

# Empirical Investigation of an Open Conjecture:

Let  $A$  be a set of positive integers. For each  $n$  in  $A$ , choose some collection  $X_n$

Agentic NL→Lean 4 Pipeline  
Job #22

April 26, 2026

## Abstract

This report documents the empirical investigation of an open mathematical conjecture that could not be formally proved or disproved in Lean 4 with Mathlib. Numerical experiments were conducted to gather evidence for or against the conjecture. The empirical verdict is: **Inconclusive**. The conjecture remains formally open.

## 1 Conjecture Statement

### Conjecture 1.

*Let  $A$  be a set of positive integers. For each  $n$  in  $A$ , choose some collection  $X_n$  of forbidden residue classes modulo  $n$ . Define  $B$  to be the set of positive integers  $m$  such that, for every  $n$  in  $A$  with  $n$  strictly less than  $m$ , the residue class of  $m$  modulo  $n$  is not one of the forbidden classes in  $X_n$ . Then,  $B$  must have a logarithmic density; i.e., the following limit exists: the limit, as  $x$  tends to infinity, of the reciprocal of  $\log(x)$  multiplied by the sum of the reciprocal of all elements in  $B$  strictly smaller than  $x$ .*

## 2 Status

**Formal Status:** OPEN — no Lean 4 proof or disproof was found.

**Empirical Verdict:** **Inconclusive**

The pipeline attempted formal verification in Lean 4 with Mathlib but was unable to produce a compiling proof or disproof. Empirical testing was then conducted to gather numerical evidence.

## 3 Basic Empirical Testing

The following output was produced by the basic numerical experiment:

```
=== EXPERIMENT PLAN ===
```

```
Conjecture: Given A ⊆ ℕ_{>0}, and for each n ∈ A a set X_n ⊆ ℤ/nℤ of
```

'forbidden' residues, define  
 $B = \{ m \in \mathbb{Z}_{>0} : \text{for every } n \in A \text{ with } n < m, (m \bmod n) \in X_n \}$ .  
Then the logarithmic density  
 $(x) = (1/\log x) * \sum_{b \in B, b < x} 1/b$   
converges as  $x \rightarrow \infty$ .

This is a generalized 'sieve by residues' problem. Classical results (Davenport-óErds, Besicovitch, óErds) show that B is always a union of 'B-sets' (sets closed under taking multiples in the prime-covering case) and that logarithmic density is more robust than natural density. Natural density of such sifted sets can FAIL to exist (e.g. classical Besicovitch counterexamples for multiples of primitive sets) but the conjecture here is about the gentler LOGARITHMIC density.

We run five independent empirical probes:

1. Random sieves. Random A, random  $X_n$ . Compute  $(x)$  along a geometric ladder of x-values and check whether the sequence  $(x_k)$  is Cauchy-like (fluctuations shrinking).
2. Adversarial sieves. Constructions designed to make natural density oscillate (prime sieves with growing forbidden sets, 'Besicovitch-like' multiples sieves). Check whether the logarithmic density nonetheless stabilizes.
3. Known closed-forms. Sieves where the limit is analytically known (e.g. sieve of Eratosthenes  $\rightarrow$  primes  $\rightarrow$  log-density 0; single modulus n with k forbidden classes  $\rightarrow$   $(n-k)/n$ ). Verify numerical convergence to the correct value.
4. Empty / trivial edge cases ( $A = \emptyset$ ,  $X_n = \emptyset$ ,  $X_n = \text{all classes}$ ).
5. Asymptotic rate of convergence. Fit  $|(x) - (x_{\text{last}})|$  vs  $1/\log x$  and test whether the oscillations decay.

--- Probe 1: Random sieves ---

Trials: 40, N=400000

Tail fluctuation (max - min over last 15 sample points):

mean = 0.05640, max = 0.07353, median = 0.05763

--- Probe 2: Adversarial / Besicovitch-style sieves ---

|  |     |         |      |
|--|-----|---------|------|
| primes sieve (forbid 0 mod p, p<=50)       | (N) | 0.31145 | tail |
| fluct = 3.49e-02                           |     |         |      |
| covering-like (2,3,4,6,12)                 | (N) | 0.14918 | tail |
| fluct = 3.02e-02                           |     |         |      |
| Besicovitch-like (0 mod sparse composites) | (N) | 0.81839 | tail |
| fluct = 2.72e-02                           |     |         |      |
| primes with growing forbidden sets         | (N) | 0.11629 | tail |
| fluct = 2.35e-02                           |     |         |      |

--- Probe 3: Known closed-form comparisons ---

|                        |                 |            |             |
|------------------------|-----------------|------------|-------------|
| n=2, forbid 1 classes  | expected=0.5000 | (N)=0.5880 | err =0.0880 |
| n=3, forbid 1 classes  | expected=0.6667 | (N)=0.7507 | err =0.0841 |
| n=5, forbid 2 classes  | expected=0.6000 | (N)=0.6967 | err =0.0967 |
| n=7, forbid 3 classes  | expected=0.5714 | (N)=0.6794 | err =0.1080 |
| n=10, forbid 4 classes | expected=0.6000 | (N)=0.7126 | err =0.1126 |
| n=11, forbid 5 classes | expected=0.5455 | (N)=0.6703 | err =0.1248 |
| n=13, forbid 6 classes | expected=0.5385 | (N)=0.6698 | err =0.1313 |

```

forbid 0 mod 3 and 0 mod 5  expected=0.5333  (N)=0.6379  |err|=0.1046

--- Probe 4: Edge cases ---
A = empty:                (N) = 1.04475  (expected 1)
forbid all mod 2:         (N) = 0.11629  (expected 0; finite B)
empty X_n for n<=100:     (N) = 1.04475  (expected 1)

--- Probe 5: Convergence-rate analysis ---
log-log slope of |(x)-(N)| vs 1/log x over 40 trials:
  mean slope = 8.512, std = 0.001
  (positive slope  fluctuation shrinks as x grows  consistent with
   convergence)

Sign test: in 40/40 trials, early std > late std (z = 6.32)
Mean std early = 3.22e-02,  mean std late = 1.46e-02

=== SUMMARY ===
Random trials with tail fluct < 1e-2 : 0/40
Random trials with tail fluct < 1e-3 : 0/40
Max closed-form error (Probe 3)       : 0.1313
Mean log-log decay slope (Probe 5)    : 8.512  (positive  convergence)

=== VERDICT ===
INCONCLUSIVE: numerical evidence was mixed; see per-probe stats.

```

### 4 Experiment Code (Basic)

```

import matplotlib
matplotlib.use("Agg")
import numpy as np
import matplotlib.pyplot as plt
import random
import math
from itertools import combinations

print("===_EXPERIMENT_PLAN_===")
print("""
Conjecture: Given A  Z_{>0}, and for each nA a set X_n  Z/nZ of
'forbidden' residues, define
  B = { m  Z_{>0} : for every nA with n<m, (m mod n)  X_n }.
Then the logarithmic density
  (x) = (1/log x) * Σ_{bB, b<x} 1/b
converges as x →∞ .

This is a generalized 'sieve by residues' problem. Classical results
(Davenport-öErds, Besicovitch, öErds) show that B is always a union of
'B-sets' (sets closed under taking multiples in the prime-covering case)
and that logarithmic density is more robust than natural density.
Natural density of such sifted sets can FAIL to exist (e.g. classical
Besicovitch counterexamples for multiples of primitive sets) but the
conjecture here is about the gentler LOGARITHMIC density.

```

```

We run five independent empirical probes:
1. Random sieves. Random A, random X_n. Compute (x) along a
   geometric ladder of x-values and check whether the sequence
   (x_k) is Cauchy-like (fluctuations shrinking).
2. Adversarial sieves. Constructions designed to make natural
   density oscillate (prime sieves with growing forbidden sets,
   'Besicovitch-like' multiples sieves). Check whether the
   logarithmic density nonetheless stabilizes.
3. Known closed-forms. Sieves where the limit is analytically known
   (e.g. sieve of Eratosthenes → primes → log-density 0; single
   modulus n with k forbidden classes → (n-k)/n). Verify numerical
   convergence to the correct value.
4. Empty / trivial edge cases (A = , X_n = , X_n = all classes).
5. Asymptotic rate of convergence. Fit |(x) - (x_last)| vs 1/log x
and test whether the oscillations decay.
"""

random.seed(20260421)
np.random.seed(20260421)

# ----- core sieve engine (vectorized) -----
def sieve(Xs, N):
    """Xs: dict n → iterable of forbidden residues mod n, all n < N.
    Returns boolean mask of length N+1; True at index m iff m ∈ B."""
    alive = np.ones(N+1, dtype=bool)
    alive[0] = False
    for n, Xn in Xs.items():
        if n < 1: continue
        for r in Xn:
            r = int(r) % n
            # flag all m > n with m mod n == r
            start = n + (r - n) % n # smallest m > n with m ≡ r (mod n)
            if start <= n:
                start += n
            if start > N:
                continue
            alive[start::n] = False
    alive[0] = False
    return alive

def log_density_curve(mask, xs):
    """For each x in xs, compute (1/log x) * sum_{b<x, b in B} 1/b."""
    N = len(mask) - 1
    idx = np.arange(1, N+1)
    contrib = np.where(mask[1:], 1.0/idx, 0.0)
    csum = np.concatenate([[0.0], np.cumsum(contrib)]) # csum[k] = sum_{b<=
    k} 1/b * [bB]
    out = []
    for x in xs:
        x = int(x)
        if x < 2:
            out.append(np.nan); continue
        s = csum[x-1] # sum over b < x

```

```

        out.append(s / math.log(x))
    return np.array(out)

def geomladder(N, steps=40, lo=200):
    return np.unique(np.geomspace(lo, N-1, steps).astype(int))

# ----- Probe 1: random sieves -----
print("\n---_Probe_1:_Random_sieves_---")
N = 400_000
xs = geomladder(N, 50)
trials_curves = []
fluct = []
n_random = 40
for t in range(n_random):
    Xs = {}
    # A is a random subset of [2, 300]; for each nA forbid a random subset
    # of residues
    A = random.sample(range(2, 300), k=random.randint(5, 80))
    for n in A:
        k = random.randint(0, max(1, n//2))
        Xs[n] = random.sample(range(n), k)
    mask = sieve(Xs, N)
    curve = log_density_curve(mask, xs)
    trials_curves.append(curve)
    tail = curve[-15:]
    fluct.append(float(np.nanmax(tail) - np.nanmin(tail)))

trials_curves = np.array(trials_curves)
print(f"Trials:_{n_random},_N={N}")
print(f"Tail_fluctuation_(max_min_over_last_15_sample_points):")
print(f"_{np.mean(fluct):.5f},_{np.max(fluct):.5f},_median=_{np.median(fluct):.5f}")

# ----- Probe 2: adversarial sieves -----
print("\n---_Probe_2:_Adversarial_/_Besicovitch-style_sieves_---")
adv_curves = {}

# (a) Multiples of primes up to 50 -> B = {1} primes>50 (log-density 0)
primes = []
is_p = np.ones(501, dtype=bool); is_p[:2]=False
for i in range(2, 501):
    if is_p[i]:
        primes.append(i)
        is_p[i*i::i] = False
Xs = {p: [0] for p in primes if p <= 50}
mask = sieve(Xs, N)
adv_curves["primes_sieve_(forbid_0_mod_p,_p<=50)"] = log_density_curve(mask,
    xs)

# (b) Covering-system attempt (öErds): n=2,3,4,6,12 with residues chosen
Xs = {2:[0], 3:[0], 4:[1], 6:[1], 12:[11]}
mask = sieve(Xs, N)
adv_curves["covering-like_(2,3,4,6,12)"] = log_density_curve(mask, xs)

```

```
# (c) Besicovitch-style: forbid 0 mod n for n in a sparse set of composites
comps = [n for n in range(2, 400) if not is_p[n]]
random.shuffle(comps)
sparse = sorted(comps[:60])
Xs = {n: [0] for n in sparse}
mask = sieve(Xs, N)
adv_curves
# ... [truncated]
```

## 5 Conclusion

The conjecture remains formally open. Numerical experiments were **inconclusive** — neither strong support nor clear counterexamples were found. Further investigation (both formal and empirical) is warranted.