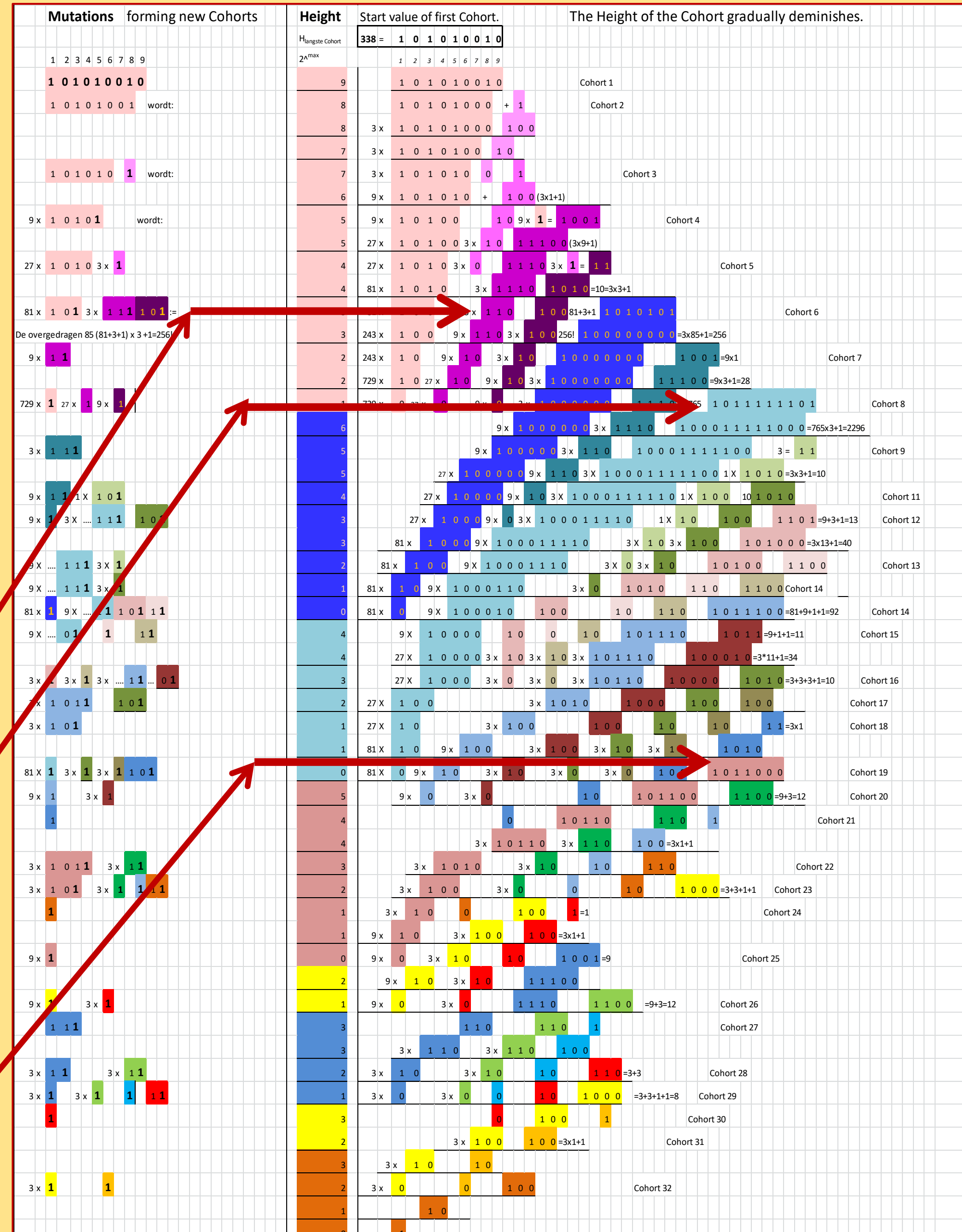


Figure 12. Inevitably the Cohorts are reduced to [1]

Behaviour of a Collatz string in space

The trajectory of a Collatz string resembles that of a ball that falls down on a skewed floor. It will bounce up and down to the right and to the left, but in the end the ball will tumble to the lowest point in the floor. Will this insight lead us to the proof of Collatz?

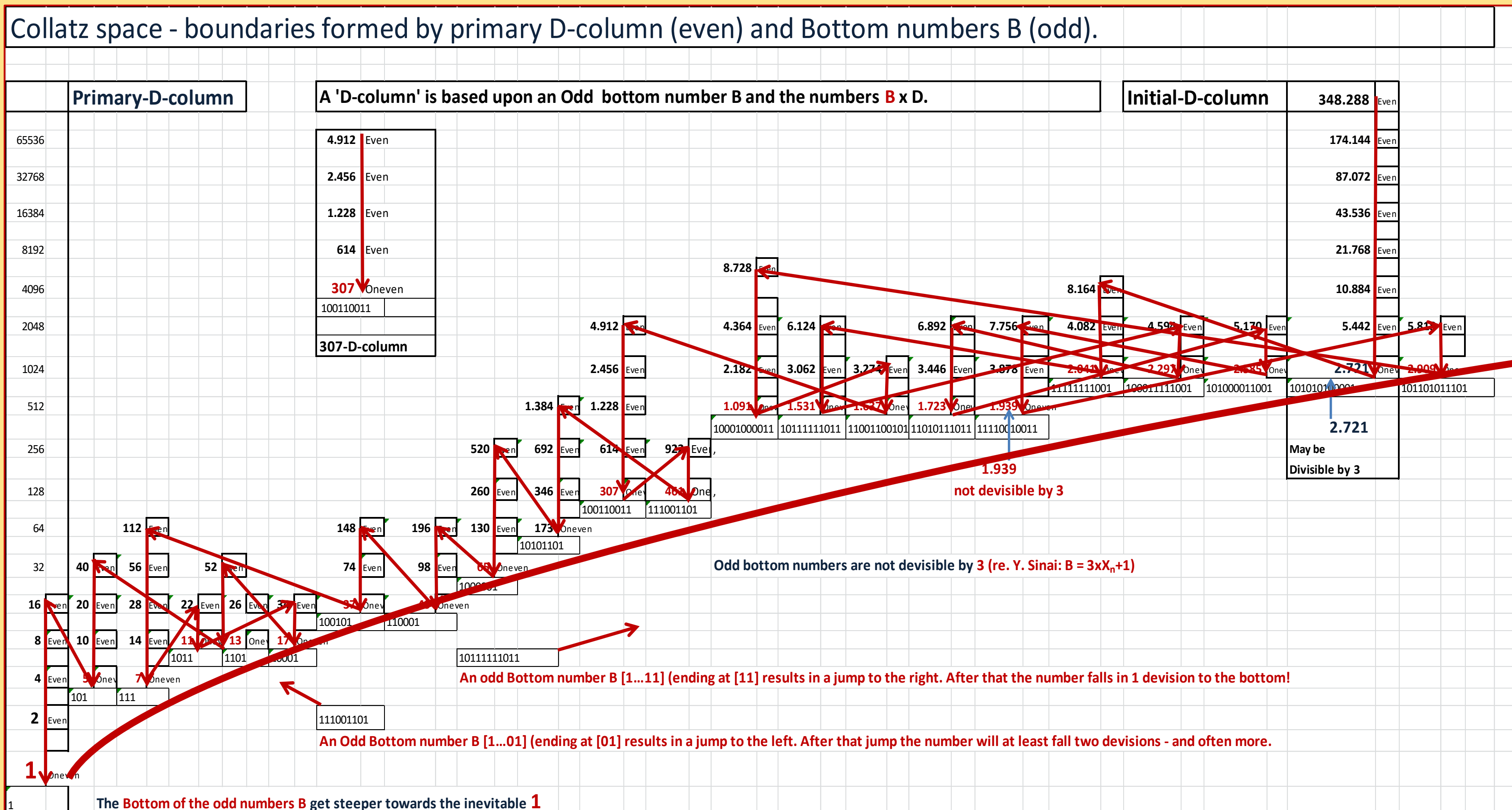


Figure 13. Inevitably the string of numbers bounces towards 1 – the lowest point in our Collatz space

Collatz space

In figure 13 we see that the trajectory of our string jumps up and down to the left and to the right. In the end the string ends at 1.

I will guide you through the graph.

The left axis of the Collatz space is the Primary D-Column, based at 1 containing D-numbers (2^n), such as 2, 4, 128, 1024. This is column 1xD.

On purpose I have used the start number 348,288 – not too big - This number is 2,721 x 128 (2^7). Thus, the string starts with 7 divisions. The string 'lands' on the Collatz floor at 2,721. Based on these odd numbers (red) C you will also find D-columns with numbers CxD.

If the primary number is Even, then it will take one or more divisions to reach the Collatz floor. These first Even number(s) form a D-Column based on an Odd number. That first Odd number can be a number that is divisable by 3. BUT…

Mr Y. Sinai indicates that the operation 3 x A + 1 dictates that a) the resulting number B is always Even and b) the next operation B/D always results in a number C that can never be divided by 3. So – apart from possibly the base number of the first D-Column – the bottom of the Collatz space consists of numbers 3 x X $\pm$ 1. Consequently – apart from numbers in the first D-Column – Collatz numbers consist of 2/3 of all numbers.

Is this a key to the solution? Do these C-numbers have special properties? We'll see later.

Prediction of the length of a string

Before diving deeper into these C-numbers, we wish to know whether we can predict the length of sub-strings that follow a certain number.
When the number X is a power of 2, e.g. [100000] = 2^5 = 32, it is easy to predict the length of the string of numbers. We just divide the principal number d (5) times to get to 1. String length = 1+5=6.
We learned that D-1 = N = [111…111] produces a sub string with a length of 2 x m (m = number of [1]'s in the number) to get to the maximum number 2 x M. String length to M = 1+2xm+1.

Can we predict the length of a string of <u>any</u> binary number? Theoretically Yes.
[1] leads to multiplication. And [0] leads to division. Multiplication is always followed by division.
So, at first sight a number produces a string length of 1 + 2xm + d. With evenly distributed [1]'s and [0]'s this rule of thumb provides a reasonable indication of string length.
But a more precise prediction will be quite complicated.

I will explain why with number [10000000001](1025) or D+1 (2^{10}+1).
Rule of thumb: Length of substring 1 = L_{ss1} = 1+2x2+9=14. But total length amounts to L_{tot} = 37!
In the TxB+H process the first [1] is shifted to the right to become a first Head. The Head follows the cycle of (1)[1]>(4)[100]>(2)[10]>(1)[1]. Every time it reaches [1] we'll have a multiplication. $T^m := T^{m+1}$

[1]-[100]-[10]-[1] → H_4 := Odd → multiply → T=3; [1]-[100]-[10]-[1] → H_7 := Odd → multiply → T=9;
[1]-[100]-[10]-[1] → H_{10} := Odd → multiply → T=27; [1]-[100]-[10]-[1] → H_{13} := Odd → multiply → T=81;
[1]-[100]-[10]-[1] → H_{16} := Odd → multiply → T=243 x[1]=Body and Head=[1] → Number 16 =244.

The primary string is only 2 numbers longer than predicted. This string needs another 20 steps before it reaches [1].

Prediction of the length of a string (cont'd)

We have seen with this example that a Head that is processed until [1] influences the long row of divisions of the Body – that has mutated to [10000000000] in the beginning of the process. The repeated [1][4][2][1] subcycle makes that T increases to T^5=243. So when the Body reaches its [1], it turns out that the Head is also just at [1] again. Together they make 244 as X_{16}.

So, the processes of the (mostly shortlived) Heads interfere with the process of TxB.

After many exercises I learned that when a row of [000] occurs, an additional multiplication may be assumed. But [11]'s may lead to annihilation of multiplications. All rules of thumb!
One could mutate when hitting a sub row of [111…111]. For example [11001101]=
TxB[11000000]+TxB[1100]+H[1] after 3 steps (mdd) := 3xB[110000]+3xB[11]+H[1]
Here annihilation of a multiplication occurs because 3xB[11] and H[1] are both Even.
After mutation the number becomes 3xB[110000]+3xB[10]+H[100] etcetera……
By hand it turns out to be too complicated to make a precise prediction.

In practice it turns out that if one counts the [000] groups as additional multiplations+divisions and [111] as less divisions, one can make a quite good prediction of the number of steps it takes to process a long number to next long number. This next number – by the way – will have the form of TxB[1]+H.

Take an random number [11101110000101001110000101001011011110101].
It contains 22 [1]'s, 19 [0]'s, 2 [00]'s and 3 [111]'s. Step prediction: 1+2x22+19+2x2 -3=65.
The next number will be $X_{n+1} \sim T/D$ in which the first [1] of the number does not lead to a multiplication. This is the end-[1]. $X_{n+1} = 3^{(22-1+2)}/2^{(22+2+19-3)} X_n = 3^{24}/2^{40}$ $X_{n+1} = 0.086\, X_n$

Rolling a dice

We write the number A as [Xxx]. In this form [X] is the binary number minus the two last digits. [xx] are the last two digits.
[xx] can be either
[00] – 2 x down in Collatz space;
[01] – jump to the left and then at least 2 x down;
[10] – 1 x down mutate and multiply
[11] – jump to the right and then 1 down to the floor.
In fact, this a module 4 approach. Each of the possible [Xxx]'s has a probability of occurance of 25%.

The high factor of 2,25 of [X11] balance the low factors of the other three X's. The total factor is 1.

So, this probability approach does not lead to a solution.
That means that the inevitable descent must have another mechanism.

25%	**A = X 0 0**		X 0 0 + 1 0 0
Factor	A = 4 . X		X 0 + 1 0
0,25	B ~ 1 . X	B =	X + 1
25%	**A = X 0 1** ;=		X 0 0 + 1
	A ~ 4 . X		3 . X 0 0 + 1 0 0
Factor	for large numbers X		3 . X 0 + 1 0
0,75	B ~ 3 . X	B =	3 . X + 1
25%	**A = X 1 0**		X 1 0
	A ~ 4 . X		X 1
			X 0 + 1
Factor	for large numbers X		3 . X 0 + 1 0 0
0,75	B ~ 3 . X	B =	3 . X + 1 0
25,0%	**A = X 1 1** ;=		X 0 0 + 1 1
	A ~ 4 . X		3 . X 0 0 + 1 0 1 0
			3 . X 0 + 1 0 1
Factor	for large numbers X		9 . X 0 + 1 0 0 0 0
2,25	B ~ 9 . X	B =	9 . X + 1 0 0 0
Total factor			
1,00			

Figure 14 General probabilistic approach does not help

Average Ascent/Descent Rate (ADR) first 59 [3n+1]'s = 0.41

Note that the floor of the Collatz space consists of odd number not divisible by 3.
T and M numbers contain many [0]'s.

So I studied the Collatz-floor numbers.
The next number is always 3xC+1. Similar to T and M I expected these numbers to have a high [0] content. On sheet 'Prediction of the length of a string' I explained how the number of devisions 'd' and multiplications 'm' can be estimated. The ascent or decline of a number will be about $T_m/D_{(m+d)}$.

For the first 59 3xC+1 numbers the average descent is about **0,41** with a downward trend.
Note that right before a step in the order of D the string may make a steep ascent.

C_{28}=83; 3xC+1= 250; T/D= 1,9

The decline is strongest when Cn is just past a step in the order of D.

C_{29}=85; 3xC+1= 256 = D^8!

The next C, the descent rate is << 1.

C_{30}=86; 3xC+1= 262; T/D= 0,035

									0,41
C_n	$3xC_n$+1		digits	[1]	[0]	[00]	m	d	T/D
1	4	1 0 0	3	1	2		0	2	0,25
5	16	1 0 0 0 0	5	1	4		0	4	0,0625
7	22	1 0 1 1 0	5	3	2		2	4	0,5625
11	34	1 0 0 0 1 0	6	2	4		1	5	0,09375
13	40	1 0 1 0 0 0	6	2	4		1	5	0,09375
17	52	1 1 0 1 0 0	6	3	3		2	5	0,28125
19	58	1 1 1 0 1 0	6	4	2		3	5	0,84375
23	70	1 0 0 0 1 1 0	7	3	4		2	6	0,140625
25	76	1 0 0 1 1 0 0	7	3	4		2	6	0,140625
29	88	1 0 1 1 0 0 0	7	3	4		2	6	0,140625
31	94	1 0 1 1 1 1 0	7	5	2		4	6	1,265625
35	106	1 1 0 1 0 1 0	7	4	3		3	6	0,421875
37	112	1 1 1 0 0 0 0	7	3	4		2	6	0,140625
41	124	1 1 1 1 1 0 0	7	5	2		4	6	1,265625
43	130	1 0 0 0 0 1 0	7	2	5		1	6	0,046875
47	142	1 0 0 0 1 1 1 0	8	4	4		3	7	0,2109375
49	148	1 0 0 1 0 1 0 0	8	3	5		2	7	0,0703125
53	160	1 0 1 0 0 0 0 0	8	2	6		1	7	0,0234375
55	166	1 0 1 0 0 1 1 0	8	4	4		3	7	0,2109375
59	178	1 0 1 1 0 0 1 0	8	4	4		3	7	0,2109375
61	184	1 0 1 1 1 0 0 0	8	4	4		3	7	0,2109375
65	196	1 1 0 0 0 1 0 0	8	3	5		2	7	0,0703125
67	202	1 1 0 0 1 0 1 0	8	4	4		3	7	0,2109375
71	214	1 1 0 1 0 1 1 0	8	5	3		4	7	0,6328125
73	220	1 1 0 1 1 1 0 0	8	5	3		4	7	0,6328125
77	232	1 1 1 0 1 0 0 0	8	4	4		3	7	0,2109375
79	238	1 1 1 0 1 1 1 0	8	6	2		5	7	1,8984375
83	250	1 1 1 1 1 0 1 0	8	6	2		5	7	1,8984375
85	256	1 0 0 0 0 0 0 0 0	9	1	8		0	8	0,0039063
163	490	1 1 1 1 0 1 0 1 0	9	6	3		5	8	0,9492188
167	502	1 1 1 1 1 0 1 1 0	9	7	2		6	8	2,8476563
169	508	1 1 1 1 1 1 1 0 0	9	7	2		6	8	2,8476563
173	520	1 0 0 0 0 0 1 0 0 0	10	2	8		1	9	0,0058594

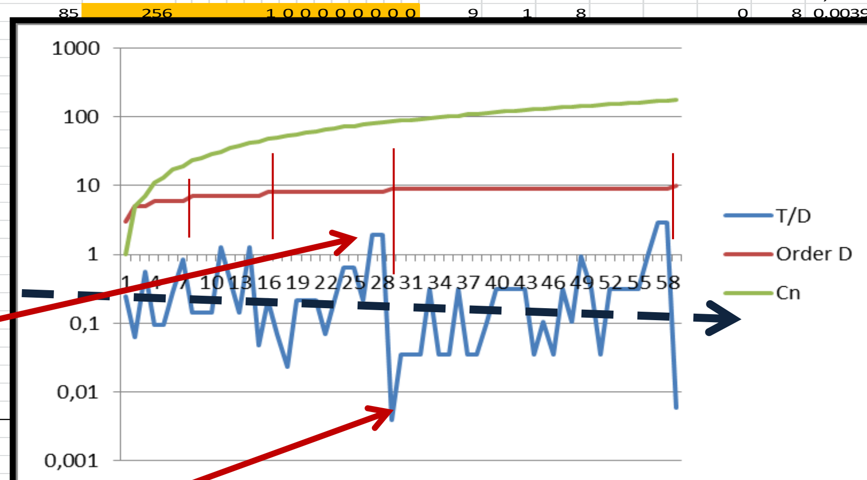


Figure 15. Average descent rate of strings starting at 3n+1 numbers (first 59 numbers)

What next?

We have seen that the probability approach does not result in a general rate of decline by eating away in steps of the last two digits. The descent/ascent Rate is determined by another mechanism.

Let's sit back a bit to determine next steps of research. What happens during the Collatz process? We start with an certain primary number. This primary number is steadily crunched away by the Binary Number Cruncher (BNC). Each time the BNC meets a [1], it is places this [1] times its tail T on the right side to form a new Cohort (mutation). The width of the mutated Body and Cohort numbers decreases. The numbers become Even. At multiplication the Tail T^n of Body and Cohorts increase to T^{n+1}. At division the heighth of the Cohorts decrease one order of D (the last [0] is crunched away). Thus, Body number and all Cohorts gradually but surely become T x [1].
So, the primary number becomes the secondary number, which is Tx[1] -1. [1] may be added to the Head or it may be used to add to the formation of a new Cohort. This secondary number eventually turns into a tertiairy number Tx[1]. Etcetera until T x [1] = 1.

We have learned that a binary number X is crunched away by the BNC in about 2x(m+d) steps (in which m = number of [1]'s and d = number of [0]'s). Consequently $X_{II} := T^m/D^{m+d} \times X_I$. This T^m/D^{m+d} is in fact the rate of decline/ascent (ADR) of the sub-string in which X_I is being processed into X_{II}. On page 21 we found the ADR to be increasingly smaller than 1 (0.41) for the first 59 C_n's. This is important. We must now dig deeper into ADR to find general values.

Let's study some more values of ADR of numbers of different characteristics.

Special types of numbers to study

1. Average ADR at increasing order of D
   ADR gradually decreases. $ADR_{average}$ is ADR of a number that has an equal number of [1]'s and [0]'s. $ADR_{average} = T^{1/2OD}/D^{(1/2OD+1/2OD)}$

2. The initial substring of extreme primary number N = D-1 ascends to 2T-2 and then becomes M=T-1. ADR depends on the distribution of [1]'s and [0]'s of M numbers.

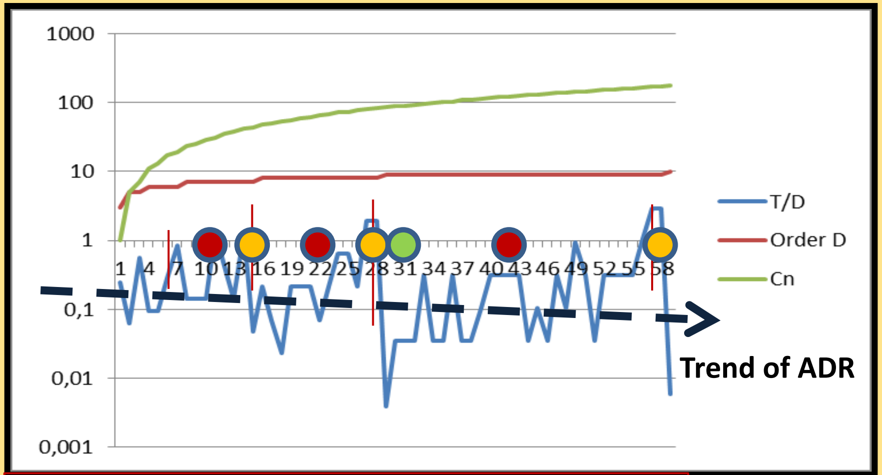


Figure 16 Types of numbers to be studied

3. When Cn is just a bit more than an Order of D, the number of [0]'s is high. ADR will be very low.

4. In general the primary number has the form of Cn x D = [X1] x D.
   [X1] is and odd number. So it may be divisible by 3. But once the string has reached the Collatz floor at $X1_{odd}$, the Collatz-floor numbers C_n will never be divisible by 3 anymore (re. Y Sinai).

Average ADR

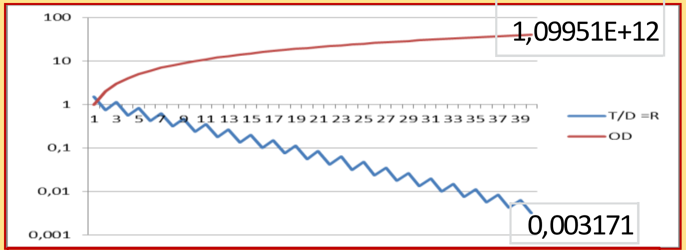


Figure 17. Average descent/ascent Rate

Suppose that numbers in between orders of D have about equal amounts of [1] and [0]'s. At increasing order of D, ADR gradually decreases. The graph shows that $ADR_{average}$ amounts to about 0.003 for a number of about 1 x 10^{12} (1 trillion) and 40 binary digits. So $X_{II[32]} := 0.003.X_{I[40]}$: Number X_{II} has about 8 binary digits less than X_{I}. The first substring has 1+2x20+20=61 numbers. Next substring, ADR is about 0,01 so next number becomes 7 binary digits shorter. $X_{III[25]} := 0.01.X_{II[32]}$. The second substring has about 2x16 + 16=48 numbers. Next $X_{IV[21]} := 0.05.X_{III[25]}$ has about 4 to 5 digits less. The third Substring has 2x12+13=38 numbers; Fourth substring: $X_{V[17]} := 0.08.X_{IV[21]}$ with 2x8+9=25 numbers;
$X_{VI[13]} := 0.15.X_{V[17]}$ – 2x6+7=19 numbers; $X_{VI[11]} := 0.3.X_{V[13]}$ – 2x6+7=19 numbers;
$X_{VII[9]} := 0.4.X_{VI[11]}$ – 2x6+7=19 numbers; $X_{VII[7]} := 0.5.X_{VII[9]}$ – 2x3+4=10 numbers but now we enter into numbers of less than 256. As we have seen earlier, these may flare up in the end. So far we have about 201 numbers. Say that the last sub-strings add another 50 to the string. This number of about 1 trillion results in a string of about 250 numbers. I have entered a number about that size. We see that after the first substring of about 60 numbers, the string flares up. It turns out that the secondary

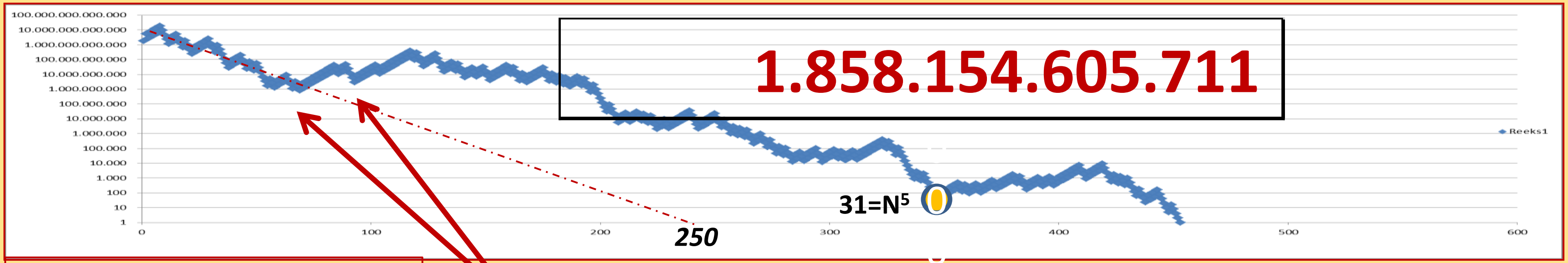


Figure 18. Random example

number contains many [1]'s in two long rows. Then it hits $31=N^5$ and the string needs more than 100 steps to end in [1].

ADR of $R_{N \to M}$

What happens when we have a number consisting of [1]'s solely (=N)?
Maximum Ascent! Such a primary number results in a steep ascent towards the secondary number M. Apart from M_3, M_5 and M_{11} = 1.594.322 R_M is increasingly smaller than 1.
The graph of N^{30} shows that the string descends evenly after M^{30}. But string N^{11} ascends further after M^{11}=[10101100111111010]. You'll understand why. It hits 127=N7 again before going to [1].

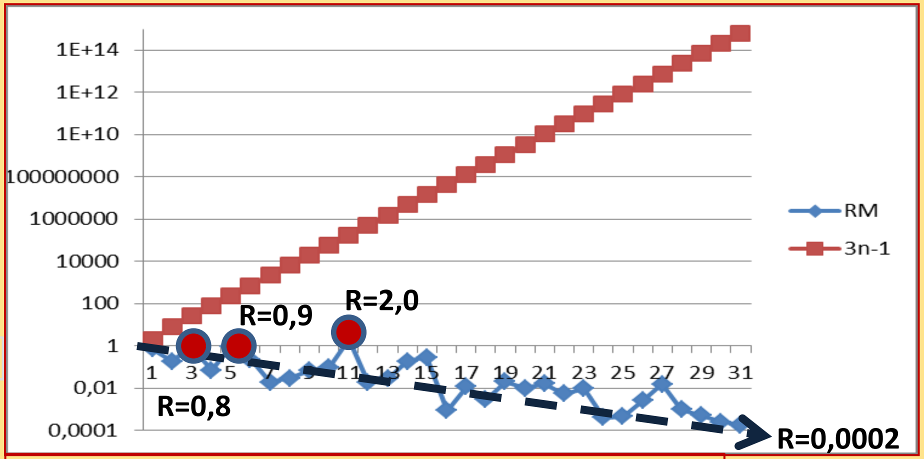
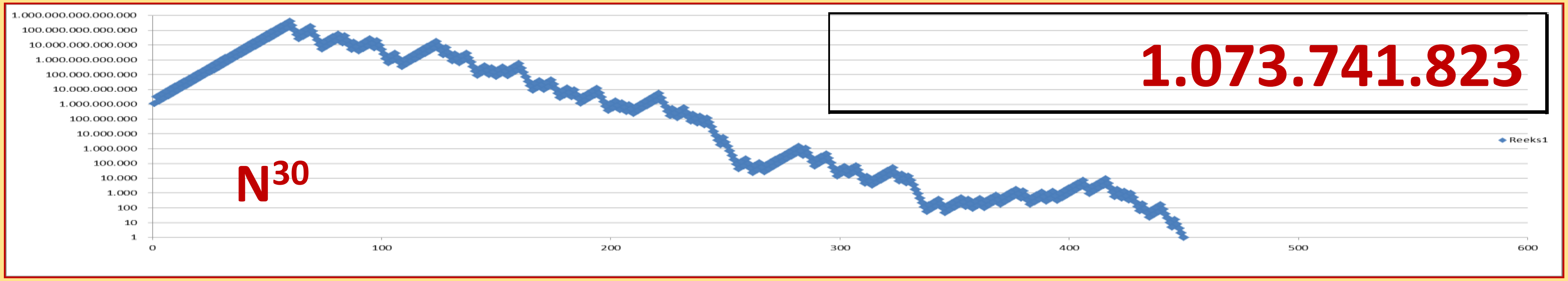


Figure 19. Descent/ascent Rate of M-strings



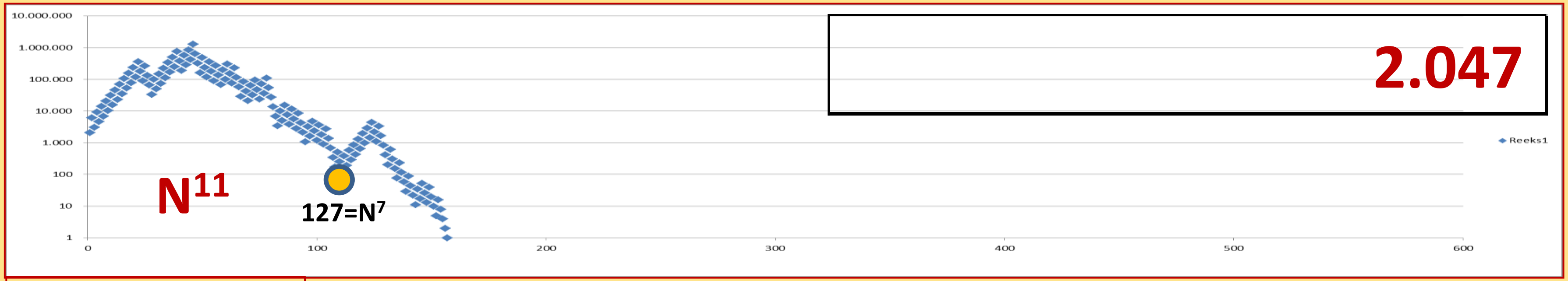


Figure 20. N-strings

ADR of R_{D30+3}

And what happens when we have a number consisting of many [0]'s?
Let's look at D^x+3. The last binary digits are [11]. $D^{30}+3$ = 1,073,741,827. We'll place this [11] in The Head. In the graph you see first a small ascent, stemming from [11], which is N^2. Hence, the maximum is reached after 2x2 steps = $\underline{2^{(30-2)} + 2.3^2-2}$. One division leads to M = 8 = [1000]. Then, the remaining row of 27 [0]'s is eaten away by the BNC. But, every 3 steps we hit a [1] of the Head cycle [100] – [10] – [1] – [100] - etcetera.

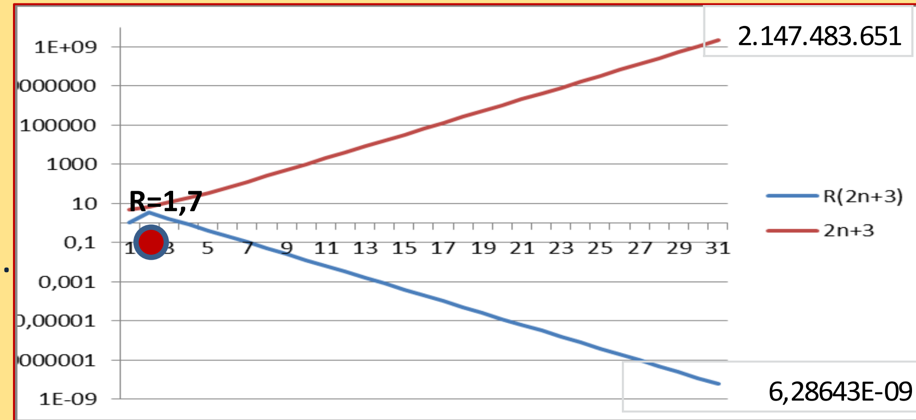


Figure 21. Descent/ascent Rate of D+3-numbers

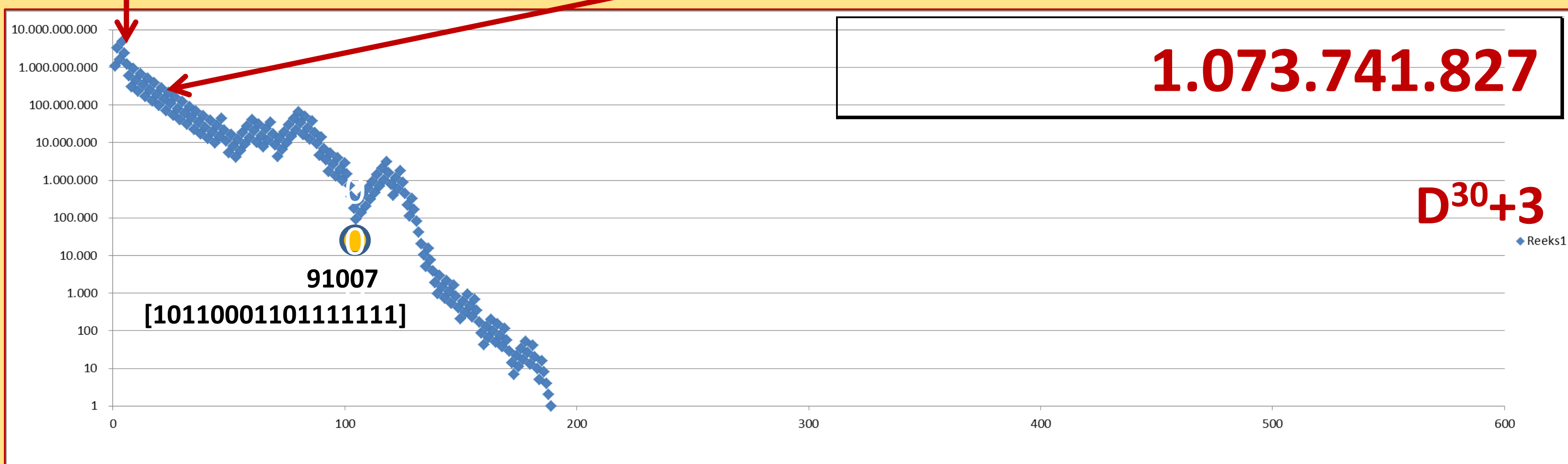


Figure 22. D^{30}+3-string

So, after 2x2+1+27/2+26=45 steps the primary number is processed with the result $3^{(2+13)}+2$ = 14348909 (number 46). Later, the string hits 91007, which has N^7 at the end of the binary number. This makes the string to flare-up in 2x7 steps before it plunges down to the inevitable [1].

Primary Collatz-floor numbers divisible by 3

The last type of number to be studied are the 'Primary Collatz-floor numbers' that a string may hit after a primary number is devided once or more and turns out to be divisible by 3. Later in the string that will never happen again. The 3n+1-value of all those numbers, starting at 1-3-9-15-21-27 is – of course - Even. The string jumps to the right and to the left. Just like all other numbers.

On the very last page the number smaller than 100,000,000 with the longest string is shown. The string has 952 numbers. The initial number is divisible by 3. Hence, it is a Primary Collatz-floor number. That means that only a number D^y x P has an amount y more numbers in the string. It turns out that such a number D^y x P can never be hit by a string not starting with a number $D^{(y+z)}$ x P.

C_n	$3xC_{n3}+1$	$(3xC_{n3}+1)/2$			
1	4	2	1		
3	10	5			
9	28	14	7		
15	46	23			
21	64	32	16	8	4
27	82	41			
33	100	50	25		
39	118	59			
45	136	68	34	17	
51	154	77			
57	172	86	43		
63	190	95			
69	208	104	52	26	13
75	226	113			
81	244	122	61		
87	262	131			
93	280	140	70	35	
99	298	149			
105	316	158	79		
111	334	167			
117	352	176	88	44	22
123	370	185			
129	388	194	97		
135	406	203			
141	424	212	106	53	
147	442	221			
153	460	230	115		
159	478	239			
165	496	248	124	62	31
171	514	257			
177	532	266	133		
183	550	275			
189	568	284	142	71	
195	586	293			
201	604	302	151		
207	622	311			
213	640	320	160	80	40
219	658	329			
225	676	338	169		

Figure 23. Primary Collatz-floor numbers

Mutation, Multiplication and Division

Now we nearly reach the end of the research – with a recap of the TxB+H method.

- The TxB+H method writes any number as a power of 3 - T – times a (binary) 'body number' plus a (binary) Head number. The Body is only mulitplied by 3, while the Head gets the full Collatz protocol. Example: Body [1100100] 100 + Head [101] 5 → 3x[1100100] + (3x5+1) [1010] 10
- TxB may consist of $D.TxB_1$+$D.TxB_2$ +….$D.TxB_n$, in which the T's are of decreasing power of 3. D may be 1 or 2. Smaller B's are called 'Cohorts'. Body and Cohorts must always be Even before multiplication and division.

Hence, the TxB+H method knows three basic mechanisms:

- Mutation The Body (or one or more Cohorts) may become Odd. Then the last [1] of the binary number times the actual T of the Body (or the Cohorts) is transferred to the right. The transferred Tx[1]'s are added to the Head's actual value or a new Head is formed. In the latter case the former Head becomes a new Cohort. Or new Cohort(s) are created, while the Head is kept as small as possible. At each mutation Body/Cohort number becomes Even.
- Multiplication When the Head becomes Odd, the Head turns into 3 x Head plus 1. All T's on the Body side shift 1 power of 3 up. The highest value of T indicates the number of multiplications in the process. $T^m := T^{m+1}$
- Division When the Head becomes Even, the last [0]'s of the binary numbers of both Body (or Cohorts) and Head disappear – all numbers are divided by 2. At each division the length of binary Body and Head numbers decreases with one digit.

Thus, the TxB+H method gradually 'crunches' away any Body or Cohort number to [1], while at each multiplication the value of T increases threefold and at each division binary numbers lose one digit. Let's have a deeper look at the process of mutation.

Multiplication, division and mutation (cont'd)

When in Cohort protocol a mutation occurs, one or more Cohorts that are reduced to Tx[1] are used to create a new Cohort, or they may be added to the Head.

We have six options, while we start with a primary number P. In the process Cohorts are created. Hence, Body numbers consist of P and Cohorts:

1. **Total number X = Even:** All Body numbers are Even and the Head is Even. Next step is a division by which all numbers decrease 1 binary digit in length. T's of Body numbers remain the same.

2. **Total number X = Odd:** All Body numbers are Even and the Head is Odd. T's of Body numbers all shift one power up. The length of Body numbers remains the same. The Head is multiplied and becomes Even. The next step is a division by which all numbers decrease 1 binary digit in length.

3. **Total number X = Even:** An Even number of Body numbers are Odd and the Head is Even. Body numbers split off their last [1]xT_{actual}. The last digit of these Body numbers become [0]. An even number of Tx[1]'s forms an Even Cohort number (Ta + …+ Tx). The smallest of these pairs may form a new Head while the former Head becomes a Cohort. Then all number are divided, hence all numbers lose their last [0]. T's of Body numbers remain the same.

4. **Total number X = Odd:** An Odd number of Body numbers is Odd and the Head is Even. First we'll mutate Odd Body numbers. An even amount of Tx[1] become a new Even Cohort number. The smallest singlet Tx[1] may become the new Head – which is an Odd number, while the Head becomes a Cohort. Then we multiply. T's are shifted one up. The total number becomes Even. Then a division follows in which all numbers decrease in length – losing their last [0].

Mutation, Multiplication and Division and (cont'd)

5. **Total number X = Odd:** An Even number of Body numbers are Odd and the Head is Odd. These Body numbers split off their last [1]xT_{actual}. The last digit of these Body numbers become [0]. The even number of Tx[1]'s forms an Even Cohort number (Ta + …+ Tx). The Head is processed by 3n+1. T's of Body numbers are shifted one up.

6. **Total number X = Even:** An Odd number of Body numbers is Odd and the Head is Odd. First we'll mutate Odd Body numbers. The even number of Tx[1]'s forms an Even Cohort number (Ta + …+ Tx). The smallest Tx[1] is added to the Head, that becomes Even as well. Then we divide all numbers. All numbers decrease in length – losing their last [0].

What happens to P in this process? It is finally reduced to T^{max}x[1]. The maximum power of T is dictated by the number of multiplications in the process to reduce P to [1]. T^{max}x[1] becomes the secondary Body number S, leading the pack of what remains of the Cohorts. Initially S is Odd. We'll split off [1] to become a new Head. Maybe some Cohorts are also turned Odd. We split off their Tx[1]'s as well. These will remain in the Body section of the number. When a singlet Tx[1] is transferred (the smallest one), it is added to the [1] split off from S. In that case the Head becomes 0.

All Body numbers gradually but surely reduce to Tx[1]. When we keep the Head as small as possible, gradually all Body numbers are reduced to Tx[1]. And gradually these T's become 3^o=1. So in the end the Body becomes 1. This 1 can be transferred to the Head. And the Head will end in cycles of 1-4-2-1-4-2-1.

In the calculation on next page you see the sequence of primary, secondary, tertiary numbers and the formation of many Cohorts.

Mutation, Multiplication and Division and (cont'd)

I have elaborated a random number in the above described way. Figure 24 shows the result.

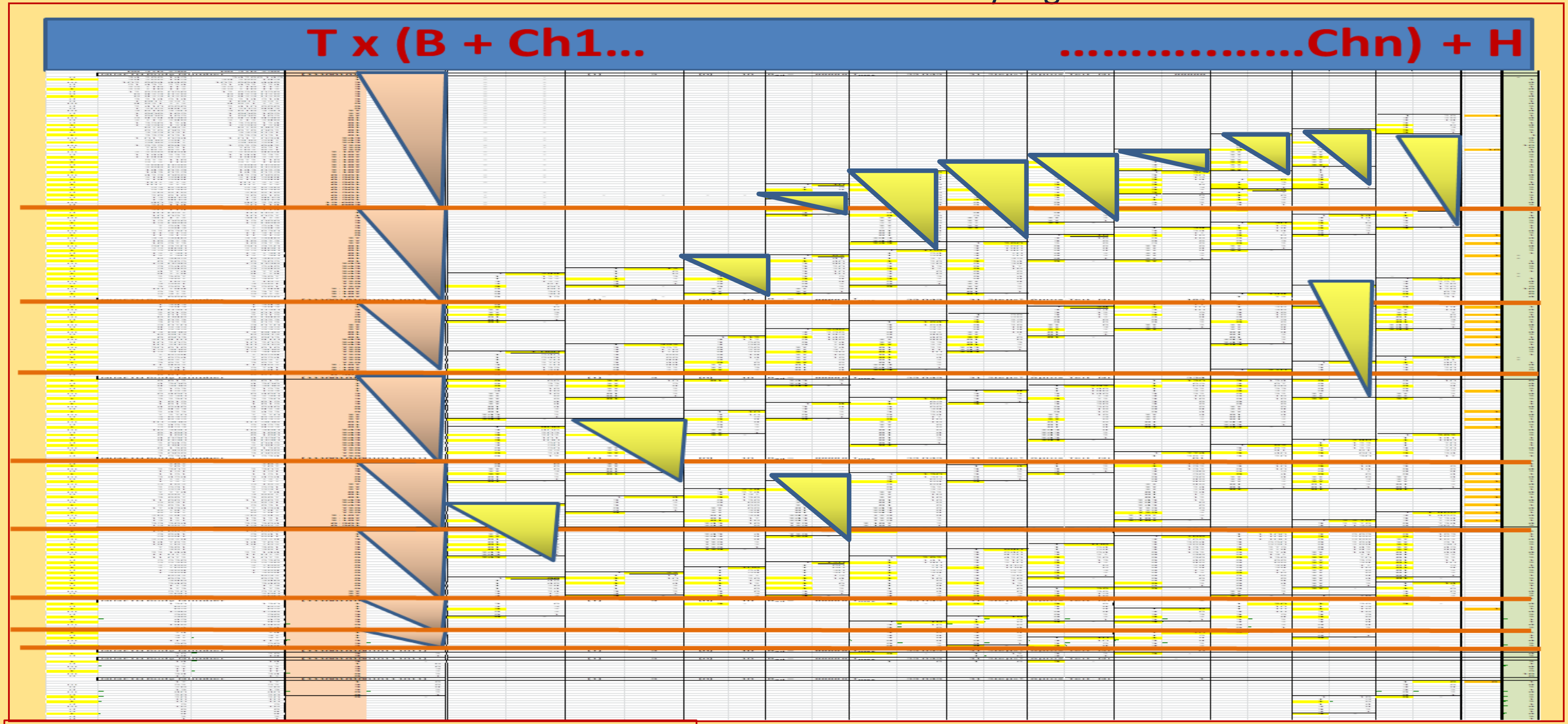


Figure 24. Decreasing Body and Cohort substrings

The primary Body number is reduced at each orange line. The Body string ends in Tx[1]. [T-1] becomes the next Body number. The [-1] is moved to the right as [+1] to form a cohort or to be added to the Head. The gradual reduction of Body and Cohort strings are represented by triangles. We see many small Cohort triangles that are formed and dissappear rather soon. The Head remains small.

Mutation, Multiplication and Division and (cont'd)

It is important to note that we can keep the Head smaller than 16. Sometimes a Cohort splits off a 3, which - added to an existing value of 2 of the Head - becomes 5. After Multiplication one gets 16. We can always split part of this Head - thus reducing the size of the Head to a value less than 16. .

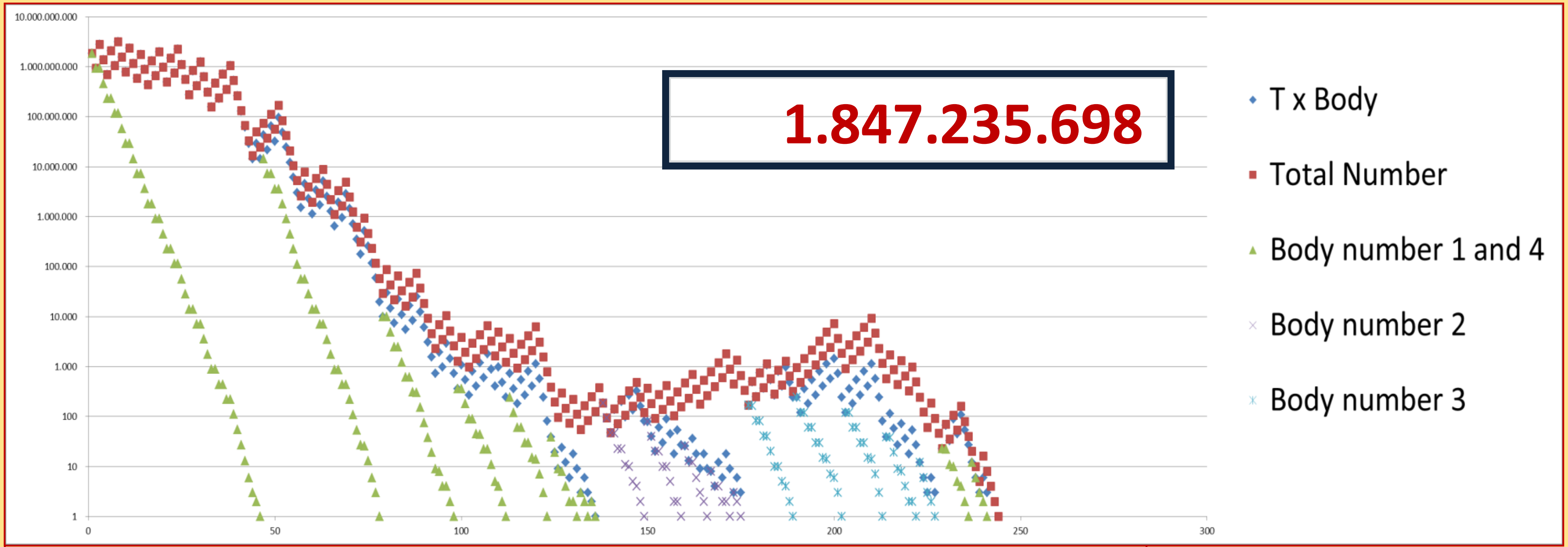


Figure 25. Body numbers 1 are reduced before the Total Number string turns to [1]

We now focus on the Body numbers. When the first Body number is reduced to 1, it's T has reached it's maximum value, which is ruled by the number of Multiplications of the Total Number so far. A new Body number starts. Since T is always Odd, it will directly split off a 1 at mutation. So the new Body number becomes T-1 (=M – we will come to this later on). In Figure 25 we see that the difference of Total number (red dots) minus TxB (blue dots) consists of a series of Cohorts and the Head. When Bodies number 1 sequel (green dots) is reduced, I have started with a Body number 2 sequel (light blue dots) with the sum of remaining Cohorts and Head.

Almost the End (cont'd)

Early in this excercise we learned that the number N = D-1 results in a maximum number with a value of 2T-2. After Division M appears, which equals T-1. Beautifull find!
My research concerning ascent/descent rates (ADR) of M-numbers learned that ADR tends to decrease with increasing M-values (Page 25). Except for M^3, M^7 and M^{11}. Look at binary number M^{11}=[10101100<u>1111111</u>010]. The long substring of [1]'s will result in a further upswing of the string. N resulting in the the maximum number values (upper Collatz boundary), let's analyse $N^{11} \rightarrow M^{11}$.

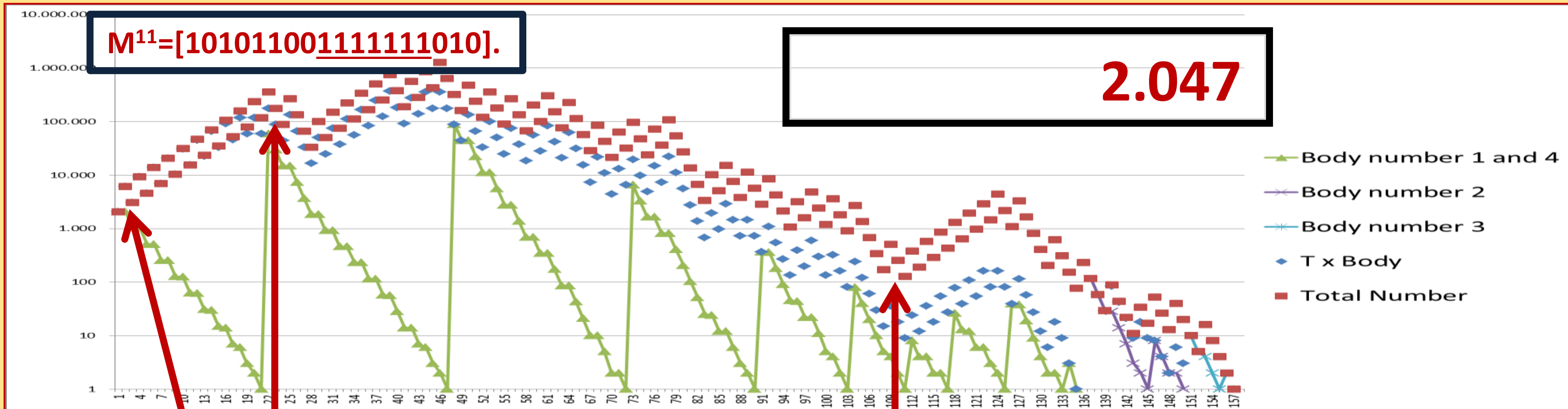


Figure 25. $N^{11} \rightarrow M^{11}$: Maximum ascent twice and later hitting 127 = N^7!

However, at the end of M^{11} string, a new number T+Head occurs with an ADR << 1. This N^{11} string is an extreme case. Other numbers X = D-2 and D-3 may have ADR's > 1.

Now we can finally answer the question Why a Collatz string always turns to 1.

.

Why does a Collatz string always end at 1?

To answer this question we write a number as T x Body + Head or, when we wish to keep Head as a small number: T x Body + Σ(T x Cohort) + Head. In this format T is a power of 3, starting with 0 that increases 1 at every multiplication. Cohort numbers are created in the process. The Primary number (T^0xB) is any number to start with.
Three processes take place in the TxB+ Σ(TxCohort)+H protocol:
Mutation, Multiplication and Division.
Before a multiplication or a division odd Body and Cohort numbers must be mutated to become Even. Hence, when the Primary is Odd, we'll directly mutate by splitting-off a [1] to create a Head. Otherwise, the Head will be formed after a first Mutation takes place.

Because Collatz conjecture = M conjecture + D-conjecture
The Body number string starts big and is gradually reduced to Tx[1]. T x Body + Σ(T x Cohort) + Head becomes a Secondary Number. We can now choose to continue with this Secondary or we can leave Cohorts and Head as they are. Then a new Body substring starts. Tx[1] is directly mutated to (T-1). Eventually the sequence of Body substrings end at 1x1. **This is the M-Conjecture**. After the primary body is processed, secondary Body substrings (and the Cohorts) all start with T→T-1.
During the process many small – but increasingly bigger – Cohorts are created. Cohort substrings all end in T^0x1 as well. From these T^0x1's new Cohorts are created.
H
The Head is kept < 16. It may become 5, but then continues -16-8…. At H=9-28 we must reduce Head to create a Cohort so H<16….eventually ending in Cycle 4-2-1-4-2-1-4 etc. **This is the D-conjecture**.

The final T^0x1 is added to the Head. And the Head always ends at 1 of the D-Cycle 1-4-2-1-etc.

Let's have a closer look at why a **T conjecture always end at T^0x1=1**

M-Conjecture

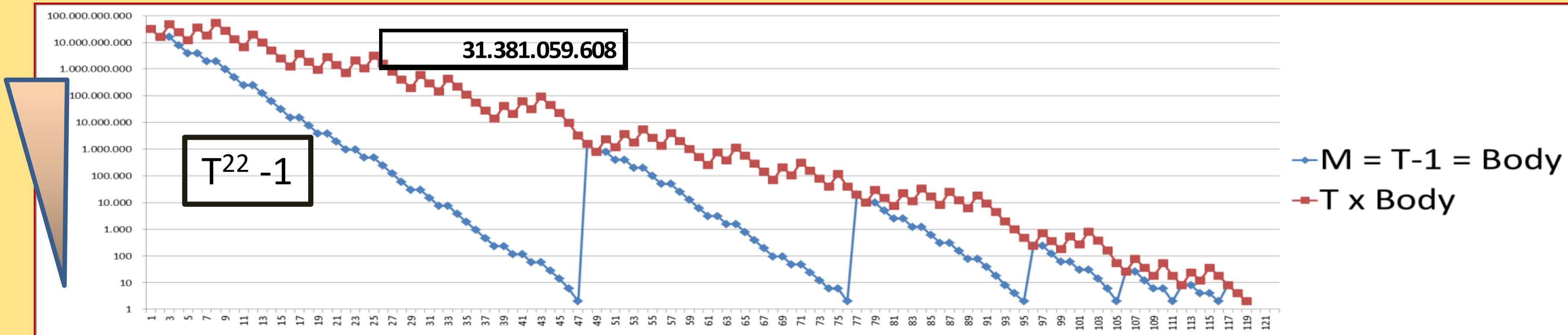


Figure 26. M-Conjecture

Suppose that in a very long string with a huge primary number a tertiary body number somewhere in the process ends at T^{22} x 1. A new Body starts at T^{22}-1=M: a [1] is split-off in the first mutation. This new Body is 'not aware' of his role in the Collatz process of the total Number X. Multiplication or Division is <u>not</u> determined by Odd/Even of the Body number. In fact the protocol is determined by the x-tary total number X being crunched away. When the cruncher meets a [1]→Multiplication and when it meets a [0]→Division. The Body number 'experiences' this protocol as an external factor. In my calculator a random generator simulates this 'external' sequence of {[1]→Mult.; [0]→Div.} (figure 27 right column). At [1] the T of the Body number shifts 1 power up. The total number X becomes Even because of the 3n+1 protocol. This is proven by [X1]:=3x[X1}+1=[2x[X1]+[X1+1]=[X00]+2+[X0]+2= 3.X.2+2.2, which is always Even→Division →3.X+2: Next Odd or Even is ruled by next [1] or [0] of X. <u>So, in the M-conjecture Multiplication is always followed by Division</u>.

Random (1-100) --> 1 or 0			
These columns determine Odd/Even of			**initial R value**
of a Body or Cohort in the middle of a			29
Collatz process - with random O/E values.			**1**
Collatz: always Even after Odd;			
Then next [digit] determines Even or Odd.			
Number of R's		**[1]- %-age**	*Control*
239		**60%**	*0,65*
			Total nmbr X
29	1	**177.146**	odd
		177.146	even
51	1	88.572	odd
		88.572	even
42	1	44.286	odd
		44.286	even
24	1	22.142	odd
		22.142	even
94	0	11.070	even
100	0	5.534	even
63	0	2.766	even
77	0	1.382	even
75	0		even

Figure 27. Odd or Even?

M-Conjecture cont'd

The sequence of [1]'s and [0]'s of x-tary total number rules the protocol. All numbers have an average of 50% of [1]'s and [0]'s. N-numbers have 100% [1]'s. D-numbers have 100% [0]s behind the first [1].

At 50% distribution of [1]'s and [0]'s the ADR of M (T-1) gets increasingly smaller $ADR=((3/2)^t/2^d)$, while t=d. For example, when t=d=16, a Body number starting at $P=T^{I}-1$ results in a number $Q=T^{II}x[1]+\Sigma T.Coh$. Q=0,010.P. The length of this substring is about 2.t+d=48 numbers.

At about 63% of [1]'s (behind the initial [1]) e.g. 30 [1]'s and 18 [0]'s, the string will keep about level.

When number X becomes N (only [1]'s) the string flares up – temporarily to 2.T-2 = 2.M.
M rules the protocol until it is crunched away. We have seen that the descent rate of increasing M's gets increasingly smaller ($T^{30}-1 \rightarrow$ 0.00067). The average percentage of [1]'s in M amounts To about 47,9% for M's from T^7 to T^{30}. M^{27} =7,625,597,484,986 = [1101110111101111001000001110111111110111010] . It has many [1]'s and N's in it. Still the string contains only 331 numbers.

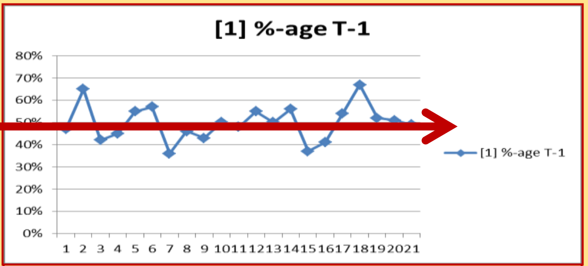


Figure 28. [1]-%-age of T-1

<u>More important is the ADR of Collatz-floor numbers C_n.</u>
My research learns that ADR of C_n gradually decreases to 0,41 at the 59^{th} C_n. Random number C_n: (3*<u>264,897,317</u>+1)/16 =49,668,247 has 56% [1]'s and an ADR of 0.143.

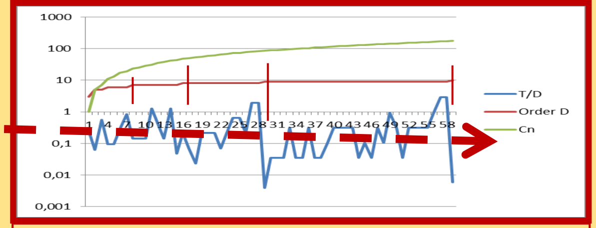


Figure 15. ADR first 59 C_n nmbrs)

Conclusion: The M-Conjecture is confirmed:
Eventually all sequals of 'triangles' get to Tx[1]. The final Tx[1]=1 is added to the Head (H< 16) that then becomes D-Cycle 1-4-2-1-etc. (D-conjecture), which is confirmed as well.

Another way of explaining

Let's have a look at the Primary number creating the maximum ascent of the string. This number is:

$P=N=D-1.$ D is a power of 2.

This is a binary number consisting of only [1]'s. The maximum string number:

$Q = 2xM= 2xT-2.$ T is a power of 3.

remember: 3 = 2 + 1 ;-) $\rightarrow M = T^{x}-1 = 2 \text{ x } T^{x-1}+ T^{x-1} -1$

$= 2 \text{ x } T^{x-1}+ 2 \text{ x } T^{x-2} + T^{x-2} -1$

$= 2 \text{ x } T^{x-1}+ 2 \text{ x } T^{x-2} + 2 \text{ x } T^{x-3} \ldots\ldots 2 \text{ x } T^{x-(x-1)} + 1 \text{ x } T^{x-(x-1)} -1$

$= 2 \text{ x } T^{x-1}+ 2 \text{ x } T^{x-2} + 2 \text{ x } T^{x-3} \ldots\ldots 2 \text{ x } T^{1} + 1 \text{ x } 3 -1$

$= 2 \text{ x } \Sigma T^{(x \text{ until } 1)} + 2$

$\rightarrow M = \Sigma T^{(x \text{ until } 1)} + 1$

In TxB+H format: $M = T^{0} \text{ x } \Sigma T^{(x \text{ until } 1)} + H$

in which H equals 1. The Body number equals $\Sigma T^{(x \text{ until } 1)}$ and T^0 is another way of writing 1.

M is a special number. All other numbers can be written as ΣDxT + H, while D = 1 or 2 and H = 1 or 2.

In TxB+TxCohort+H format: $P = T^{0}xT^{max} \text{ x } \Sigma DxT + H$

(graphically) in the process: P = (T x + Σ (T x + T x .....etc) + **H**

The triangles represent Body- and Cohort numbers. Prior to each multiplication or division these numbers are mutated to an Even number. Tx1's are combined to new B's and Cohorts. The triangles always reduce to 1 (see page 36). **This is the M-conjecture**.

In the process we keep H<16. When – due to 3.n+1 - the Head would become too big, we reduce it by creating a Cohort. E.g. H=9→28;= 28+0 or 24+4 or 16+12.

So, in the end all (TxB + ΣTxCoh)'s reduce to 1+1→1 which is added to the Head.

<u>Head +1 always ends at 1 of the Cycle 1-4-2-1-etc. **That is the D-Conjecture.**</u>

In fact the Collatz conjecture consists of an M-conjecture and a D-conjecture.

A nasty number as desert

In underneath example we allow the Head to develop until the Body B_1 – after several cycles – is finally reduced to T x [1]=1. Then we let number B_2 := Head and the total number S = T^0 x B_2 + 1. After TxB_2 is reduced to 1 x 1, a number B_3 is created out of the 1 + Head and so on. It takes four B-sequals to end at the Final 1. In each B-sequal the Head growing (space between Red and Blue dots). We can choose to keep the Head small and create Cohorts. Then you'll find Σd.T between the dots ΣD.T becomes a number with many [1]'s. Like e.g. ΣD.T = $T^5+T^4+T^2+2xT^1+T^0$ = 85 + 170 = [1010101] + [10101010] = [11111111]. This may temporarily result in an ascent of the string.
When we keep H=<4, the Head cycles 1-4-2-1-4-2-1. In the Collatz string of 63,728,127 the string apparently meets several times a numbers with many [N]'s.

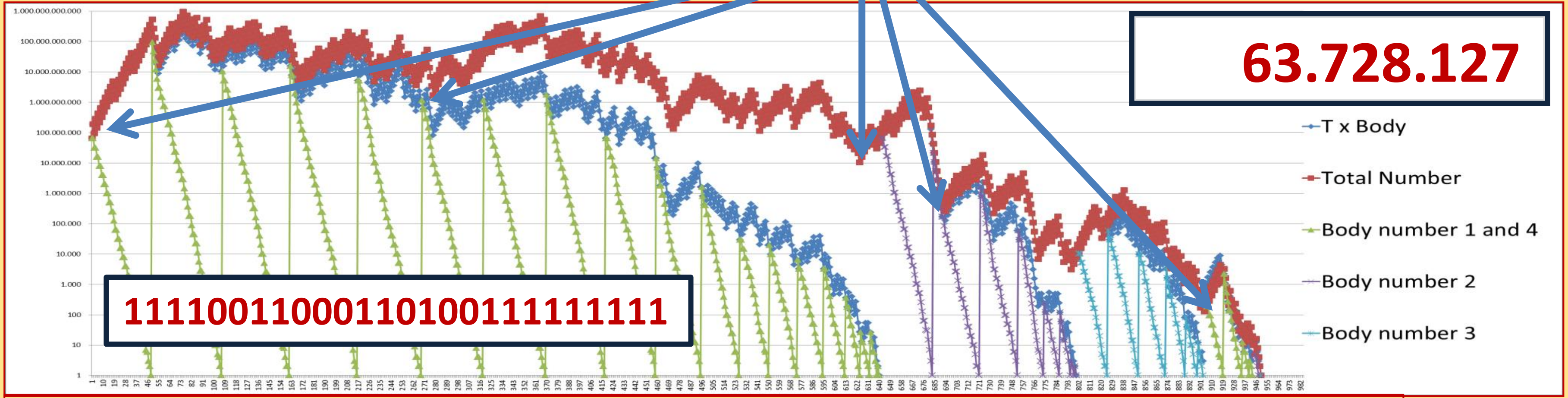


Figure 26. Number 63,728,127. Note the N^9 at the end of the number resulting in an initial up swing

This number 63,728,127 is the number smaller than 100,000,000 with the largest number of steps. It takes 951 steps to reduce this number to 1. In the end (TxB + ΣtxCoh)'s reduce to 1.

This nasty number also ends at 1 of the Cycle 1-4-2-1-etc.

References

Wikipedia: https://en.wikipedia.org/wiki/Collatz_conjecture
- this entry contains an extended listing of more than 40 references
Y. Sinai – statisical 3x+1 problem – arXiv:math/0201102v2 - January 13 2003
T. Tao – recent article on Tao in New Scientist (Dutch edition).

I am not familiar with mathematical language as used in many articles on Collatz.
That's why I refer to only a few publications.