# Empirical Investigation of an Open Conjecture: Every even integer strictly greater than 2 can be expressed as the sum of two pr

Agentic NL→Lean 4 Pipeline
Job #3

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.**

*Every even integer strictly greater than 2 can be expressed as the sum of two prime numbers.*

## 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 (Goldbach, strong form):
    Every even integer n > 2 can be written as n = p + q where p, q are
       primes.

This script gathers empirical evidence via multiple complementary tests:

 (1) EXHAUSTIVE CHECK on a dense range [4, N_small]:
```

```
    For every even n in [4, N_small], verify there exist primes p,q with p+
       q=n.
    Count the number of Goldbach representations g(n).
    This is the strongest direct test: any failure is a counterexample.

 (2) LARGE-SCALE RANDOM SAMPLING on [4, N_huge]:
    Sample thousands of random even integers up to ~10^8 and verify
       Goldbach
    holds for each. Because g(n) grows roughly like n/(ln n)^2, tests at
    very large n are independent corroboration.

 (3) EDGE CASES & BOUNDARIES:
    Test n = 4 (the smallest case, 2+2), small even n, powers of 2,
    primorial-related even numbers, and the vicinity of famously-checked
    record ranges.

 (4) ASYMPTOTIC BEHAVIOR -- HardyLittlewood heuristic:
    Compare the observed number of representations g(n) against the-
    HardyLittlewood prediction
        g(n) ~ 2 * C_2 * n/(ln n)^2 * prod_{p|n, p>2}(p-1)/(p-2)
    where C_2   0.6601618158... is the twin prime constant.
    A good fit across orders of magnitude is powerful indirect evidence
    that Goldbach cannot fail asymptotically.

 (5) COUNTEREXAMPLE SEARCH:
    Explicitly attempt to break the conjecture by examining "hard" n
    (n with few small prime factors, large n sampled near 10^8, etc.).
    Report the minimum representation count min_g(n) observed - if it
    ever hits 0, that is a counterexample.

 (6) STATISTICAL SANITY CHECK:
    Pearson correlation and log-log regression between observed g(n)
    and -HardyLittlewood prediction to verify the predicted growth law.

[1] Exhaustive check on even n in [4, 200000] ...
    Even integers tested: 99999
    Failures (g(n)=0):   0
    min g(n) over range:  1 (attained at 2*argmin = 4)
    max g(n) over range:  3931

[2] Random sampling: 12000 even n in [4, 100000000] ...
    Trials: 12000, range [4, 100000000]
    Failures: 0
    Mean smallest witness prime p: 30.04, max: 601

[3] Edge cases & boundary conditions ...
    n =            4: OK  witness = (2, 2)
    n =            6: OK  witness = (3, 3)
    n =            8: OK  witness = (3, 5)
    n =           10: OK  witness = (3, 7)
    n =           12: OK  witness = (5, 7)
    n =           14: OK  witness = (3, 11)
    n =           16: OK  witness = (3, 13)
    n =           18: OK  witness = (5, 13)
```

```
    n =           20: OK  witness = (3, 17)
    n =          100: OK  witness = (3, 97)
    n =         1000: OK  witness = (3, 997)
    n =        10000: OK  witness = (59, 9941)
    n =       100000: OK  witness = (11, 99989)
    n =      1048576: OK  witness = (3, 1048573)
    n =     33554432: OK  witness = (61, 33554371)
    n =     67108866: OK  witness = (7, 67108859)
    n =     10000000: OK  witness = (29, 9999971)
    n =     10000002: OK  witness = (11, 9999991)
    n =    100000000: OK  witness = (11, 99999989)
    n =     99999998: OK  witness = (67, 99999931)
    n =     99999998: OK  witness = (67, 99999931)
    n =     99999994: OK  witness = (5, 99999989)
    (hard) n =        60060: OK  witness = (19, 60041)
    (hard) n =      1021020: OK  witness = (19, 1021001)
    (hard) n =     67108862: OK  witness = (3, 67108859)
    (hard) n =     16777220: OK  witness = (7, 16777213)

[4] Asymptotic: comparing g(n) with -HardyLittlewood prediction ...
    Samples: 209
    Pearson r (log g vs log HL):   0.99834
    Log-log slope (expect ~1):     0.97878
    Mean observed/predicted ratio: 0.6378  (std 0.0709)

[5] Targeted counterexample search ...
    Minimum g(n) in [4,200000] = 1, attained at [4, 6, 8, 12] (count=4)
    Adversarial large-n trials: 3000, failures: 0

[6] Producing plots ...

=== SUMMARY ===
Exhaustive range:   all even 4..200000   -> 0 failures
Random sampling:    12000 even n up to 100000000   -> 0 failures
Ed
... [truncated]
```

# 4 Experiment Code (Basic)

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

print("===␣EXPERIMENT␣PLAN␣===")
print("""
Conjecture (Goldbach, strong form):
```

```
    Every even integer n > 2 can be written as n = p + q where p, q are
       primes.

This script gathers empirical evidence via multiple complementary tests:

 (1) EXHAUSTIVE CHECK on a dense range [4, N_small]:
     For every even n in [4, N_small], verify there exist primes p,q with p+
        q=n.
     Count the number of Goldbach representations g(n).
     This is the strongest direct test: any failure is a counterexample.

 (2) LARGE-SCALE RANDOM SAMPLING on [4, N_huge]:
     Sample thousands of random even integers up to ~10^8 and verify
        Goldbach
     holds for each. Because g(n) grows roughly like n/(ln n)^2, tests at
     very large n are independent corroboration.

 (3) EDGE CASES & BOUNDARIES:
     Test n = 4 (the smallest case, 2+2), small even n, powers of 2,
     primorial-related even numbers, and the vicinity of famously-checked
     record ranges.

 (4) ASYMPTOTIC BEHAVIOR -- HardyLittlewood heuristic:
     Compare the observed number of representations g(n) against the-
     HardyLittlewood prediction
         g(n) ~ 2 * C_2 * n/(ln n)^2 * prod_{p|n, p>2}(p-1)/(p-2)
     where C_2  0.6601618158... is the twin prime constant.
     A good fit across orders of magnitude is powerful indirect evidence
     that Goldbach cannot fail asymptotically.

 (5) COUNTEREXAMPLE SEARCH:
     Explicitly attempt to break the conjecture by examining "hard" n
     (n with few small prime factors, large n sampled near 10^8, etc.).
     Report the minimum representation count min_g(n) observed - if it
     ever hits 0, that is a counterexample.

 (6) STATISTICAL SANITY CHECK:
     Pearson correlation and log-log regression between observed g(n)
     and -HardyLittlewood prediction to verify the predicted growth law.
""")

t0 = time.time()

#
   ----------------------------------------------------------------------------

# Sieve of Eratosthenes (vectorized) up to N
#
   ----------------------------------------------------------------------------

def sieve(N):
    s = np.ones(N+1, dtype=bool)
    s[:2] = False
    for p in range(2, int(math.isqrt(N))+1):
```

```
        if s[p]:
            s[p*p::p] = False
    return s

#
    ------------------------------------------------------------------------

# (1) Exhaustive check on [4, N_small]; also compute g(n)
#
    ------------------------------------------------------------------------

N_small = 200_000
print(f"[1] Exhaustive check on even n in [4, {N_small}] ...")
is_prime = sieve(N_small)
primes = np.nonzero(is_prime)[0]
prime_set = set(int(p) for p in primes)

failures = []
g_values = np.zeros(N_small//2 + 1, dtype=np.int64)  # g_values[k] = g(2k)

# For each even n, count primes p <= n/2 with (n-p) also prime.
# Vectorize by iterating over primes p and adding to n = p + q for q prime,
    q >= p.
for p in primes:
    p = int(p)
    # q ranges over primes >= p with p+q <= N_small
    # n = p+q, n even iff p,q same parity. p=2 => q even prime => only q=2 (
        n=4).
    if p == 2:
        # only contributes n=4
        if 4 <= N_small:
            g_values[2] += 1  # (2,2)
        continue
    # p odd: q odd primes >= p, p+q <= N_small
    qmax = N_small - p
    if qmax < p:
        continue
    # odd primes in [p, qmax]
    mask_q = is_prime[p:qmax+1].copy()
    # only odd indices matter (but primes>2 are odd anyway)
    qs_idx = np.nonzero(mask_q)[0] + p
    sums = qs_idx + p  # even integers
    # If p == q, unordered pair counted once; if p < q, also once.
    # Using q >= p ensures unordered counting.
    np.add.at(g_values, sums//2, 1)

# Verify every even n in [4, N_small] has g(n) >= 1
for k in range(2, N_small//2 + 1):
    n = 2*k
    if g_values[k] == 0:
        failures.append(n)

print(f"    Even integers tested: {N_small//2 - 1}")
print(f"    Failures (g(n)=0):    {len(failures)}")
```

```
if failures:
    print(f"␣␣␣␣First␣few␣failures:␣␣{failures[:10]}")
print(f"␣␣␣␣min␣g(n)␣over␣range:␣␣{int(g_values[2:].min())}␣(attained␣at␣"
      f"2*argmin␣=␣{2*int(np.argmin(g_values[2:])␣+␣2)})")
print(f"␣␣␣␣max␣g(n)␣over␣range:␣␣{int(g_values[2:].max())}")

#
    ----------------------------------------------------------------------------

# (2) Large-scale random sampling up to N_huge
#
    ----------------------------------------------------------------------------

N_huge = 10**8
n_random_trials = 12_000
print(f"\n[2]␣Random␣sampling:␣{n_random_trials}␣even␣n␣in␣[4,␣{N_huge}]␣...
    ")

# Precompute primes up to sqrt(N_huge) for Miller-Rabin fallback? We'll use
# a deterministic Miller-Rabin for 64-bit integers.

def is_prime_mr(n):
    if n < 2:
        return False
    small = [2,3,5,7,11,13,17,19,23,29,31,37]
    for p in small:
        if n == p:
            return True
        if n % p == 0:
            return False
    d = n-1
    r = 0
    while d % 2 == 0:
        d //= 2

# ... [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.