A fractal algorithm shows prime number patterning.

Birke Heeren[1][0000-0001-6731-583X]

[1] independent researcher, Greifswald, MV, 17489, Germany
info@birkeheeren.de 17. September 2025

Abstract. This article focuses on modeling the deterministic, symbolic, and fractal **patterns** underlying prime numbers and their gaps, revealing structured prime generation **beyond** traditional statistical **distribution** theories. We present a novel mathematical machine with a novel fractal algorithm for modeling the pattern of prime numbers and gaps. This framework leverages three fractal processes with deterministic rules to construct six walksets. Walksets A_n and AP_n act as collectors, B_n and BP_n determine prime or composite, C_n starts with all natural numbers and CP_n contains the fractal process with always periodic patterns. Primes and gaps arise as emergent phenomena within this rule-based system – proving that this Ansatz holds for all prime numbers and gaps. We call the machine "Synchronous Factory Automaton" SFA. It was implemented in Java.

Keywords: prime numbers, gaps, periodic patterning, fractals. patterns

1 Introduction

The SFA does not merely sieve primes – it grows primes and gaps. Primes emerge as stable survivors in A_n and AP_n in a deterministic symbolic landscape. Thus, this work demonstrates what for 2000 years was thought of impossible. Du Sautoy (2004) "Es ist unmöglich, für eine Liste von Primzahlen vorherzusagen, wann die nächste Primzahl auftauchen wird. Die Liste erscheint chaotisch, zufällig, und es gibt keinerlei Hinweise, wie man die nächste Zahl bestimmen könnte."
(my video: https://www.youtube.com/watch?v=2W1oi7g1rdE)

2 Tools

2.1 Walksets

For the SFA it was necessary to define sorted sets, which were coined "walksets" (W) as a distinction to sets, which are always unsorted. Walksets can be thought of as a walk on the number ray and are different from intervals, as they only can contain discrete numbers and symbols. Walksets have direction, they can be empty, they are written with angle brackets. Infinity is only possible either at the start or end of the

walkset or otherwise as a periodic or infinite term that encompasses the whole walkset.

$$W_{empty} := < > \quad (1)$$
$$W_{natural\ numbers} := <1, 2, \ldots \infty> \quad (2)$$
$$W_{symbol\ pattern\ 1} := <L, \ldots \infty> = < \bar{} L> \quad (3)$$
$$W_{symbol\ pattern\ 2} := < \bar{} LM> \quad (4)$$
$$W_{symbol\ pattern\ 3} := < \bar{} MLMLMM> \quad (5)$$
$$W_{symbol\ pattern\ 4} := < \bar{} LMLMMM> \quad (6)$$
$$W_{infinity\ of\ natural\ numbers} := < \infty_N> \quad (7)$$

2.2 Symbols

The symbol L (live) means undetermined whether prime or composite number. The symbol M (multiple) means composite number and the symbol P (prime) means prime number. The symbol 1 means the number one which is not a prime number.

2.3 Primorial

The size of the pattern (= width of the sieve = period length) in CP_n is calculated by multiplying all prime numbers up to and including the lower bound (B_n, BP_n) of the sieve, thus the primorial. This size increases rapidly (Fig. 1).

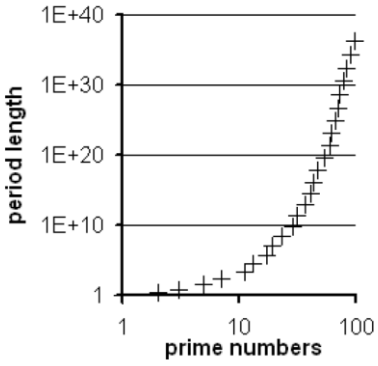


Figure 1 Size of the pattern = width of the sieve = period length.

2.4 Algorithms

The fractal algorithm is called FA, and a second implementation of the FA is called BLOX.

3 SFA with FA

SFA with FA is shown on Fig. 2. These are the first steps:

$A_{start} = <>$ $B_{start} = <>$ $C_{start} = <1, 2, \ldots \infty>$ (8)

$A_1 = <>$ $B_1 = <1>$ $C_1 = <2, 3, \ldots \infty>$ (9)

$A_2 = <1>$ $B_2 = <2>$ $C_2 = <3, 4, \ldots \infty>$ (10)

$A_3 = <1, 2>$ $B_3 = <3>$ $C_3 = <4, 5, \ldots \infty>$ (11)

$A_4 = <1, 2, 3>$ $B_4 = <4>$ $C_4 = <5, 6, \ldots \infty>$ (12)

$A_5 = <1, 2, 3, 4>$ $B_5 = <5>$ $C_5 = <6, 7, \ldots \infty>$ (13)

At the same time, the FA pattern-walksets with symbols are kept synchronized with the numbers.

$AP_{start} = <>$ $BP_{start} = <>$ $CP_{start} = <\bar{}L>$ (14)

$AP_1 = <>$ $BP_1 = <1>$ $CP_1 = <\bar{}L>$ (15)

$AP_2 = <1>$ $BP_2 = <P>$ $CP_2 = <\bar{}LM>$ (16)

$AP_3 = <1, P>$ $BP_3 = <P>$ $CP_3 = <\bar{}MLMLMM>$ (17)

$AP_4 = <1, P, P>$ $BP_4 = <M>$ $CP_4 = <\bar{}LMLMMM>$ (18)

$AP_5 = <1, P, P, M>$ $BP_5 = <P>$ $CP_5 =$

$<\bar{}MLMMMLMLMMMLMLMMMLMMMMMLMLMMMM>$ (19)

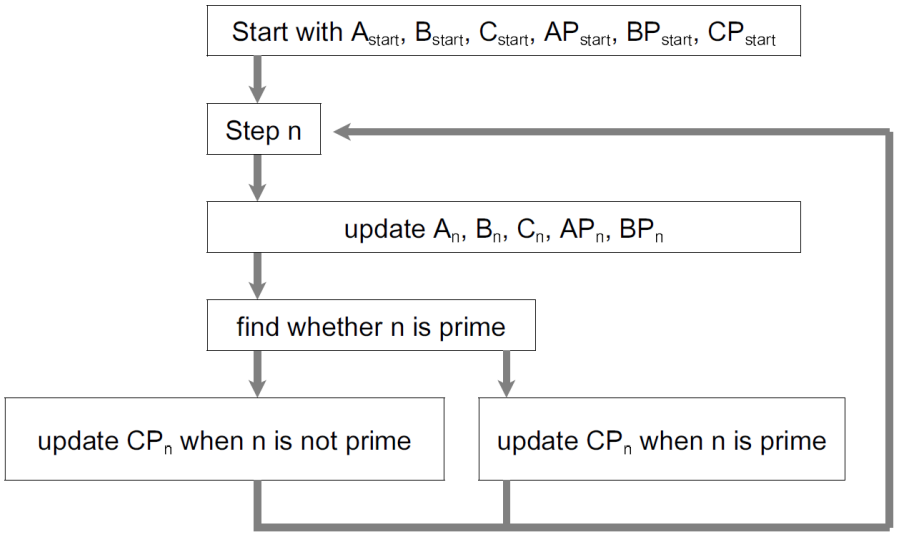


Figure 2 SFA with FA

3.1 Update of walkset A_{n-1} to A_n

The only element of B_{n-1} is cut from B_{n-1} and pasted into A_{n-1} on the right side. Thus set A_{n-1} becomes A_n. At the start B_{start} is an empty walkset. Since there is no element in B_{start} to cut, A_1 is an empty walkset.

3.2 Update of walkset B_{n-1} to B_n

The leftmost element of C_{n-1} is **cut** and pasted into B_{n-1} left empty by the update of A. Thus B_{n-1} again contains one element, that is the step number and thus becomes B_n. At the start B_{start} is an empty walkset therefore as B is filled for the first time it becomes B_1.

3.3 Update of walkset C_{n-1} to C_n

The update of walkset C has already taken place during the update of B by cutting the leftmost element of C_{n-1}. Thus C_{n-1} has become C_n.

3.4 Update of pattern walkset APn-1 to APn

The only element of BP_{n-1} is cut from BP_{n-1} and pasted into AP_{n-1}. The new element becomes the rightmost element to keep AP in match with A. Thus AP_{n-1} becomes AP_n. At the start BP_{start} is an empty walkset. Therefore, AP_1 is an empty walkset, as there is no element in BP_{start} to cut.

3.5 Update of pattern walkset BPn-1 to BPn

The leftmost element of CP_{n-1} is **copied** from CP_{n-1} and pasted into BP_{n-1} left empty by the update of AP. Thus BP_{n-1} again contains one element that is the type of the step number and thus becomes BP_n. At the start BP_{start} is an empty walkset, therefore as BP_{n-1} is filled for the first time it becomes BP_1.

3.6 Lemma

The equivalence of cutting in C and moving in CP. Proof:

At step number n = 3 CP_n:	<‾MLMLMM>
This means:	<MLMLMM, MLMLMM, MLMLMM, M... ∞>
Cutting the first letter leaves:	<LMLMM, MLMLMM, MLMLMM, M... ∞>
This can be rewritten as:	<LMLMMM, LMLMMM, LMLMMM, ... ∞>
At step number n = 4 CP_n is:	<‾LMLMMM>

3.7 Find whether n is prime.

Element n, that is the step number is contained in B_n. Its type-information is contained in BP_n. In the case that BP_n contains the element M then the current step number is not prime. If BP_n contains the element L, then the current step number n is prime.

This is denoted by changing L into P in BP_n except for number one which turns L into symbol 1.

3.8 Update pattern walkset CP_{n-1} to CP_n when n is not prime.

Fractal procedure: move.
As can be seen in the description (3.5) "Update of pattern walkset BP_{n-1} to BP_n" the type-information of step number n is **not cut** from CP_{n-1}. Instead, this leftmost type-information is now **moved** to the rightmost place of the periodic term. Thus CP_{n-1} becomes CP_n (see lemma).

3.9 Update pattern walkset CP_{n-1} to CP_n when n is prime.

Fractal procedure: move.
The first procedure is the same as for "n is not prime".

Fractal procedure: copy.
The pattern size is increased by copying the pattern and pasting it n-1 times to the right of itself.

Fractal procedure: change.
The types of all numbers x for which applies

$$x*n \text{ with } (x \in \mathbb{N} \wedge x > 1 \wedge x*n \leq \text{pattern size}_n + n)$$

inside CP_{n-1} are turned from undetermined L types into M types, unless they are already of type M. Thus CP_{n-1} becomes CP_n.

4 SFA with FA results

Let us think of the walksets B_n, C_n, and CP_n of SFA as sieves (Fig. 3). The pattern of all L's is the envelope ("Hüllkurve" like in physics) of all prime numbers above the step number n in B_n. The pattern of all M's is the area outside the prime number envelope.

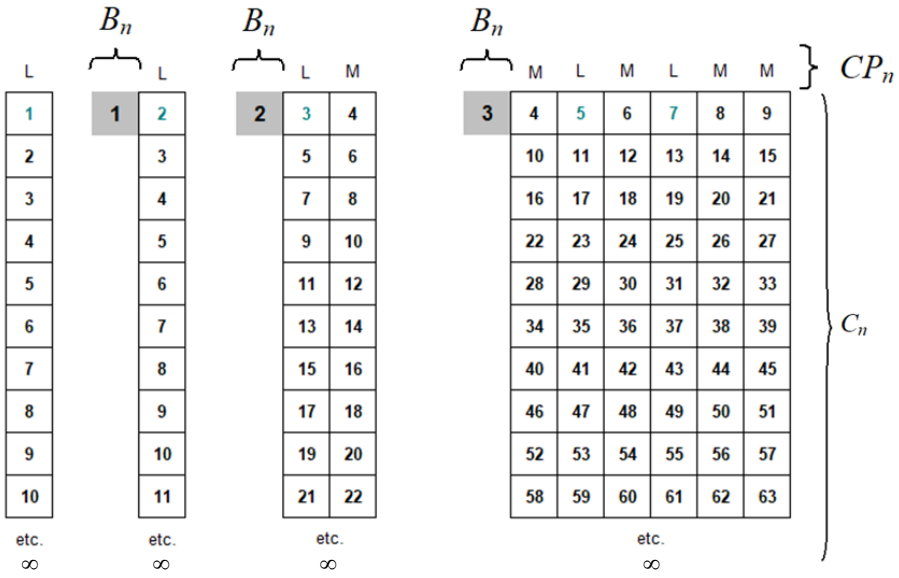


Figure 3 The sieves of starting state and step n = 1 to 3.

At first sight the sieves could be mistaken for Eratosthenes sieves, but there are important differences. The SFA sieves have a lower bound (B_n), no upper bound and consist of vertical columns, which can be described by linear equations. The L columns contain not only prime numbers but also composite numbers, therefore they are envelopes to prime numbers.

4.1 Proof

The stringent logic of the algorithm is proof. The algorithm was implemented in Java. We are professionals in science and computer science and want to pass the baton on this to the higher mathematicians.

4.2 Prime number envelops – L-columns.

All envelope equations come directly from the SFA sieves Fig. 3.

At step number n = 1 the L column starts with 2 and then 3, 4, 5, to infinity. This leads to the trivial prime number envelope equation with $x \in N_0$

$$f(x)=1x+2 \tag{20}$$

Fig. 3 at step number 2 the L column starts with 3, the width of the sieve is 2 this leads to the envelope equation with $x \in N_0$

$$f(x)=2x+3 \tag{21}$$

Fig. 3 at step number 3 the L columns start with 5 and 7, the width of the sieve is 6 this leads to a family of prime number envelope equations with $x \in N_0$.

f(x)=6x+5 (22)
f(x)=6x+7 (23)

Followed by an even larger family of equations for step number n = 5 envelope (no Fig.) with $x \in N_0$.

"Eq. 22" spawns:
f(x)=30x+11 (24)
f(x)=30x+17 (25)
f(x)=30x+23 (26)
f(x)=30x+29 (27)

"Eq. 23" spawns:
f(x)=30x+7 (28)
f(x)=30x+13 (29)
f(x)=30x+19 (30)
f(x)=30x+31 (31)

The size of the sieves increases from step to step with n primorial (Fig. 1). Also, for each following step of the SFA with FA the number of prime number envelope equations increases rapidly (Fig. 4).

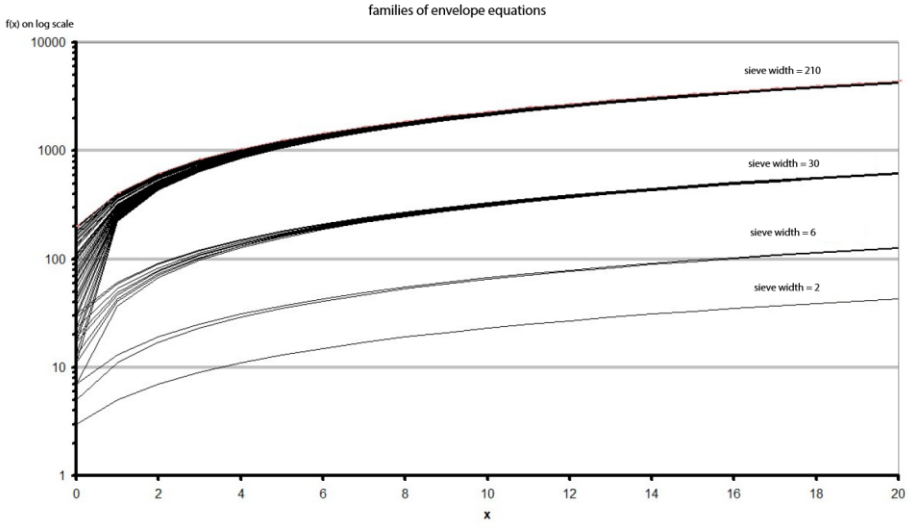


Figure 4 Families of envelope equations on log_{10} scale with $x \in N_0$

For each prime number the envelopes can be determined. For example, prime number 193877777 belongs to envelopes with $x \in N_0$.

f(x)=1x+2 (32)
f(x)=2x+3 (33)

$$f(x)=6x+5 \quad (34)$$
$$f(x)=30x+17 \quad (35)$$
$$f(x)=210x+107 \quad (36)$$
$$f(x)=2310x+1787 \quad (37)$$

4.3 Composites – M-columns

All the M columns by the logic of the FA contain only composite numbers (gaps).

4.4 Hypotheses

The following equations (we call them latisses, 38 to 75) are hypothesized with $x \in N_0$. With each step of the FA the number of latisses equations increases rapidly.

$$6x_1+7 = 6x_2+5 * 6x_3+5 \quad (38)$$
$$6x_1+7 = 6x_2+7 * 6x_3+7 \quad (39)$$
$$6x_1+5 = 6x_2+5 * 6x_3+7 \quad (40)$$

$$30x_1+7 = 30x_2+7 * 30x_3+31 \quad (41)$$
$$30x_1+7 = 30x_2+11 * 30x_3+17 \quad (42)$$
$$30x_1+7 = 30x_2+13 * 30x_3+19 \quad (43)$$
$$30x_1+7 = 30x_2+23 * 30x_3+29 \quad (44)$$

$$30x_1+13 = 30x_2+7 * 30x_3+19 \quad (45)$$
$$30x_1+13 = 30x_2+13 * 30x_3+29 \quad (46)$$
$$30x_1+13 = 30x_2+17 * 30x_3+31 \quad (47)$$
$$30x_1+13 = 30x_2+19 * 30x_3+23 \quad (48)$$

$$30x_1+19 = 30x_2+7 * 30x_3+7 \quad (49)$$
$$30x_1+19 = 30x_2+11 * 30x_3+29 \quad (50)$$
$$30x_1+19 = 30x_2+13 * 30x_3+13 \quad (51)$$
$$30x_1+19 = 30x_2+19 * 30x_3+31 \quad (52)$$
$$30x_1+19 = 30x_2+23 * 30x_3+23 \quad (53)$$

$$30x_1+31 = 30x_2+7 * 30x_3+13 \quad (54)$$
$$30x_1+31 = 30x_2+11 * 30x_3+11 \quad (55)$$
$$30x_1+31 = 30x_2+17 * 30x_3+23 \quad (56)$$
$$30x_1+31 = 30x_2+19 * 30x_3+19 \quad (57)$$
$$30x_1+31 = 30x_2+29 * 30x_3+29 \quad (58)$$
$$30x_1+31 = 30x_2+31 * 30x_3+31 \quad (59)$$

$$30x_1+11 = 30x_2+7 * 30x_3+23 \quad (60)$$
$$30x_1+11 = 30x_2+11 * 30x_3+31 \quad (61)$$
$$30x_1+11 = 30x_2+13 * 30x_3+17 \quad (62)$$
$$30x_1+11 = 30x_2+19 * 30x_3+29 \quad (63)$$

$30x_1+17 = 30x_2+7 * 30x_3+11$ (64)
$30x_1+17 = 30x_2+13 * 30x_3+29$ (65)
$30x_1+17 = 30x_2+17 * 30x_3+31$ (66)
$30x_1+17 = 30x_2+19 * 30x_3+23$ (67)

$30x_1+23 = 30x_2+7 * 30x_3+29$ (68)
$30x_1+23 = 30x_2+11 * 30x_3+13$ (69)
$30x_1+23 = 30x_2+17 * 30x_3+19$ (70)
$30x_1+23 = 30x_2+23 * 30x_3+31$ (71)

$30x_1+29 = 30x_2+7 * 30x_3+17$ (72)
$30x_1+29 = 30x_2+11 * 30x_3+19$ (73)
$30x_1+29 = 30x_2+13 * 30x_3+23$ (74)
$30x_1+29 = 30x_2+29 * 30x_3+31$ (75)

4.5 Trees

A way of visualizing the L- and M-columns is that all L-columns build an L-tree (Fig. 19), and all M-columns build an M-tree.

5 Prime gaps, three consecutive primes and twin primes

We define letters a, b, c and d.
a := <LMLMMM> (76)
b := <LMMMMM> (77)
c := <MMLMMM> (78)
d := <MMMMMM> (79)

Because step number n is larger than two, with each step only one L of letter a can be turned to M (Fig. 5).

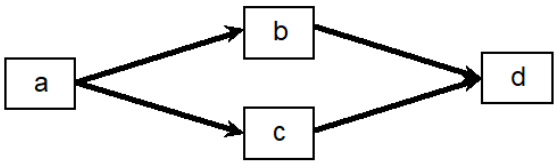


Figure 5. Letter a can turn into b or c. Letters b and c can turn into letter d. Letter d cannot change anymore.

Letter d and clusters of letter d are prime gaps. As the SFA walks from step to step all prime gaps are multiplied and transferred to the next sieve. Thus, any large prime gap that was ever discovered will appear repeatedly to infinity and can only become larger.

(80)

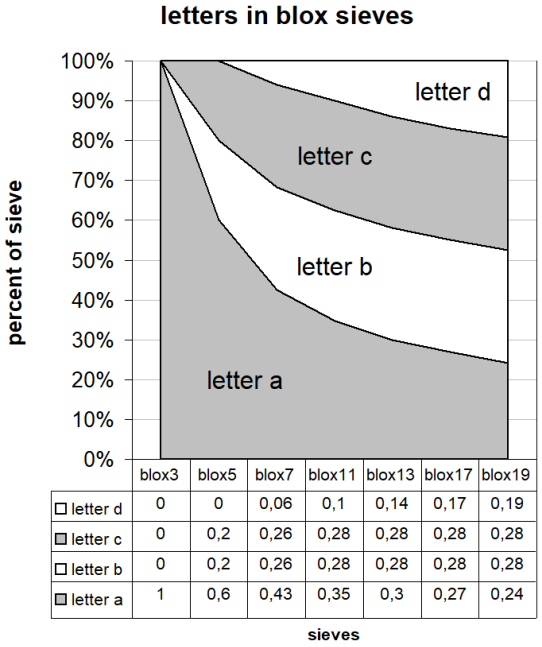
It holds true that no matter in which sequence the letters a, b, c, and d are arranged it is impossible to build the pattern of three consecutive primes <LMLML> beyond 3, 5, and 7.

(81)

For the twin primes pattern <LML> which appears only in letter a, it can be said that the density in the natural numbers decreases with each step of the FA, as relatively more "letters a" turn into b, c and ultimately d until blox19 (Fig 6).

In the SFA symbolic framework, "letter a" encodes the twin prime template <LML>. Only "letter a" can produce twin primes in subsequent sieve steps. If "letter a" were to vanish beyond some step, only decaying letters b, c, and d would remain, possibly leading eventually to a prime-empty pattern: all letter d — contradicting the known infinitude of primes. Therefore, the FA probably generates infinitely many letters a. This suggests — in the context of the model — the existence of plausibly, infinitely many actual twin primes.

Two fractal procedures "copy" and "change" each have different effects on the symbols L and M and therefore on the letters. Procedure "copy" multiplies letter a (the prey). Procedure "change" (the predator) diminishes "letter a". This reminds of "predator-prey" interactions. We hypothesize that the density of letter a comes to a steady state, possibly with some oscillations.



	blox3	blox5	blox7	blox11	blox13	blox17	blox19
letter d	0	0	0,06	0,1	0,14	0,17	0,19
letter c	0	0,2	0,26	0,28	0,28	0,28	0,28
letter b	0	0,2	0,26	0,28	0,28	0,28	0,28
letter a	1	0,6	0,43	0,35	0,3	0,27	0,24

Figure 6. How "letter a" decreases from sieve to sieve- and letters b, c and d increase -until blox19.

6 Factorization of semiprimes

Semiprimes are the product of two prime numbers. Semiprimes up to 12 digits with similar factor sizes were factorized with an office laptop by making use of latisses and targeted trial and error to determine the x_2 and x_3 values of the right-hand side, which can be computed in parallel. For example, the semi prime 82557089 was found to be of latisses.

$6x_1+5 = 6x_2+5 * 6x_3+7$ (82)

and

$82557089 = (2310*4 + 839) * (2310*3 + 1261)$ (83)

leading to prime number factors

$82557089 = (10079) * (8191)$ (84)

7 Prime Number candidate production

For using large sieves, the BLOX variant of the FA was developed. Each sieve is a file that contains only L columns (= L envelope). The computation started with one file at step n = 3 containing 5 and 7 and ended through successive computation at step n = 29 with 667 files, each containing about 1.6 million L entries in sieve blox29.

The BLOX computation is done by the equivalence of the FA. But the BLOX does not grow primes, but envelopes of primes, therefore prime candidates, which need to be confirmed by the usual primality tests.

The BLOX was implemented in Java, just like the SFA with FA.

From step number n = 3 to n = 5 five copies of the L-columns are made in the following way (Fig. 7). The sieve-size of n = 3 that is 6 is used. Five and composites of five are cut.

5	cut
7	7
5 + 1*6 = 11	11
7 + 1*6 = 13	13
5 + 2*6 = 17	17
7 + 2*6 = 19	19
5 + 3*6 = 23	23
7 + 3*6 = 25	cut
5 + 4*6 = 29	29
7 + 4*6 = 31	31

Figure 7 blox5

For blox7 sieve-size of n = 5 is used, that is 30. Seven copies are made. 7 and composites of 7 are cut. And so on and so forth. The BLOX sieves contain all prime numbers in the range of the sieve and some composite numbers.

With blox29 three cases were considered.

1.) Picking 310-digit prime numbers at random from all natural numbers (**0.12% success rate**) (Fig. 8).
2.) Picking 310-to-312-digit prime numbers at random from inside the envelope with sieve blox29 (**0.90% success rate**) (Fig. 9).
3.) Picking 310-to-312-digit prime numbers outside the envelope with blox29, where no prime numbers in 200,000 picks could be found.

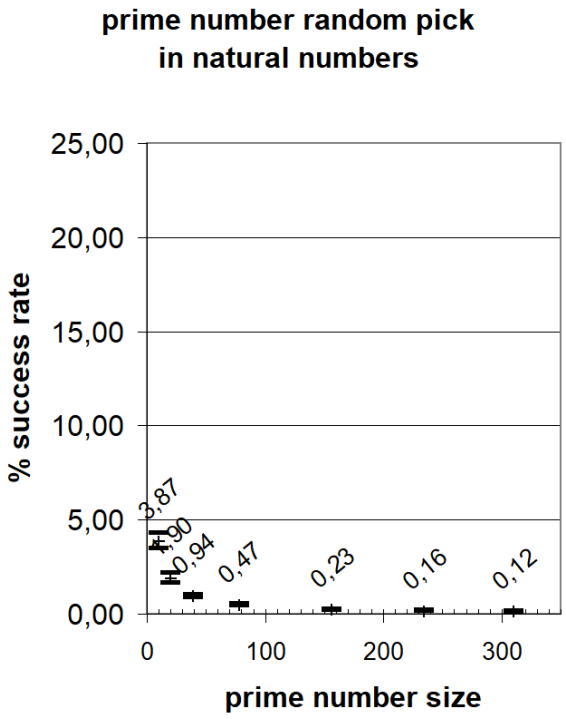


*Figure 8. Prime number random pick **in natural numbers** for prime sizes 10, 20, 39, 78, 156, 234, and 310; 1000 picks each with standard deviation.*

The picking of large, here 310-to-312-digit prime candidates is done the following way. A large sieve size 743 primorial was used which has 310 digits (= sieve size). A number from the blox29 picked at random (= L) is used and a random integer factor between 0 and 99 (= x). The formula for the prime candidate is:

prime candidate = sievesize743 * x + Lblox29 (85)

Because sievesize743 (85) contains factor 29 primorial, the prime candidate is picked out of an L column of blox29, thus inside the envelope of blox29. For limits of computation resources, x factors were kept to the small range of between 0 and 99.

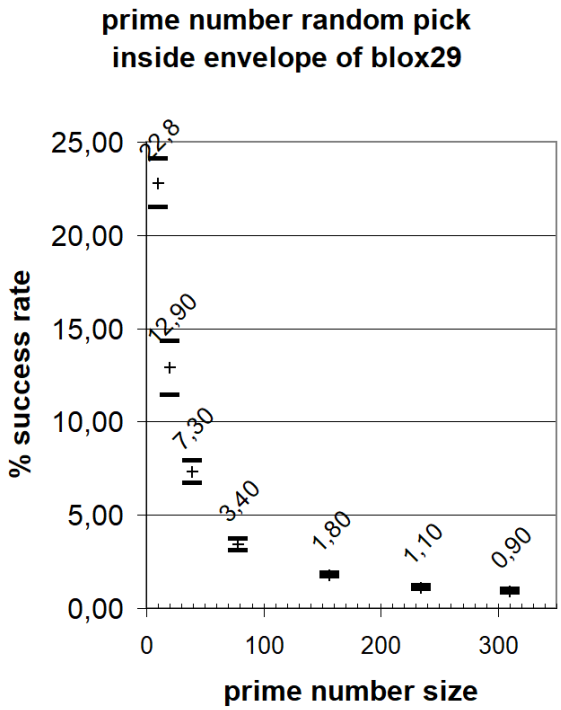


*Figure 9 Prime number random pick **in envelope blox29**, for sievesizes 10, 20, 39, 78, 156, 234, and 310; 1000 picks each with standard deviation.*

8 The composites in the prime envelopes

For envelopes equations (22) and (23) the composites follow a simple emergence law, which is an algorithm, Fig. 10 and Fig. 11. In Fig. 10 it starts f(x)=6x+5 with x = 0 f(x)=5. Now every 5th number is a composite. The first composite is 35 = 5*7. From this point on every 7th number is a composite, too. The next composite on Fig. 10 is 65=5*13. From this point on every 13th number is composite, too. The next composite is 77=7*11. From this point on every 11th number is a composite, too. It is hypothesized that this goes on and so forth.

On Fig. 11 it starts f(x)=6x+7 with x = 0 f(x)=7. Now every 7th number is a composite. In logic with latisses (38) and (39) the first composite comes from 25=5*5. From this point on every 5th number in the envelope is a composite. The next composite is 49=7*7, nothing happens, because we already got every 7th pattern. But the next composite is 55=5*11. From this point on every 11th number is a composite, too. It is hypothesized that this goes on and so forth.

The emergence law of composites inside the envelopes very much reminds us of the emergence law of composites in all natural numbers. But the order of prime numbers appearing is different. On Fig. 10 it is: 5, 7, 13, 11, 19, 17, 31, 23, 37, 29, 43. On Fig. 11 it is: 7, 5, 11, 17, 13, 23, 19, 29, 41, 31, 37.

	5x	7x	13x	11x	19x	17x	31x	23x	37x	29x	43x
	5										
	11										
	17										
	23										
	29										
5*7	35	35									
	41	41									
	47	47									
	53	53									
	59	59									
5*13	65	65	65								
	71	71	71								
7*11	77	77	77	77							
	83	83	83	83							
	89	89	89	89							
5*19	95	95	95	95	95						
	101	101	101	101	101						
	107	107	107	107	107						
	113	113	113	113	113						
7*17	119	119	119	119	119	119					
5*5*5	125	125	125	125	125	125					
	131	131	131	131	131	131					
	137	137	137	137	137	137					
11*13	143	143	143	143	143	143					
	149	149	149	149	149	149					
5*31	155	155	155	155	155	155	155				
7*23	161	161	161	161	161	161	161	161			
	167	167	167	167	167	167	167	167			
	173	173	173	173	173	173	173	173			
	179	179	179	179	179	179	179	179			
5*37	185	185	185	185	185	185	185	185	185		
	191	191	191	191	191	191	191	191	191		
	197	197	197	197	197	197	197	197	197		
7*29	203	203	203	203	203	203	203	203	203	203	
11*19	209	209	209	209	209	209	209	209	209	209	
5*43	215	215	215	215	215	215	215	215	215	215	215
13*17	221	221	221	221	221	221	221	221	221	221	221
	227	227	227	227	227	227	227	227	227	227	227
	233	233	233	233	233	233	233	233	233	233	233
	239	239	239	239	239	239	239	239	239	239	239
5*7*7	245	245	245	245	245	245	245	245	245	245	245
	251	251	251	251	251	251	251	251	251	251	251
	257	257	257	257	257	257	257	257	257	257	257
	263	263	263	263	263	263	263	263	263	263	263
	269	269	269	269	269	269	269	269	269	269	269
5*5*11	275	275	275	275	275	275	275	275	275	275	275
	281	281	281	281	281	281	281	281	281	281	281

Figure 10 The emergence of composites in envelope 6x+5 up to 281.

	7x	5x	11x	17x	13x	23x	19x	29x	41x	31x	37x
	7										
	13										
	19										
5*5	25	25									
	31	31									
	37	37									
	43	43									
7*7	49	49									
5*11	55	55	55								
	61	61	61								
	67	67	67								
	73	73	73								
	79	79	79								
5*17	85	85	85	85							
7*13	91	91	91	91	91						
	97	97	97	97	97						
	103	103	103	103	103						
	109	109	109	109	109						
5*23	115	115	115	115	115	115					
11*11	121	121	121	121	121	121					
	127	127	127	127	127	127					
7*19	133	133	133	133	133	133	133				
	139	139	139	139	139	139	139				
5*29	145	145	145	145	145	145	145	145			
	151	151	151	151	151	151	151	151			
	157	157	157	157	157	157	157	157			
	163	163	163	163	163	163	163	163			
13*13	169	169	169	169	169	169	169	169			
5*5*7	175	175	175	175	175	175	175	175			
	181	181	181	181	181	181	181	181			
11*17	187	187	187	187	187	187	187	187			
	193	193	193	193	193	193	193	193			
	199	199	199	199	199	199	199	199			
5*41	205	205	205	205	205	205	205	205	205		
	211	211	211	211	211	211	211	211	211		
7*31	217	217	217	217	217	217	217	217	217	217	
	223	223	223	223	223	223	223	223	223	223	
	229	229	229	229	229	229	229	229	229	229	
	235	235	235	235	235	235	235	235	235	235	
	241	241	241	241	241	241	241	241	241	241	
13*19	247	247	247	247	247	247	247	247	247	247	
11*23	253	253	253	253	253	253	253	253	253	253	
5*37	259	259	259	259	259	259	259	259	259	259	259
	265	265	265	265	265	265	265	265	265	265	265
	271	271	271	271	271	271	271	271	271	271	271
	277	277	277	277	277	277	277	277	277	277	277
	283	283	283	283	283	283	283	283	283	283	283

Figure 11 The emergence of composites in envelope 6x+7 up to 283.

9 Relationship of the sieves to “Uhrenrechner”

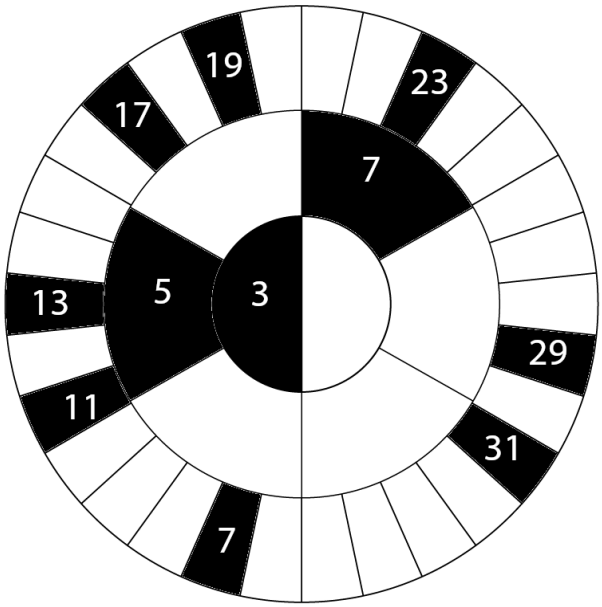


Figure 12 CP_n and C_n as concentric circles like a “Uhrenrechner”

10 How the algorithm developed

The algorithm was originally developed in the year 2007 on paper strips with ideas coming from wave equations.

3	4
L	M

Figure 13 This was the first paper strip.

All multiples (M) of prime number two knock out the even composites (Fig. 13). All uneven numbers still live (L) and only in L there can be prime numbers. Please compare Fig. 13 with Fig. 3. Our question was: If we set our game piece on number three (L) how long does the paper strip has to be, to show the pattern for all natural numbers greater than 3? To us it was like the wave equations. If you overlay two waves the resulting pattern is the first wavelength times the second wavelength, which we learned in school. Here it is wavelength 2 times wavelength 3 equals 6 letters (Fig. 14). We quickly noticed that num-

ber three had turned prime and was not part of the pattern any longer. And that number three turns number nine inside the pattern from L to M (Fig. 14).

4	5	6	7	8	9
M	L	M	L	M	M

Figure 14 Second paper strip.

5	6	7	8	9	10
L	M	L	M	M	M

Figure 15 Third paper strip.

Then we mentally set our game piece to number four. We noticed that the pattern does not change but moves one step, since number four is an M, see Fig. 14 and 15.

This was how we got the idea of moving (Fig.16) through the natural numbers. We made paper strips of length 30 and 210 and worked out, if you start with a single L how the rules must be defined. The beauty of how all natural numbers are governed by periodic patterns lay before us in the year 2007.

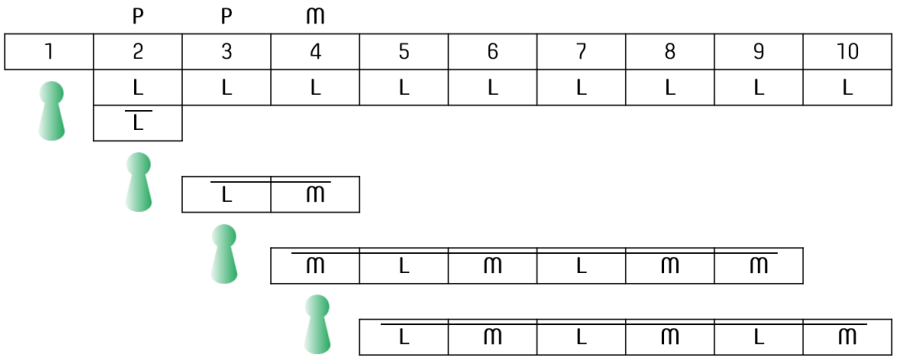


Fig. 16 FA

11 Fractal Dimension of L's in CP_n

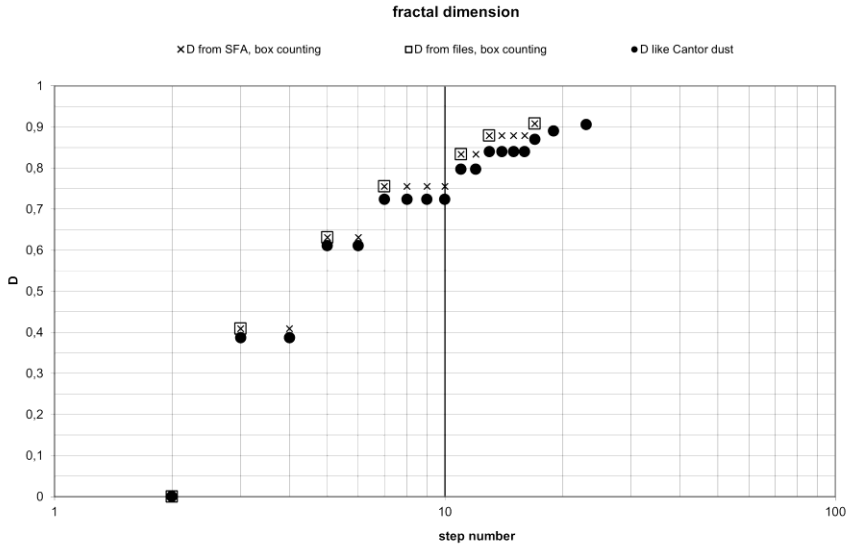


Fig. 17 SFA with FA. The fractal dimension of the L's in CP_n.

The structure of CP_n is fractal (Fig. 17). First, the fractal dimension was calculated from the current SFA with FA. Second, the fractal dimension was calculated from files from a previous implementation of SFA with FA from 2024. Third, the fractal dimension was calculated like Cantor dust.

The fractal dimension (85) from the current implementation of 2025 was identical to the results from Files from 2024 from step number 2 to 17. To look at larger sieves, partial sieves of CP_n were created, but the fractal dimension was skewed (no Fig.).

$$D = \frac{\sum_{i=1}^{k} \left(\log(1/\epsilon_i) - \overline{\log(1/\epsilon)}\right)\left(\log N(\epsilon_i) - \overline{\log N(\epsilon)}\right)}{\sum_{i=1}^{k} \left(\log(1/\epsilon_i) - \overline{\log(1/\epsilon)}\right)^2} \tag{85}$$

11.1 Cantor dust

The Ls are the segments in this Cantor dust. When an L is swapped for an M, it means deleting a segment (86 to 108) (Zeitler & Pagon, 2000). Step number n, CP_n, segments, fractal dimension d_c:

1.)	L	[0, 1]	d_c=ln1/ln1 = NaN	(86)
2.)	LL => LM	[0, ½]	d_c=ln1/ln2 = 0	(87)
3.)	MLMLML => MLMLMM	[1/6, 2/6] [3/6, 4/6]	d_c=ln2/ln6 = 0,387	(88)
4.)	LMLMMM	[0, 1/6] [2/6, 3/6]	d_c=ln2/ln6 = 0,387	(89)
5.)	8 Ls from 30 symbols		d_c=ln8/ln30 = 0,611	(90)
6.)	8 Ls from 30 symbols		d_c=ln8/ln30 = 0,611	(91)

7.)	48 Ls from 210 symbols	d_c=ln48/ln210 = 0,724	(92)
8.)	48 Ls from 210 symbols	d_c=ln48/ln210 = 0,724	(93)
9.)	48 Ls from 210 symbols	d_c=ln48/ln210 = 0,724	(94)
10.)	48 Ls from 210 symbols	d_c=ln48/ln210 = 0,724	(95)
11.)	480 Ls from 2310 symbols	d_c=ln480/ln2310 = 0,797	(96)
12.)	480 Ls from 2310 symbols	d_c=ln480/ln2310 = 0,797	(97)
13.)	5760 Ls from 30030 symbols	d_c=ln5760/ln30030 = 0,840	(98)
14.)	5760 Ls from 30030 symbols	d_c=ln5760/ln30030 = 0,840	(99)
15.)	5760 Ls from 30030 symbols	d_c=ln5760/ln30030 = 0,840	(100)
16.)	5760 Ls from 30030 symbols	d_c=ln5760/ln30030 = 0,840	(101)
17.)	92160 Ls from 510510 symbols	d_c=ln92160/ln510510 = 0,870	(102)
18.)	92160 Ls from 510510 symbols	d_c=ln92160/ln510510 = 0,870	(103)
19.)	1658880 Ls from 9699690 symbols	d_c=ln1658880/ln9699690 = 0,890	(104)
20.)	1658880 Ls from 9699690 symbols	d_c=ln1658880/ln9699690 = 0,890	(105)
21.)	1658880 Ls from 9699690 symbols	d_c=ln1658880/ln9699690 = 0,890	(106)
22.)	1658880 Ls from 9699690 symbols	d_c=ln1658880/ln9699690 = 0,890	(107)
23.)	36495360 Ls from 223092870 symbols	d_c=ln36495360/ln223092870 = 0,906	(108)

The Cantor dust dimension d_c closely matches the dimension D from box-counting (Fig. 17).

Fig.18 shows how the density of Ls decreases and the density of Ms increases.

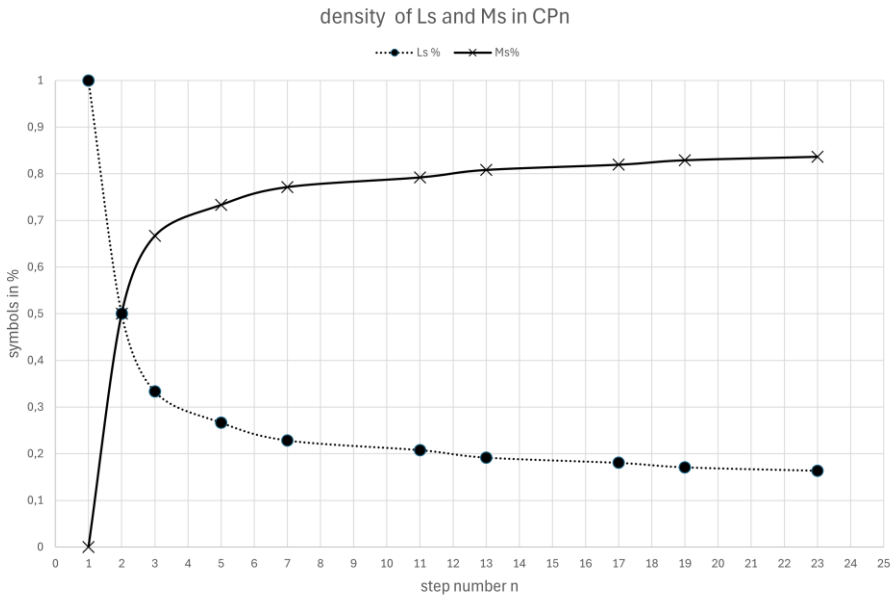


Fig. 18 Density of Ls and Ms in CP_n

12 Conclusion

This paper presents a fully deterministic framework for modeling prime-number **pattern-ing** through symbolic substitution and recursive refinement. The fractal algorithm (FA) acts as a constructive sieve, growing symbolic structures, whose periodic envelopes en-code all prime candidates. Rather than randomness, the observed irregularities in primes and prime gaps arise from high complexity. Every label, envelope, and gap emerge from a rule-based mechanism operating on walksets — ultimately revealing that primes are the result of structured, deterministic propagation.

The Synchronous Factory Automaton (SFA) synchronizes numeric progression with symbolic, fractal pattern dynamics, binding the arithmetic and symbolic realms through recursive update and envelope growth. This symbolic coordination inaugurates a broader program in Generative Sieve Methods, wherein primes are not merely filtered but grown within recursive symbolic architectures.

13 Code

The full Java implementation of the SFA with FA is publicly available at:
https://github.com/cerebrummi/fractalalgorithm.

The Java implementation of counting letters is publicly available at
https://github.com/cerebrummi/letterfa

Java implementation Fig. 19 of the Envelope Equations are publicly available at:
https://github.com/cerebrummi/primeenvelopes

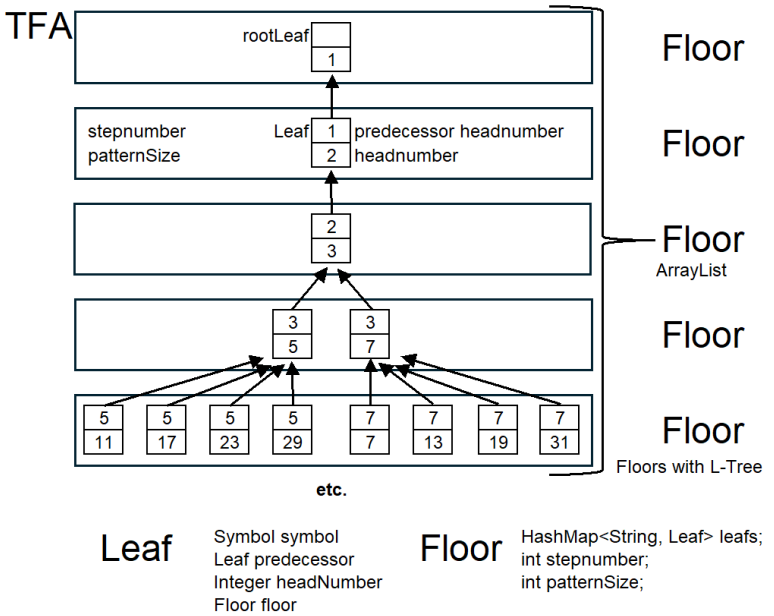


Figure 19 data structure L-tree

14 Outlook

This Outlook is **intentionally speculative**. It reflects my current intuitions (Marcus du Sautoy: Die Musik der Primzahlen) and hypotheses rather than a survey of the literature. The SFA touches many areas of mathematics and probably is a crossroads to link different areas of mathematics.

The fractal algorithm and symbolic machine introduced here provide a deterministic model that reimagines prime number generation. While the current approach yields exhaustive coverage and constructivist propagation, many avenues remain open for exploration:

1. **Formal proof** of the content of this paper with higher mathematics.
2. **Limit Behavior of Twin Primes**: As symbolic density decreases with each FA step, the long-term behavior of patterns like <LML> (twin primes) remains unknown. Future studies could determine whether such patterns vanish in the infinite limit or stabilize.
3. **Prime Gaps and Cluster Structure**: The recursive multiplication of gap-encoding letters (e.g. d) shows that prime gaps follow symbolic propagation rules. Analyzing these rules could lead to deterministic characterizations of large gaps.
4. **Connections to Goldbach's Conjecture:** The M-column in sieve step $n = 2$ identifies precisely the even numbers ≥ 4 for which the full Goldbach conjecture must be verified. This symbolic encoding offers a new perspective: rather than checking arbitrary sums, one could study how these composites emerge structurally across envelopes, potentially revealing deterministic pathways to decomposition.
5. **Symbolic Factorization and Cryptographic Relevance**: Successful factorization of semiprimes using structured latisses opens the door to algorithmic sieving methods for secure key analysis. Scaling such techniques may challenge assumptions in modern cryptosystems.
6. **Mathematical Determinism and Complexity**: The displacement of probabilistic intuition by deterministic recursive symbolic rules invites philosophical and theoretical reconsideration of randomness in number theory. This Ansatz challenges the narrative that prime gaps must arise from statistical behavior.
7. **Riemann Hypothesis**: It is more likely now that it proofs correct.
8. **Chaos theory and symbolic fractals**: The recursive envelope dynamics exhibit hallmarks of chaotic systems — self-similarity, and emergent structure. Modeling the FA as a nonlinear dynamical system could reveal bifurcations or attractor-like regimes in prime gap propagation. Future work might compare fractal dimensions of letter-cluster distributions across sieve steps.
9. **Predator-Prey interactions**: The L (prey) and the M (predator) interactions by fractal procedures copy (multiply L) and change (eat L) are open for modelling.

15 Funding Information

The author funded the research herself. The project started in 2007.

16 References

Herbert Zeitler, Dusan Pagon (2000) Fraktale Geometrie – Eine Einführung
ISBN 978-3-663-08041-1 (eBook)
Marcus du Sautoy (2004) Die Musik der Primzahlen. ISBN 3-423-34299-4

17 Author

Ms. Birke Heeren has a Master of Science in Oceanography from OSU (Oregon, USA) and a Diploma in Biology from Hamburg University (Hamburg, Germany), and is a software engineer (Java, Perl).