Yagel Numbers: Primorial Generalization of Mersenne Numbers

And the minor quest for Yagel Primes

By Mr. Yagel Bar

Date: 2024-10-23

Preface

The idea for Yagel numbers came to me in a sudden moment of inspiration – a curiosity about whether the concept of Mersenne numbers could be generalized using primorials to improve the chances of generating primes. It wasn't born out of a formal academic pursuit but rather out of a personal fascination with prime numbers and a desire to explore them through a fresh lens.

Like many ideas sparked by intuition, it began with optimism – a belief that this new approach might offer insights or even practical breakthroughs in prime number searches. I also won't deny that naming these numbers after myself carried its own appeal; it's human nature to want to leave something behind with your name attached to it.

However, as the work unfolded, so did the realization that competing with the efficiency of the Lucas-Lehmer test for Mersenne primes was futile. What started as an ambitious pursuit transformed into something humbler yet equally meaningful – a mathematical exploration that highlights patterns, raises questions, and invites further study.

This paper is not about claiming superiority over existing methods. Instead, it serves as a thought experiment – a blend of theory and computation that reflects the joy of discovery and the process of learning. It is my hope that this work, regardless of its ultimate applications, sparks curiosity in others and perhaps inspires someone to take the next step.

Disclosure: While all mathematical concepts, algorithms, and interpretations in this paper are entirely my own, I used OpenAI's ChatGPT to assist with wording and structure, improving the readability of the text. As a non-native English speaker, this support was especially valuable for presenting my ideas in a structured and accessible manner without altering their originality.

Mr. Yagel Bar

Abstract

Yagel numbers represent a novel generalization of Mersenne numbers, constructed using primorials, the product of consecutive prime numbers, to amplify prime density. Defined as:

$$Y_k(n) = \left(\prod_{p=1}^{k-1} primes\right) \cdot p_k^n - 1 \qquad n, k \in \mathbb{N}^*$$

where p_k is the k -th prime raised to the power n.

These numbers systematically exclude divisibility by small primes, creating a refined search space for primes. Inspired by the sieve-like behavior of Mersenne numbers, Yagel numbers extend this idea to higher orders, revealing structural biases that favor primality.

This paper explores Yagel numbers' growth patterns, prime densities, and computational feasibility. Through numerical experiments and probabilistic primality testing (Miller-Rabin), Yagel numbers exhibit an unexpectedly high density of primes within specific ranges. While their exponential growth imposes practical limitations for large-scale searches, their observed deviations from expected prime densities raise theoretical questions about prime structures and sieving mechanisms.

Rather than competing with established methods like the Lucas-Lehmer test for Mersenne primes, Yagel numbers offer a thought experiment – combining intuition and computation to examine prime generation through structured filtering. This exploration aims to inspire further research into prime-rich constructs and sieve-based strategies for prime discovery.

Introduction

This paper introduces **Yagel numbers**, a natural generalization of Mersenne numbers designed to amplify prime density using primorial factors. A Yagel number of order k and exponent n , denoted as Y , where k and n are positive natural numbers, is defined as:

$$Y_k(n) = p_{k-1}\# \cdot p_k^n - 1 = \left(\prod_{p=1}^{k-1} primes\right) \cdot p_k^n - 1 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot \cdots \cdot p_{k-1} \cdot p_k^n - 1$$

where the product includes all primes up to the $(k-1)$-th prime and p_k^n is the k -th prime raised to the power n.

Here, the order parameter k refers to the number of prime factors used in the scalar factor, or primorial, while n represents the exponent of the k -th prime. This generalization builds directly on the structure of Mersenne numbers, which are a special case and the first-order Yagel numbers, where:

$$M_n = \mathrm{Y}_1(\mathrm{n}) = 2^{\mathrm{n}} - 1$$

The Intuition

The primary motivation behind Yagel numbers is to construct integers systematically that are not divisible by the first k primes, increasing the likelihood of discovering "Yagel primes". This approach takes inspiration from the Sieve of Eratosthenes and the sieve-like behavior of Mersenne numbers, which exclude all multiples of 2. Yagel numbers extend this principle to higher orders of primes. By increasing k, they eliminate more non-prime candidates, creating a progressively refined space for potential primes.

Numerical Examples

To illustrate, consider a few small examples:

- $Y_2(1) = 2 \cdot 3^1 - 1 = 5 \rightarrow \text{Prime}$
- $Y_2(2) = 2 \cdot 3^2 - 1 = 17 \rightarrow \text{Prime}$
- $\mathrm{Y}_7(2) = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17^2 - 1 = 8{,}678{,}669 \rightarrow \text{Prime}$

These examples demonstrate the effectiveness of Yagel numbers in generating primes at small scales. However, as k and n grow, the rapid increase in magnitude complicates both computation and primality testing, presenting challenges that this work explores in depth.

The tie between Mersenne numbers and Yagel numbers provides a familiar entry point into this new framework. While Mersenne primes rely exclusively on powers of 2, Yagel numbers leverage the multiplicative and exclusionary properties of primorials, offering a broader and potentially richer prime-generating structure. In this paper, I examine the construction, properties, and limitations of Yagel numbers, focusing particularly on their growth behavior, primality characteristics, and the computational implications of their use.

Methodology

Calculation of a Yagel Number

The Yagel numbers formula provides a straightforward calculation for any pair. While small values can be computed by hand, larger values require computational support. The following pseudocode illustrates the procedure.

```
function CalculateYagelNumber(k, n):
    p ← GetKthPrime(k)              // Retrieve the k-th prime
    f ← ProductOfAllPrimes(k - 1)   // Product of the first (k-1) primes
    Y ← f * p^n - 1              // Calculate the Yagel number
    return Y
```

***Algorithm 1**. Calculate Yagel Number (k, n). Return $Y_k(n)$.*

- **Input 1**: Integer $k > 0$
- **Input 2**: Integer n > 0
- **Output**: A potentially very large integer

Primality Test

To verify the primality of Yagel numbers, the **Miller-Rabin primality test** is employed. This probabilistic method balances computational efficiency with reliability, especially for large numbers.

```
function IsPrime(n, rounds):
      let s > 0 and d odd > 0 such that n - 1 = 2^s d
      // by factoring out powers of 2 from n - 1
      repeat rounds times:
            a ← random(2, n - 2)  // n is probable prime to base 1 and n - 1
            x ← a^d mod n
            repeat s times:
                  y ← x^2 mod n
                  if y = 1 and x ≠ 1 and x ≠ n - 1 then
                        // nontrivial square root of 1 modulo n
                        return false
                  x ← y
            if y ≠ 1 then
                  return false
      return true
```

***Algorithm 2**. Test Primality of Yagel Number.*

- **Input 1**: n > 2, an odd integer to be tested for primality
- **Input 2**: rounds > 0, the number of rounds of testing to perform
- **Output**: False if composite, True if probably prime

Note: The computational constraint of Miller-Rabin test is reviewed in the discussion section.

Bulk Generation of Yagel Numbers

Generating multiple Yagel numbers efficiently involves looping over a range of orders (k) and exponents (n), subject to constraints like maximum digit length. The primality test is performed with seven iterations to reduce the false-positive rate to approximately 0.0061%.

```
function CalculateRangeOfYagelNumbers(min_k, max_k, max_digits):
    results ← []
    for k ← min_k to max_k - 1 do:
        n ← 1
        while true do:
            Y ← CalculateYagelNumber(k, n)
            digits ← CountDigits(Y)

            if digits > max_digits:
                break

            is_prime ← IsPrime(Y, 7) // use 7 rounds
            results.append((k, n, Y, digits, is_prime))

            n ← n + 1
    return results
```

***Algorithm 3**. Generate Range of Yagel Numbers.*

- **Input**: Inclusive lower bound min_k, exclusive upper bound max_k, and max_digits.
- **Output**: A list of tuples containing $Y_k(n)$ for each valid combination.
- **Example**: Generate Yagel numbers for $k \in [1, 51)$ and up to 5,000 digits (< 10^{5000}).

Results and Observations

Yagel Numbers Per Order and Exponent – Examples

Order (k)	**k_{th} Prime**	**Power (n)**	**$Y_k(n)$ #Digits**	**$Y_k(n)$ Prime**	**Yagel Number - $Y_k(n)$**
1	2	1	1		1
1	2	2	1	V	3
1	2	3	1	V	7
1	2	10	4		1,023
1	2	107	33	V	162,259,276,829,213,363,391,578,010,288,127
1	2	11,213	3,376	V	2,814,…,392,191
1	2	16,606	4,999		8,018,…,904,063
2	3	1	1	V	5
2	3	2	1	V	7
2	3	3	2	V	13
2	3	10	6		118,097
2	3	131	63	V	636,669,967,197,883,491,262,127,134,516,276,007,946 ,623,425,982,895,419,891,235,893
2	3	9,204	4,392	V	530,955,…,888,161
2	3	10,476	4,999		4,200,…,663,841
3	5	1	2	V	29
3	5	2	3	V	149
3	5	3	3		749
3	5	10	8		58,593,749
3	5	479	336	V	384,399,…,468,749
3	5	4,418	3,089	V	67,240,…,593,749
3	5	7,150	4,999		2,592,…,093,749
20	73	1	27		573,657,473,228,859,495,079,173,569
20	73	2	29	V	41,876,995,545,706,743,140,779,670,609
20	73	3	31		3,057,020,674,836,592,249,276,915,954,529
20	73	167	335	V	1,175,…,229,729
20	73	966	1,814	V	7,331,…,253,009
20	73	2,686	4,998		237,276,…,335,889
27	103	1	41		23,984,823,528,925,228,172,706,521,638,692,258,396, 209
27	103	2	43		2,470,436,823,479,298,501,788,771,728,785,302,614,8 09,629
27	103	3	45	V	254,454,992,818,367,745,684,243,488,064,886,169,325 ,391,889
27	103	1,561	3,181	V	2,546,…,788,209
27	103	2,464	4,998		995,453,…,108,669
38	163	1	64		5,766,152,219,975,951,659,023,630,035,336,134,306,5 65,384,015,606,066,319,856,068,809
38	163	2,216	4,964	V	57,072,…,826,669
38	163	2,231	4,997		86,944,…,381,689

47	211	1	85		1,645,783,550,795,210,387,735,581,011,435,590,727,981,167,322,669,649,249,414,629,852,197,255,934,130,751,870,909
47	211	32	157	V	1,858,…,360,009
47	211	53	206	V	11,996,…,414,109
47	211	93	299	V	11,229,…,518,109
47	211	2,115	4,998		561,672,…,667,309
48	223	1	87		367,009,731,827,331,916,465,034,565,550,136,732,339,800,312,955,331,782,619,462,457,039,988,073,311,157,667,212,929
48	223	63	233	V	14,440,…,907,969
48	223	66	240	V	160,137,…,148,989
48	223	658	1,630	V	2,517,…,889,789
48	223	2,092	4,997		74,155,…,538,109
50	229	1	92		19,078,266,889,580,195,013,601,891,820,992,757,757,219,839,668,357,012,055,907,516,904,309,700,014,933,909,014,729,740,189
50	229	2	94		4,368,923,117,713,864,658,114,833,227,007,341,526,403,343,284,053,755,760,802,821,371,086,921,303,419,865,164,373,110,503,509
50	229	3	97		1,000,483,393,956,475,006,708,296,808,984,681,209,546,365,612,048,310,069,223,846,093,978,904,978,483,149,122,641,442,305,303,789
50	229	281	753	V	108,261,…,764,189
50	229	379	984	V	198,770,…,044,589
50	229	1,979	4,760	V	10,842,…,124,589
50	229	2,080	4,998		239,059,…,551,109

***Table 1**. Yagel Numbers Per Order k and Exponent n – Examples.*

Note: Due to the impracticality of presenting the entire dataset within this paper, Table 1 provides a representative subset of Yagel numbers – some prime, some composite, spanning both high and low magnitudes. The following Figure 1 visualizes all Yagel primes identified in the specified range, offering insights into their distribution. For replication and further analysis, the complete dataset, covering values for $k \leq 50$ *and up to 5000 digits, may be hosted online in future releases at yagelnumbers.org .*

Distribution of Yagel Primes per Order k

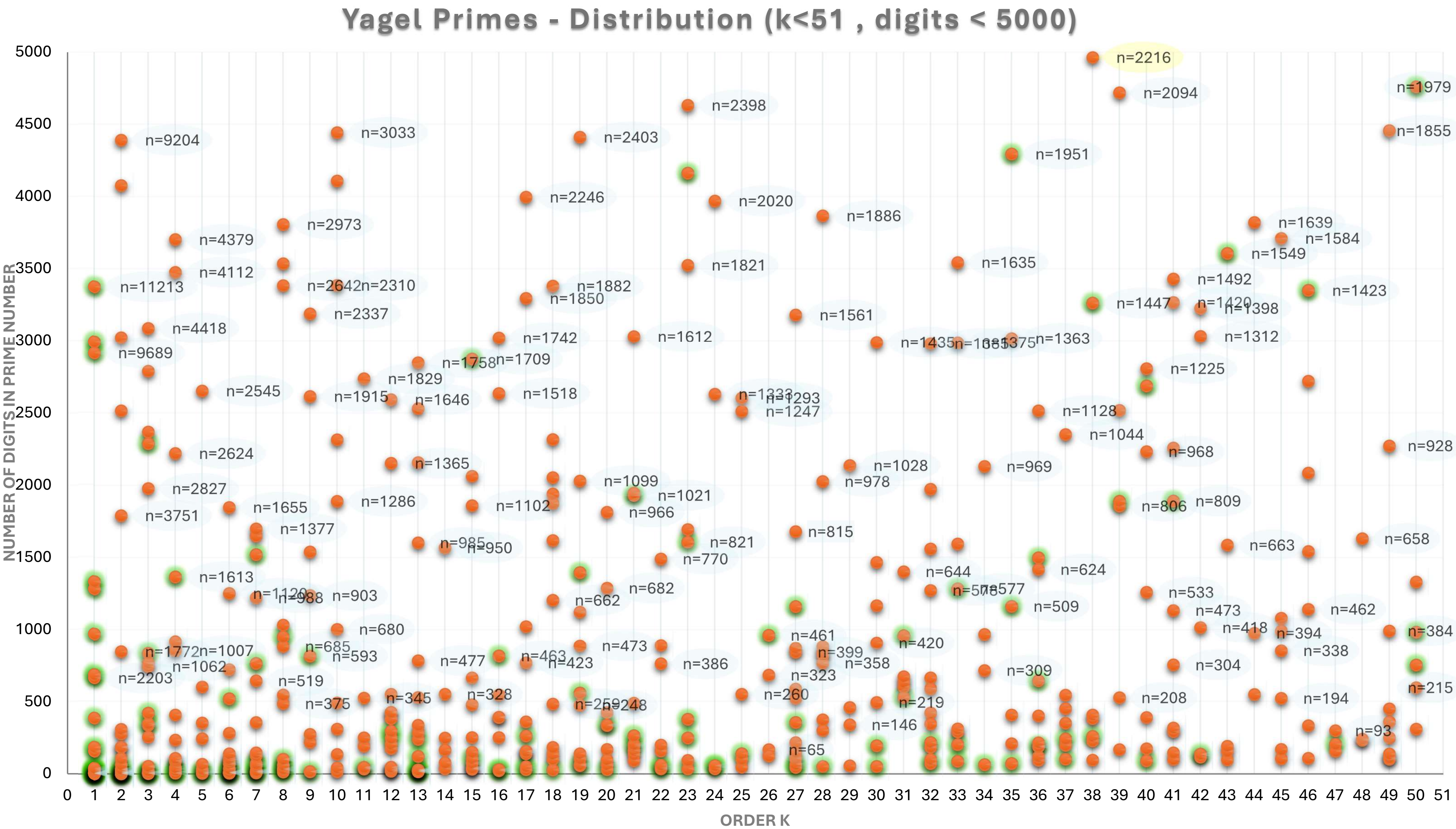

***Figure 1**. All Yagel primes, for the first 50 orders and up to 5,000-digit numbers.*

Note: Marked with **green glow**: when n itself is also a prime number.

Primes Distribution $k \leq 50$

I counted the number of Yagel primes found for each order k that has more than 100 digits, and up to 5000 digits, and binned them under same prime count.

Primes Count	Number of Orders	Orders
0	0	
1	0	
2	1	24
3	5	29, 34, 44, 47, 48
4	5	5, 11, 14, 26, 43
5	3	22, 25, 35
6	8	20, 30, 31, 38, 39, 40, 42, 50
7	8	6, 9, 16, 23, 33, 37, 45, 46
8	3	15, 28, 36
9	7	4, 7, 10, 17, 19, 21, 49
10	4	8, 12, 13, 41
11	5	1, 2, 18, 27, 32
12	1	3

Table 2. *Yagel primes distribution* $k \leq 50$.

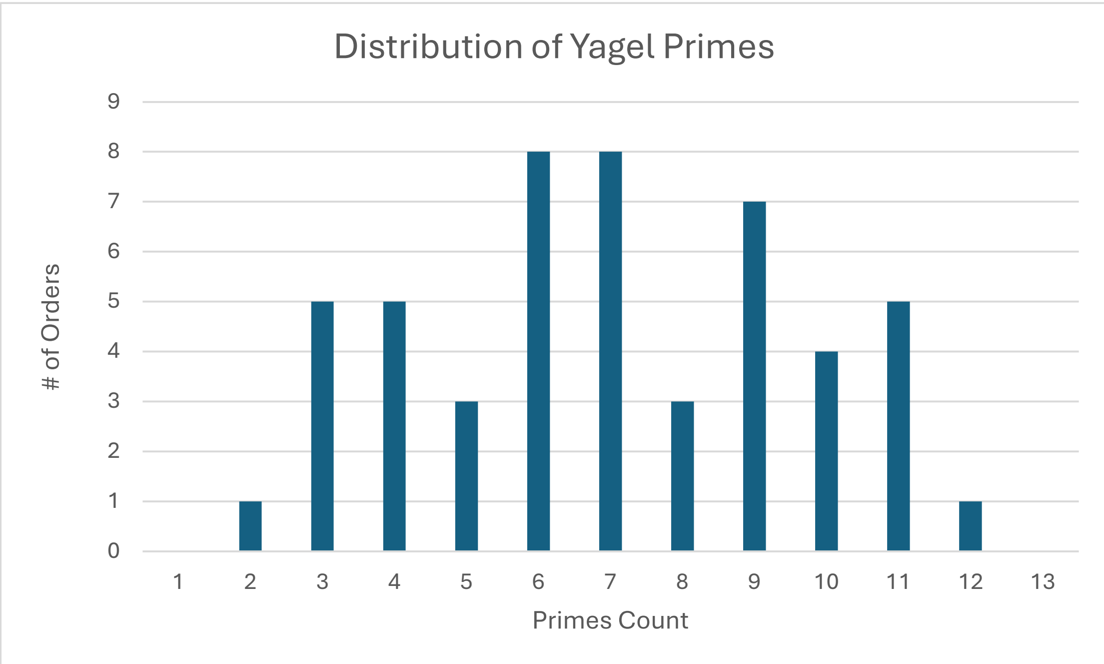

Figure 2. *Distribution of number of Yagel primes counted at the 100 digits to 5000 digits range,* $k \leq 50$. *Histogram.*

Observations

- An interesting characteristic observed in Table 1 is the consistency of the **least significant digits** (LSD) among Yagel numbers of higher orders. For $k \geq 3$, the unit digit is always 9, and the tens digit is always even, due to the result of subtracting 1 from multiples of composite factors involving both 2, 3 and 5. For orders $k < 3$, the LSD is never 9, due to the opposite reason.
- From Figure 1, it can be seen that $k = 47$ and $k = 48$, have **only** 3 primes each under 5000 digits.
- The **largest** Yagel prime found in the range is $Y_{38}(2216)$ – Marked in yellow.
- From Figure 2, it is noticeable that Yagel primes are distributed around 7 primes in the 100 to 5000 digits range, but with no correlation to the value of k itself.

Discussion

Growth Characteristics

In analyzing the growth behavior of Yagel numbers, it becomes evident that both increasing k and increasing n lead to substantial growth, albeit at different rates.

Growth Effects of k

Increasing the order parameter k primarily impacts the base factor f in the formula:

$$Y_k(n) = f \cdot p^n - 1$$

Where f is the product of the first $k - 1$ primes. The growth of f follows the asymptotic relation: [1]

$$f \sim e^{(1+o(1))k \ln k}$$

implying it grows faster than any polynomial but slower than pure exponential functions. Additionally, the k-th prime p_k in contributes to the magnitude of the expression through its own growth, particularly when raised to a power n.

Growth Effects of n

By contrast, increasing the exponent n affects the exponent in p_k^n directly, leading to true exponential growth. Even modest increases in n can result in dramatic size escalation.

For example:

- If $p_k = 17$ and n increases from 1 to 2, p_k^n jumps from 17 to 289
- When n increases further to 3, it leaps to 4913.

Comparison the Growth Rates: k vs. n

While both larger k and n increase $\mathrm{Y_k(n)}$, the growth caused by n has a dominates because exponential scaling in p_k^n outpaces the quasi-exponential growth caused by the product of primes as k increases.

Summary of Growth Trends:

Growth driven by n (exponential) is **significantly larger** than growth driven by k (quasi-exponential)

Sieve Density and Prime Bias

The central hypothesis behind Yagel numbers is that excluding divisibility by small primes increases the likelihood of generating primes. This concept is rooted in the sieve-like behavior inspired by the **Sieve of Eratosthenes**.

By removing all numbers divisible by primes up to the $(k-1)$-th prime, the probability of primality should increase among the remaining candidates. This can be modeled as:

$$f = \prod_{p \leq k} \left(1 - \frac{1}{p}\right)$$

Slow Convergence of Sieve Density

While this formula converges to zero as $k \to \infty$, the convergence is slow

$$\prod_{p \leq k} \left(1 - \frac{1}{p}\right) \approx \frac{e^{-\gamma}}{\ln{(k)}}$$

where γ is the **Euler-Mascheroni constant**. [2]

For example, to remove 95% of all natural numbers requires using the 7398th prime. At that stage, the primorial of this prime exceeds $10^{32,000}$, making computation infeasible.

Practical Implications

Because of this slow convergence, the computational cost of filtering candidates outweighs the benefits of sieve-like cleanup when searching for very large primes. This observation highlights an inherent limitation of Yagel numbers for large-scale prime searches.

Density of Primes

To evaluate whether Yagel primes occur more frequently than random primes, we compare observed counts against predictions based on the **Prime Number Theorem** (PNT).

For a sufficiently large number N, the probability that a **random integer** is prime approximates: [3]

$$P(N) \approx \frac{1}{\ln(N)}$$

This result suggests that primes become **sparser** as numbers grow larger. To estimate the **expected number of primes** within a given range, we integrate this approximation.

Deriving the Expected Value Formula

The **cumulative count of primes** up to a number x is represented by the **prime-counting function** $\pi(x)$. The Prime Number Theorem asserts that: [4]

$$\pi(x) \sim \frac{x}{\ln(x)} \ as\ x \to \infty$$

For a more **precise approximation**, we use the **logarithmic integral function**: [3]

$$\mathrm{Li(x)} = \int_2^x \frac{1}{\ln(t)} \mathrm{dt}$$

which better models prime density than $\frac{x}{\ln(x)}$.

The **expected number of primes** within a range can therefore be estimated as: [3]

$$E(x) = Li(10^{\max\_digits}) - Li(10^{\min\_digits})$$

This approach reflects the asymptotic density of primes as numbers grow exponentially within specified ranges.

Applying the Formula to the Observed Range

For the range analyzed in this paper – numbers with **100 to 5000 digits**, (as shown in Table 2), we approximate:

$$Li(10^{5000}) - Li(10^{100})$$

Simplifying this logarithmic interval, we focus on the **logarithmic approximation**: [4]

$$\ln(5000) - \ln(100) \approx 3.9$$

This result reflects the **expected density of primes per logarithmic scale** rather than the **total count of primes** in the range. The observed counts of **Yagel primes** can then be **compared against this density** to determine whether their occurrence demonstrates **bias** or **deviation** from random expectations.

Observed Data

Yagel numbers grow systematically by multiplying primes, creating a sequence that differs fundamentally from the **random distribution** of integers assumed in the **Prime Number Theorem**. Despite this difference, their growth rate—dominated by the **largest prime factor raised to a power** (p_k^n)—ensures that Yagel numbers span ranges comparable to those modeled by **logarithmic intervals**.

This growth pattern allows us to **approximate their density** using predictions derived from the Prime Number Theorem and logarithmic integrals, which estimate the **expected number of primes** within exponentially growing ranges.

By comparing the **observed distribution of Yagel primes** to this **expected density**, we test whether their **structured construction** introduces a **bias favoring primes** beyond what is predicted for randomly distributed integers. Table 3 summarizes the results:

Primes Count	Orders Count
< 4	6
$= 4$	5
> 4	39

***Table 3**. Distribution of Yagel $k \leq 50$ by bins less than, equal to, and greater than 4.*

Key Findings

The data show a higher-than-expected prime count, with 39 cases exceeding the predicted value of 4. Yagel numbers show deviations from the distribution expected for randomly selected integers.

Statistical Evidence of Bias or Structured Patterns

The deviation observed in Table 3 points to structural features of Yagel numbers that favor primality. Potential explanations include:

- **Mathematical Bias**: Their multiplicative construction may enhance prime-rich patterns.
- **Systematic Filtering**: Avoidance of small-prime factors inherently reduces composite candidates.
- **Prime Density Oscillations**: Correlations with density peaks could amplify observed clustering.

While these biases provide interesting insights, their practical significance diminishes as k increases due to slow convergence.

Primality Testing of Mersenne Primes: An Interlude to the Lucas-Lehmer Test

The Special Structure of Mersenne Primes

Mersenne numbers, defined as:

$$M_p = 2^p - 1$$

where p is prime, exhibit unique mathematical properties that allow for exceptionally efficient primality testing.

Modulo $2^p - 1$: Periodicity and Computational Efficiency

A key property of Mersenne numbers is their **binary structure**, consisting of p consecutive 1's in binary representation. This structure grants them a **cyclic residue system** when performing modular arithmetic, simplifying reductions and enabling rapid computation.

In particular, modular arithmetic on Mersenne numbers:

- Wraps cleanly due to the form $2^p - 1$.
- Allows reductions that exploit bitwise operations instead of general-purpose arithmetic.

Comparing Efficiency: Lucas-Lehmer vs. Miller-Rabin

To highlight the computational advantage of the Lucas-Lehmer test, we compare it with the Miller-Rabin primality test using a 10,000-digit number

Miller-Rabin Primality Test (Probabilistic)

For a number n with approximately 10,000 decimal digits:

$$n \approx 10^{10,000}$$

The time complexity of the Miller-Rabin test is: [5]

$$O(k \cdot \log^3 n)$$

where k is the number of iterations (witnesses) required. Using $k = 10$ rounds, we calculate:

$$log\, n = log(10^{10,000}) = 10,000$$

$$(log\, n)^3 = 10,000^3 = 10^{12}$$

Total complexity:

$$O(10 \cdot 10^{12}) = 10^{13}$$

Thus, Miller-Rabin requires approximately **10 trillion operations** to test primality probabilistically.

Lucas-Lehmer Primality Test (Deterministic)

For the same 10,000-digit number, the exponent is approximately:

$$p = log_2(10^{10,000}) \approx 33{,}219$$

The time complexity of the Lucas-Lehmer test is: [6]

$$O(p \cdot log^2 p)$$

We calculate:

$$log(33{,}219) \approx 4.5$$

$$4.5^2 \approx 20.25$$

Total complexity:

$$O(33{,}219 \cdot 20.25) \approx 6.7 \times 10^5$$

Comparative Analysis

- Miller-Rabin Test (Probabilistic) $\approx 10^{13}$ operations.
- Lucas-Lehmer Test (Deterministic) $\approx 10^6$ operations.

The Lucas-Lehmer test is therefore **7 orders of magnitude faster** than the Miller-Rabin test for numbers of similar size.

Key Observations

Structural Advantage of Mersenne Numbers:

- The binary periodicity of $2^p - 1$ simplifies modular arithmetic, enabling deterministic testing.
- This property eliminates the need for probabilistic tests and reduces computational overhead dramatically.

Limitations of Yagel Numbers:

- Yagel numbers lack the binary periodicity that Mersenne primes leverage.
- As a result, they rely on probabilistic tests like Miller-Rabin, making primality verification computationally expensive for large sizes.

Conclusion

This paper introduced **Yagel numbers**, a primorial-based generalization of Mersenne numbers, as part of an exploratory effort to uncover new patterns in prime number construction. The inspiration for this work stemmed from a simple yet intriguing idea – whether systematically filtering out divisors of small primes could create a prime-rich framework similar to Mersenne primes.

The journey began with optimism, supported by early numerical evidence showing higher-than-expected densities of primes among Yagel numbers. These results, along with theoretical insights about sieve density, suggested that structural biases inherent to Yagel numbers may promote primality. However, deeper investigation revealed limitations tied to computational growth and primality testing.

The **key findings** can be summarized as follows:

- The observation that 39 orders (k) outperform the expectation suggests that Yagel numbers exhibit biases favoring primes due to their structured construction. While these results differ from purely random distributions, further statistical testing would be required to determine whether this reflects persistent patterns or local fluctuations.
- Their **growth characteristics**, driven by quasi-exponential scaling in k and true exponential scaling in n, lead to rapid size escalation, making large-scale exploration computationally expensive.
- Unlike Mersenne numbers, which exploit **binary periodicity** to enable deterministic primality testing with the **Lucas-Lehmer test**, Yagel numbers lack such structural shortcuts. Consequently, they rely on **probabilistic tests** like **Miller-Rabin**, which become computationally prohibitive for larger numbers.

While this exploration did not yield a computational competitor to Mersenne primes, it raised questions worth pursuing:

- Are there deterministic algorithms that can leverage the multiplicative structure of Yagel numbers, similar to how Lucas-Lehmer does for Mersenne primes?
- Could hybrid frameworks combining **primorials** and **arithmetic progressions** yield **new classes of prime-rich sequences**?
- What mathematical mechanisms might explain the deviations from random prime density expectations observed in Yagel numbers?

The limitations uncovered in this study do not diminish its value. Instead, they highlight the **importance of speculative research** – ideas pursued out of curiosity rather than expectation. While Yagel numbers may not revolutionize prime discovery, they contribute to our understanding of prime patterns and sieve-based methods.

In the end, this paper stands as both a **personal exploration** and an invitation for others to **build upon its foundation**, whether by refining algorithms, proving conjectures, or simply searching for the next mathematical surprise

References

[1] G. H. Hardy and J. E. Littlewood, "Some problems of ‘Partitio numerorum’; III: On the expression of a number as a sum of primes," *Acta Math,* vol. 44, pp. 1-70, December 1923.

[2] J. B. Rosser and L. Schoenfeld, "Approximate Formulas for Some Functions of Prime Numbers," *Illinois Journal of Mathematics,* vol. 6, no. 1, pp. 64-94, 1962.

[3] H. Davenport, Multiplicative Number Theory, 2nd ed., New York, New York: Springer-Verlag, 2000.

[4] G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers, Oxford: Oxford University Press, 2008, pp. 6-14.

[5] M. Rabin, "Probabilistic Algorithm for Testing Primality," *Journal of Number Theory,* vol. 12, no. 1, pp. 128-138, 1980.

[6] H. Riesel, Prime Numbers and Computer Methods for Factorization, 2nd ed., Boston, Massachusetts: Birkhäuser, 1994.