ON A CONJECTURE OF ESSER, TOTARO, AND WANG

JIHAO LIU

Abstract. We disprove a conjecture of Esser and Totaro on relative primeness of two explicitly constructible integers. Consequently, the large-index conjecture of Esser, Totaro, and Wang fails in dimension 159. The main result of this paper is obtained by generative AI, particularly Chatgpt 5.5 pro, and re-verified by the Rethlas system.

1. Introduction

Esser, Totaro, and Wang constructed klt Calabi–Yau varieties with large indices in arbitrary dimensions [ETW22]. More precisely, [ETW22, Section 7] constructs a special class of klt Calabi–Yau varieties with conjecturally large indices, formulated in [ETW22, Conjecture 7.10]. Later, Esser and Totaro transformed [ETW22, Conjecture 7.10] into the following number-theoretic conjecture [ET24, Proposition 7.3].

Conjecture 1.1 ([ET24, Conjecture 7.4])**.** *For every integer $n \geq 2$, $\gcd(m'_n, E_n) = 1$.*

Here m'_n and E_n are two explicitly computable numbers defined as in [ET24, Sections 4 and 7]. See Construction A.1 below for an explicit definition of m'_n and E_n when n is odd. Conjecture 1.1 was verified for $n \leq 30$ [ET24, Section 7]. The following computation gives a counterexample to Conjecture 1.1 and hence to the expected large-index construction in dimension 159.

Theorem 1.2. *Conjecture 1.1 fails when $n = 159$ as*

$$53 \mid E_{159} \quad \text{and} \quad 53 \mid m'_{159}. \tag{1.1}$$

As a consequence, [ETW22, Conjecture 7.10] *fails in dimension* 159.

Proof. This follows from the computation in Proposition B.1, together with [ET24, Proposition 7.3]. □

Remark 1.3. The main result of this paper is obtained by generative AI by asking 2 questions to Chatgpt 5.5 pro without any mathematical inputs. Chatgpt 5.5 pro provided the right counterexample and the Python code. The Rethlas system then verified that the proof and the Python code are both correct. See [Ju+26] for a detailed introduction to the Rethlas system.

Due to the limitation of generative AI, it is possible that we have missed some related references in the literature, and we welcome any comments from experts.

Acknowledgements. The author was partially supported by the National Key R&D Program of China #2024YFA1014400. The author would like to thank the Rethlas team, namely Haocheng Ju, Jiedong Jiang, Shurui Liu, Guoxiong Gao, Yuefeng Wang, Zeming Sun, Bin Wu, Liang Xiao, and Bin Dong, for their contributions to the development of Rethlas and its customized version used for the problem studied in this paper. The author would like to thank Kaiyuan Gu, Ruicheng Hu, and Sheng Qin for assistance with the verification of an earlier blueprint of this paper. The author would like to thank Ruochuan Liu and Gang Tian for constant support and encouragement.

Date: May 21, 2026.

2020 *Mathematics Subject Classification.* 14E30, 14J32.

Key words and phrases. Calabi-Yau varieties, volumes.

Appendix A. The numbers

We construct the numbers used in Conjecture 1.1.

Construction A.1. In this paper, $s_0, s_1, s_2, \dots$ denote Sylvester's sequence

$$s_0 = 2, \qquad s_{j+1} = s_j(s_j - 1) + 1. \tag{A.1}$$

In the following, we fix a positive integer r. We define:

- $n := 2r + 1$.
- $b_i := s_i$ for $0 \le i \le r$.
- b_{r+i} and B_I are defined recursively in the following way:
  - For a finite tuple of indices $I = (i_1, i_2, \dots, i_k)$, we define the alternating product

$$B_I = b_{i_1} b_{i_2} \cdots b_{i_k} - b_{i_1} \cdots b_{i_{k-1}} + \cdots + (-1)^{k-1} b_{i_1} + (-1)^k, \tag{A.2}$$

    with the convention $B_\emptyset = 1$.
  - For any $1 \le i \le r + 1$, we define

$$b_{r+i} = 1 + (b_{r+1-i} - 1)^2 B_{r+1,r,r+2,r-1,\dots,r-1+i,r+2-i}. \tag{A.3}$$

- $t_1 = B_{r+1,r,\dots,2r,1}$.
- $g_1 = B_{1,2r,\dots,r,r+1}$.
- We define

$$b'_{2r+1} = \frac{1 + b_1 b_2 \cdots b_{2r} + (s_1 - 1)t_1}{2}, \tag{A.4}$$

- We define

$$E_{2r+1} = \frac{b_1 b_2 \cdots b_{2r} + 1}{2}, \tag{A.5}$$

- We define

$$m'_{2r+1} = b_0 b'_{2r+1} g_1 - b_0 + 1. \tag{A.6}$$

Appendix B. Code

Below is the Python code used in the proof of Theorem 1.2.

Proposition B.1. *The Python program below computes*

$$(P, G = g_1, T = t_1, b'_{159}, E_{159}, m'_{159}, b_{159}) \equiv (52, 34, 46, 34, 0, 0, 35) \pmod{53}.$$

where $P = \prod_{i=1}^{158} b_i$.

```
p = 53
r = 79

def B(I, b):
    """Alternating␣product␣B_{i_1...i_k}␣via␣prefix␣products."""
    m = len(I)
    pref = [1]
    for i in I:
        pref.append((pref[-1] * b[i]) % p)
    return sum((1 if j % 2 == 0 else -1) * pref[m - j]
               for j in range(m + 1)) % p

def B_recursive(I, b):
    """Recursion␣check:␣B_{i_1␣i_2␣...␣i_k}␣=␣b_{i_1}␣B_{i_2␣...␣i_k}␣+␣(-1)^k.
    ↪ """
    if len(I) == 0:
```

```
        return 1 % p
    sign = 1 if len(I) % 2 == 0 else -1
    return (b[I[0]] * B_recursive(I[1:], b) + sign) % p

s = [0] * (r + 2)
s[0] = 2
for i in range(r + 1):
    s[i + 1] = (s[i] * (s[i] - 1) + 1) % p

b = [None] * (2 * r + 2)
for i in range(r + 1):
    b[i] = s[i]

for i in range(1, r + 2):
    I = []
    for q in range(i - 1):
        I.extend([r + 1 + q, r - q])
    b1 = (1 + (b[r + 1 - i] - 1) ** 2 * B(I, b)) % p
    b2 = (1 + (b[r + 1 - i] - 1) ** 2 * B_recursive(I, b)) % p
    assert b1 == b2
    b[r + i] = b1

P = 1
for j in range(1, 2 * r + 1):
    P = (P * b[j]) % p

I_T = []
for q in range(r):
    I_T.extend([r + 1 + q, r - q])
T = B(I_T, b)
assert T == B_recursive(I_T, b)

I_G = []
for q in range(r):
    I_G.extend([1 + q, 2 * r - q])
G = B(I_G, b)
assert G == B_recursive(I_G, b)

bprime = ((1 + P + (s[1] - 1) * T) * pow(2, -1, p)) % p
E = ((P + 1) * pow(2, -1, p)) % p
mprime = (b[0] * bprime * G - b[0] + 1) % p

assert (P, T, G, bprime, E, mprime, b[2*r + 1]) == (52, 34, 46, 34, 0, 0, 35)

print("p␣=", p, "r␣=", r, "n␣=", 2*r + 1)
print("P␣=", P, "T␣=", T, "G␣=", G)
print("bprime_159␣=", bprime, "␣E_159␣=", E, "␣mprime_159␣=", mprime)
print("b_159␣=", b[2*r + 1])
```

References

[ET24] L. Esser and B. Totaro, *Log canonical pairs with conjecturally minimal volume*, Manuscripta Math. **175** (2024), 865–895.

[ETW22] L. Esser, B. Totaro, and C. Wang, *Calabi-Yau varieties of large index*, Algebr. Geom., to appear, arXiv:2209.04597.

[Ju+26] H. Ju, G. Gao, J. Jiang, B. Wu, Z. Sun, L. Chen, Y. Wang, Y. Wang, Z. Wang, W. He, P. Wu, L. Xiao, R. Liu, B. Dai, and B. Dong, *Automated Conjecture Resolution with Formal Verification*, arXiv:2604.03789.

Department of Mathematics, Peking University, No. 5 Yiheyuan Road, Haidian District, Beijing 100871, China

Beijing International Center for Mathematical Research, Peking University, No. 5 Yiheyuan Road, Haidian District, Beijing 100871, China
Email address: liujihao@math.pku.edu.cn