Empirical Investigation of an Open Conjecture: There are infinitely many pairs of prime numbers that differ by exactly 2.

Agentic NL→Lean 4 Pipeline
Job #5

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.
There are infinitely many pairs of prime numbers that differ by exactly 2.

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: There are infinitely many pairs of primes (p, p+2) - the
Twin Prime Conjecture.

Since 'infinitely many' cannot be verified by finite computation, we
gather empirical evidence via the following tests:

```

```
1. DIRECT ENUMERATION: Count twin primes up to increasing bounds N
   (10^3, 10^4, ..., up to ~10^7). If twin primes keep appearing and
   never stop, that's supportive evidence.

2. HARDY-LITTLEWOOD CONJECTURE: The strong form predicts
      pi_2(N) ~ 2 * C_2 * N / (ln N)^2
   where C_2   0.6601618 is the twin prime constant.
   We compare observed counts to this asymptotic and compute the ratio.
   If the ratio stays near 1, it's strong evidence the density does
   not fall to zero, i.e. infinitely many twins.

3. GAP BETWEEN CONSECUTIVE TWIN PRIMES: We track the largest observed
   gap vs N. If twin primes were finite, there would eventually be a
   last one and gaps would blow up. We look for steady growth
   consistent with Cramér-like heuristics rather than termination.

4. RANDOM SAMPLING IN LARGE INTERVALS: In windows [M, M+L] for very
   large M, we count twin primes and compare to the local HL density.
   Tests whether the density law continues far above small N.

5. COUNTEREXAMPLE / TERMINATION SEARCH: After the largest observed
   twin prime, we actively search a large forward window to verify
   more exist. This directly falsifies any claim of a "last" twin.

6. LOG-LOG REGRESSION: Fit log(pi_2(N)) vs log(N) and log(ln N).
   HL predicts slope 1 in N, slope -2 in ln N. Test this fit.

A passing result on all fronts constitutes strong empirical support;
it does not constitute a proof.

[1] Sieving primes up to N_MAX = 10,000,000 ...
    Found 664,579 primes up to 10,000,000
    Found 58,980 twin-prime pairs with p <= 10,000,000
    First 10 twin pairs: [(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41,
         43), (59, 61), (71, 73), (101, 103), (107, 109)]
    Last  5 twin pairs: [(9998969, 9998971), (9999047, 9999049), (9999161,
        9999163), (9999929, 9999931), (9999971, 9999973)]

[2] -HardyLittlewood asymptotic pi_2(N) ~ 2*C2*N/(ln N)^2
            N    pi_2(N) obs        HL pred      ratio
        1,000             35           27.7     1.2649
       10,000            205          155.6     1.3171
      100,000          1,224          996.1     1.2288
    1,000,000          8,169        6,917.5     1.1809
   10,000,000         58,980       50,822.1     1.1605

[3] Gaps between consecutive twin-prime pairs
    Total gaps: 58,979
    Max gap   : 1,722
    Mean gap  : 169.55
    Median gap: 126

[4] Random-window sampling at large scales (via sympy isprime)
    M~1e+08  window=2000  mean twins/window=7.900  HL expected=7.782
```

```
    M~1e+09  window=2000  mean twins/window=5.950  HL expected=6.149
    M~1e+10  window=2000  mean twins/window=4.950  HL expected=4.981
    M~1e+11  window=2000  mean twins/window=3.950  HL expected=4.116
    M~1e+12  window=2000  mean twins/window=4.100  HL expected=3.459

[5] Counterexample search: do twin primes persist beyond observed max?
    Largest enumerated twin-prime lower: p = 9,999,971
    Scanned 618 further candidates, found 5 more twin pairs:
       (10,000,139, 10,000,141)  isprime check = True, True
       (10,000,451, 10,000,453)  isprime check = True, True
       (10,000,721, 10,000,723)  isprime check = True, True
       (10,000,871, 10,000,873)  isprime check = True, True
       (10,001,207, 10,001,209)  isprime check = True, True

    Searching for a twin prime near 10^15 ...
     Found twin prime near 10^15: (1,000,000,950,034,577,
        1,000,000,950,034,579)
      Verification: isprime(1,000,000,950,034,577) = True, isprime
         (1,000,000,950,034,579) = True

[6] Regression: log pi_2(N) vs log N  (HL slope = 1 in N, -2 in ln N)
    Fit: log pi_2(N) = 0.592 + 0.978 * log N + -1.936 * log log N
    HL predi
... [truncated]
```

4 Experiment Code (Basic)

```
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import numpy as np
from sympy import sieve, isprime, primerange
import math
import random
from scipy import stats

print("=" * 70)
print("=== EXPERIMENT PLAN ===")
print("=" * 70)
print("""
Conjecture: There are infinitely many pairs of primes (p, p+2) - the
Twin Prime Conjecture.

Since 'infinitely many' cannot be verified by finite computation, we
gather empirical evidence via the following tests:

1. DIRECT ENUMERATION: Count twin primes up to increasing bounds N
   (10^3, 10^4, ..., up to ~10^7). If twin primes keep appearing and
   never stop, that's supportive evidence.

2. HARDY-LITTLEWOOD CONJECTURE: The strong form predicts
```

```
      pi_2(N) ~ 2 * C_2 * N / (ln N)^2
   where C_2  0.6601618 is the twin prime constant.
   We compare observed counts to this asymptotic and compute the ratio.
   If the ratio stays near 1, it's strong evidence the density does
   not fall to zero, i.e. infinitely many twins.

3. GAP BETWEEN CONSECUTIVE TWIN PRIMES: We track the largest observed
   gap vs N. If twin primes were finite, there would eventually be a
   last one and gaps would blow up. We look for steady growth
   consistent with Cramér-like heuristics rather than termination.

4. RANDOM SAMPLING IN LARGE INTERVALS: In windows [M, M+L] for very
   large M, we count twin primes and compare to the local HL density.
   Tests whether the density law continues far above small N.

5. COUNTEREXAMPLE / TERMINATION SEARCH: After the largest observed
   twin prime, we actively search a large forward window to verify
   more exist. This directly falsifies any claim of a "last" twin.

6. LOG-LOG REGRESSION: Fit log(pi_2(N)) vs log(N) and log(ln N).
   HL predicts slope 1 in N, slope -2 in ln N. Test this fit.

A passing result on all fronts constitutes strong empirical support;
it does not constitute a proof.
""")

# ---------------------------------------------------------------------
# 1. DIRECT ENUMERATION - build primes up to N_max via sieve
# ---------------------------------------------------------------------
N_MAX = 10_000_000
print(f"[1] Sieving primes up to N_MAX = {N_MAX:,} ...")

def sieve_eratosthenes(n):
    sieve = np.ones(n + 1, dtype=bool)
    sieve[:2] = False
    for i in range(2, int(math.isqrt(n)) + 1):
        if sieve[i]:
            sieve[i*i::i] = False
    return np.nonzero(sieve)[0]

primes = sieve_eratosthenes(N_MAX)
print(f"    Found {len(primes):,} primes up to {N_MAX:,}")

# Twin primes: p such that p and p+2 are both prime
prime_set = np.zeros(N_MAX + 3, dtype=bool)
prime_set[primes] = True
twin_mask = prime_set[:-2] & prime_set[2:]  # index i => i and i+2 both prime
twin_lowers = np.nonzero(twin_mask)[0]
twin_lowers = twin_lowers[twin_lowers >= 3]  # standard: (3,5) is first
print(f"    Found {len(twin_lowers):,} twin-prime pairs with p <= {N_MAX:,}")
print(f"    First 10 twin pairs: {[(int(p), int(p)+2) for p in twin_lowers[:10]]}")
```

```python
print(f"    Last  5 twin pairs: {[(int(p), int(p)+2) for p in twin_lowers[-5:]]}")

# ---------------------------------------------------------------------
# 2. HARDY-LITTLEWOOD asymptotic check
# ---------------------------------------------------------------------
print("\n[2] -HardyLittlewood asymptotic pi_2(N) ~ 2*C2*N/(ln N)^2")
C2 = 0.6601618158468695739278121100145
bounds = np.array([10**k for k in range(3, 8)])  # 1e3 ... 1e7
observed = np.array([int(np.sum(twin_lowers <= b)) for b in bounds])
predicted = 2 * C2 * bounds / (np.log(bounds) ** 2)
ratio = observed / predicted

print(f"    {'N':>10} {'pi_2(N) obs':>14} {'HL pred':>14} {'ratio':>10}")
for b, o, p, r in zip(bounds, observed, predicted, ratio):
    print(f"    {b:>10,} {o:>14,} {p:>14,.1f} {r:>10.4f}")

# ---------------------------------------------------------------------
# 3. GAP BETWEEN CONSECUTIVE TWIN PRIMES
# ---------------------------------------------------------------------
print("\n[3] Gaps between consecutive twin-prime pairs")
twin_gaps = np.diff(twin_lowers)
print(f"    Total gaps: {len(twin_gaps):,}")
print(f"    Max gap   : {int(twin_gaps.max()):,}")
print(f"    Mean gap  : {twin_gaps.mean():.2f}")
print(f"    Median gap: {int(np.median(twin_gaps)):,}")

# Record running maximum gap vs N for the plot
running_max_gap = np.maximum.accumulate(twin_gaps)

# ---------------------------------------------------------------------
# 4. RANDOM WINDOW SAMPLING AT LARGE M
# ---------------------------------------------------------------------
print("\n[4] Random-window sampling at large scales (via sympy isprime)")
random.seed(42)
window_results = []
# Pick several large starting points up to 10^12; count twin primes in windows.
window_size = 2000
scales = [10**8, 10**9, 10**10, 10**11, 10**12]
for M in scales:
    # 20 windows per scale
    cnt = 0
    trials = 20
    for _ in range(trials):
        start = random.randint(M, 2 * M)
        local = 0
        # scan start .. start + window_size for twin primes
        # use mil
# ... [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.