TABULATING ABSOLUTE LUCAS PSEUDOPRIMES

Chloe Helmreich
Department of Mathematical Sciences, Butler University, Indianapolis, Indiana

Jonathan Webster
Department of Mathematical Sciences, Butler University, Indianapolis, Indiana
jewebste@butler.edu

Received: 10/24/23, Revised: 3/27/24, Accepted: 5/2/24, Published: 5/27/24

Abstract

In 1977, Hugh Williams studied numbers that were Lucas pseudoprimes to all Lucas sequences of a fixed discriminant. These are composite numbers analogous to Carmichael numbers, and they satisfy a Korselt-like criterion: n must be a product of distinct primes p_i and $(p_i - \delta_{p_i}) \mid (n - \delta_n)$ where δ_n is a Legendre symbol with the first argument being the discriminant of the Lucas sequence. Motivated by tabulation algorithms for Carmichael numbers, we give algorithms to tabulate these numbers and provide some asymptotic analysis of the algorithms. We show that there are only finitely many absolute Lucas pseudoprimes $n = \prod_{i=1}^{k} p_i$ with a given set of $k-2$ prime factors for $k > 2$. We also provide the first known tabulations up to 2^{64} for discriminants $5, -7, -11$, and 13.

1. Introduction

A *base a Fermat pseudoprime* is a composite integer n such that

$$a^{n-1} - 1 \equiv 0 \pmod{n}.$$

It is well-known that Carmichael numbers are the composite integers for which that congruence holds for all a such that $(a, n) = 1$. Korselt showed that such a number n is a product of $k > 2$ distinct primes $p_1, p_2, \ldots, p_k$ and $(p_i - 1) \mid (n - 1)$. The least example is $561 = 3 \cdot 11 \cdot 17$. From a computational view, Fermat's Little Theorem was a step into primality testing and Carmichael numbers are a roadblock to this being a successful test. One notable way to strengthen this test is by combining it with a seemingly conflicting test based on Lucas sequences. An example of this would be the Baillie-PSW test [2, 1], which is what the GNU Multiple Precision library [9] currently implements. Another example would be Grantham's Frobenius

DOI: 10.5281/zenodo.11352771

pseudoprimes [10]. The pseudoprimes to the Lucas sequences are our motivating interest.

Since Carmichael numbers inform us about the reliability of the Fermat test, it would make sense to examine the analogous numbers for Lucas sequences. These numbers are, perhaps, less well-known. H.C. Williams showed that these numbers also satisfy a Korselt-like criterion [21]. Using this result as a starting point, we continue a study of these numbers from an algorithmic point of view with an aim of tabulating them. The strategy is to consider numbers of the form $n = Pqr$ for P a squarefree, odd number with q and r prime. Our key contributions are as follows:

1. We prove theorems establishing finiteness and boundedness conditions. For Carmichael numbers, these theorems were initially proved by Beeger for a prime P and generalized by Duparc for P being composite [4, 5].

2. We provide an algorithmic interpretation of these theorems in the spirit of [15, 18]. In particular, the bounds on two primes are $O(P^2)$ and $O(P^3)$ but we can find both primes after creating only an average of $O(P(\log P)^2)$ candidates.

3. We implemented the algorithms in C++ and produced four tabulations for $n < 2^{64}$ using the discriminants $5, -7, -11$, and 13.

These discriminants were chosen to match the choice of discriminants for the Lucas sequences used in the Baillie-PSW test[1].

The rest of the paper is organized as follows. Section 2 gives the background on Lucas sequences, defines what absolute Lucas pseudoprimes are, and states the Korselt-like criterion. Section 3 is a comment on how we will account for asymptotic cost. Section 4 establishes the new theorems providing bounds that may be used for algorithmic purposes. Sections 5 and 6 state algorithms for tabulating these numbers and provide some asymptotic analysis; these two sections are bifurcated by a "small" input size vs a "large" input size. Finally, Section 7 addresses the practical issues with the implementation, provides some statistics on the tabulation, and concludes by pointing out a curious non-uniformity of distribution of the numbers with respect to the Jacobi symbols.

2. Lucas Sequences

There are many equivalent definitions of the Lucas U-sequence. We state two of them and encourage the reader to consult standard sources (such as [14, 20]) for a

[1]Technically, there are at least 8 different ways to choose the specific parameters for the Lucas sequence. Method A, $A^\star$, B, $B^\star$ all start with 5 and uses successive odd discriminants fixing the sign so that $d \equiv 1 \pmod 4$

more robust account. First, they may be defined by expressions involving roots of a certain polynomial:

$$U_n = U_n(A, B) = (\alpha^n - \beta^n)/(\alpha - \beta),$$

where α, β are the zeros of $x^2 - Ax + B$, and A, B are relatively prime integers with $A > 0$. Let the discriminant be $d = A^2 - 4B$. Alternatively, we may define these sequences with a recurrence relation:

$$U_0(A, B) = 0, U_1(A, B) = 1, \text{ and } U_n(A, B) = AU_{n-1}(A, B) - BU_{n-2}(A, B).$$

This latter definition is used to derive identities that allow efficient computation of $U_n(A, B) \pmod m$ for large n with an algorithm akin to square-and-multiply [13]. We will frequently suppress the A, B notation since our work concerns all Lucas sequences with a fixed discriminant. There is a well-known analog of Fermat's little theorem for Lucas sequences with A and B fixed.

Theorem 1 (Analog of Fermat's Little Theorem)**.** *If p is an odd prime and $p \nmid dB$, then*

$$U_{p-\delta_p}(A, B) \equiv 0 \pmod p,$$

where δ_p is the Jacobi symbol $\left(\frac{d}{p}\right)$.

In everything that follows, we use δ_p for the Jacobi symbol $\left(\frac{d}{p}\right)$. Since the first argument is always d, which is understood from context, we choose a notation that suppresses this.

As with Fermat's Little theorem, the contrapositive of this theorem can be used to detect if an integer is composite. And, one can find composite numbers which the contrapositive of the above theorem does not detect, which motivates the following definition.

Definition 1. An (A, B)-*Lucas pseudoprime* is a composite integer n satisfying

$$U_{n-\delta_n}(A, B) \equiv 0 \pmod n.$$

For example, the Fibonacci pseudoprimes (sequence A081264 in the OEIS) are $(1, -1)$-Lucas pseudoprimes. The first 15 are: 323, 377, 1891, 3827, 4181, 5777, 6601, 6721, 8149, 10877, 11663, 13201, 13981, 15251, and 17119.

Definition 2. An *absolute Lucas pseudoprime (to the discriminant d)* is a composite integer n satisfying

$$U_{n-\delta_n}(A, B) \equiv 0 \pmod n$$

for all Lucas sequences with discriminant d and $(n, dB) = 1$.

From the above 15 Fibonacci pseudoprimes, the numbers $323, 6601, 6721, 11663$, and 17119 are absolute Lucas pseudoprimes. This can be checked with a Korselt-like criterion.

Theorem 2 (Williams' Criterion [21])**.** *A composite number n is an absolute Lucas pseudoprime if and only if n is squarefree and $(p-\delta_p) \mid (n-\delta_n)$ for all prime divisors p of n.*

If $d = 1$, the absolute Lucas pseudoprimes are Carmichael numbers and the divisibility statement in Theorem 2 becomes $(p-1)|(n-1)$. In the algorithms for tabulating Carmichael numbers, it was common to need the Carmichael function $\lambda(n)$. We will need a similar function but only state what values it takes for squarefree numbers, which is our only concern.

Definition 3. For a product $n = \prod_{i=1}^{j} p_i$ of distinct primes, define

$$\lambda_d(n) = \operatorname{lcm}(p_1 - \delta_{p_1}, \ldots, p_j - \delta_{p_j}).$$

If $d = 1$, $\lambda_1(n)$ is the Carmichael function. In the same way that $\lambda_1(n)$ is the least universal exponent, $\lambda_d(n)$ is the least universal "rank of apparition" for Lucas sequences with discriminant d. While the asymptotic behavior of $\lambda_1(n)$ has been well-studied (e.g., [7, 8]), we know of no asymptotic results on $\lambda_d(n)$ for $d \neq 1$.

3. Boundedness Theorems

As with the tabulation algorithms for Carmichael numbers, we will explicitly construct the prime factorization of the number. For numbers with exactly k prime factors, we will start with a *preproduct* that has either $k-1$ (the "large" case) or $k-2$ (the "small" case) prime factors. Then we will use the below theorems to find the remaining one or two prime factors which we will usually call q and r. These theorems will limit both the number and size of primes that may complete the preproduct. We show that criterion in Theorem 2 may be used to get analogous boundedness and finiteness results.

First, we establish the analog of Proposition 1 of [15].

Theorem 3. *Let $n = \prod_{i=1}^{k} p_i$ be an absolute Lucas pseudoprime less than B with $k > 2$ prime factors listed in increasing order by subscript, and $P_j = \prod_{i=1}^{j} p_i$ for some $j < k$. Then the following statements hold:*

(i) $p_{j+1} < (B/P_j)^{1/(k-j)}$ *and* $p_{j+1} - \delta_{p_{j+1}}$ *is relatively prime to* p_i *for all* $i \leq j$*;*

(ii) $P_{k-1}p_k \equiv \delta_{P_{k-1}}\delta_{p_k} \pmod{\lambda_d(P_{k-1})}$ *and* $p_k - \delta_{p_k}$ *divides* $P_{k-1} - \delta_{P_{k-1}}$*;*

(iii) *each* p_i *satisfies* $p_i \leq P_{k-1} + 2 < \sqrt{2n} < \sqrt{2B}$.

Proof. Parts (i) and (ii) follow from the fact that $(p_i - \delta_{p_i})$ divides $(n - \delta_n)$ for each i. For (iii), $n = P_{k-1}p_k$ and $(p_i - \delta_{p_i}) \mid (n - \delta_n)$ imply $p_k \leq P_{k-1} + 2 < 2P_{k-1}$. Now, $p_k^2 < 2P_{k-1}p_k = 2n < 2B$. □

Theorem 3 requires $k > 2$; we will address the case of $k = 2$ below. The requirement that $p_{j+1} - \delta_{p_{j+1}}$ is relatively prime to p_i for all $i \leq j$ is stronger than the square-free requirement in Theorem 2. We call a square-free composite number P_j *admissible* if all of its prime divisors satisfy the relatively prime criterion of Theorem 3(i). Further, we say P_j is *bounds admissible* (with respect to B) if it also satisfies the inequality in Theorem 3(i).

When $d = 1$, the admissible numbers are also called *cyclic* (in the group theory sense) numbers. In [6], Erdős proved that the counting function of cyclic numbers is asymptotic to

$$\frac{e^{-\gamma}B}{\log\log\log B},$$

where $\gamma \approx 0.5772\ldots$ is the Euler-Mascheroni constant. We believe that his proof holds for $d \neq 1$ due to a formal replacement of various "1's" in the proof to some appropriate Jacobi symbol. This factor of $\log\log\log B$ plays no role in the analysis that follows, so we do not attempt to prove this result.

Second, we establish the analog of Proposition 2 of [15].

Theorem 4. *Let n be an absolute Lucas pseudoprime of the form $n = Pqr$ with q and r primes, $q < r$, and $P > 1$. There are integers $1 \leq D < P < C$ such that, putting $\Delta = CD - P^2$, we have*

$$q - \delta_q = \frac{(P - \delta_P)(\delta_q P + \delta_r D)}{\Delta}, \tag{4.1}$$

$$r - \delta_r = \frac{(P - \delta_P)(\delta_r P + \delta_q C)}{\Delta}, \tag{4.2}$$

$$\frac{(p-1)P^2 - 2P}{p+1} < CD < \frac{(p+3)P^2 + 2P}{p+1}, \tag{4.3}$$

where p is the largest prime dividing P.

Proof. Since

$$(q - \delta_q) \mid (Pqr - \delta_P\delta_q\delta_r) = Pqr - Pr\delta_q + Pr\delta_q - \delta_P\delta_q\delta_r$$

it follows that $(q - \delta_q) \mid (Pr - \delta_P\delta_r)$. Similarly, $(r - \delta_r) \mid (Pq - \delta_P\delta_q)$. Hence, we define positive integers

$$D = \frac{Pq - \delta_P\delta_q}{r - \delta_r} \quad \text{and} \quad C = \frac{Pr - \delta_P\delta_r}{q - \delta_q},$$

satisfying $1 \leq D < P < C$. We have

$$C(q - \delta_q) = P\left(\frac{Pq - \delta_P\delta_q}{D} + \delta_r\right) - \delta_P\delta_r$$

so that

$$CD(q - \delta_q) = P^2q - P\delta_P\delta_q + PD\delta_r - D\delta_P\delta_r.$$

Further,

$$\begin{aligned}(CD - P^2)(q - \delta_q) &= P^2\delta_q - P\delta_P\delta_q + PD\delta_r - D\delta_P\delta_r \\ &= (P - \delta_P)(\delta_qP + \delta_rD).\end{aligned}$$

Note that $\Delta = CD - P^2 \neq 0$, so that

$$q - \delta_q = \frac{(P - \delta_P)(\delta_qP + \delta_rD)}{\Delta}.$$

and similarly

$$r - \delta_r = \frac{(P - \delta_P)(\delta_rP + \delta_qC)}{\Delta}.$$

Note that $p + 1 \leq q - \delta_q$ so

$$p + 1 \leq q - \delta_q = \frac{(P - \delta_P)(\delta_qP + \delta_rD)}{\Delta}.$$

So,

$$|CD - P^2| < \frac{(P + 1)(P + D)}{p + 1} < \frac{2P(P + 1)}{p + 1}$$

implies

$$-\frac{2P(P + 1)}{p + 1} + P^2 < CD < \frac{(2P)(P + 1)}{p + 1} + P^2$$

which is equivalent to

$$\frac{(p - 1)P^2 - 2P}{p + 1} < CD < \frac{(p + 3)P^2 + 2P}{p + 1}.$$

□

Corollary 1. *There are only finitely many absolute Lucas pseudoprimes with $k > 2$ prime factors assuming a set of $k - 2$ of the prime factors are fixed.*

Corollary 2. *With the notation above, $q < 2(P + 1)^2$ and $r < (P + 1)^3$.*

A naive interpretation of the above corollary would imply $O(P^2 \log P)$ arithmetic operations are required to use a sieve of Eratosthenes to find candidate primes q for P. This, in turn, requires $\Omega(P^2 \log P)$ arithmetic operations to find r because there is at least $O(1)$ arithmetic operations required for a given pair P and q. We will see below that we can do much better than this.

4. Model of Computation

It is common to measure the asymptotic cost of an algorithm in either bit operations or arithmetic operations. Informally, asymptotic notation (especially big-O) is often used as a way to give guidance about the run-time of implemented algorithms. Our theorem statements will count the number of candidate pairs created for q and r but our exposition may speak more loosely as if this were measuring time. The theorems state the asymptotic count of arithmetic operations to create q and r without testing if they are prime. We could multiply the asymptotic costs in this model by the asymptotic cost of primality testing to get a result that would be an asymptotic result measuring arithmetic operations. However, this result would not be of much guidance for the run-time of an implementation because primality testing is not often the bottle-neck. It is often the case that q and r may be checked with $O(1)$ arithmetic operations. Here are some examples: they may be too big, they may not be integers, they may not satisfy certain divisibility statements, or they may be found in a look-up table. So, it could be the case that the average cost is $O(1)$ arithmetic operations. Our implementation uses strong Fermat tests with the bases $\{2, 3, 5, 7, 11\}$ and this is sufficient to prove primality for all 32-bit integers [12]. Whenever some factorizations of $n-1$ or $n+1$ are known, there are fast primality tests[2] (see Sections 4.1 and 4.2 of [3] or [20] for more details). As we will see below, it is often the case that we know a complete or partial factorization of $q-\delta_q$ or $r-\delta_r$ and these tests would be helpful. Given the variety of approaches that are available, we believe that it is best to provide asymptotic arguments in terms of the counts of candidates q and r rather than the more traditional bit or arithmetic operations. For empirical evidence supporting this, see Example 1 where about 15.6 million candidate primes are created and the algorithm only invoked a primality test 68 times.

5. Algorithms for Small Preproducts

In [15, 18], Carmichael numbers are constructed of the form $n = Pqr$. We provide a sketch of the main algorithmic ideas of this section. Using Theorem 4, the inequality $1 \leq D < P$ may be used in a for loop. Then, there are two approaches. First and following the approach of [15], we use (4.3) to construct valid C for the inner for loops. With C and D, one can construct q and r and perform the required checks. Second and following the approach of [18], we use the numerator of $q-\delta_q$ in (4.1) to construct all of its possible divisors Δ. These divisors are efficiently obtained via the use of some variant of the sieve of Eratosthenes. With D and Δ, we may construct

[2]It is perhaps fitting for this work that these tests are also inspired by Édouard Lucas and some of the variants bear his name.

C and r and perform the required checks. Before a more thorough explanation and analysis, we deal with the smallest possible preproduct, $P = 1$. This situation is unique to these numbers and cannot arise with Carmichael numbers.

5.1. $P = 1$

A complete tabulation must account for the case that n is a product of exactly two primes, $n = p_1p_2$. In [21], it is proved that this only happens when $p_1 = p_2 - 2$, $\delta_{p_1} = -1$, and $\delta_{p_2} = 1$. Therefore, it suffices to tabulate twin primes in set residue classes[3]. For example, with $d = 5$ we need the primes that are $17, 19 \pmod{30}$. A straightforward implementation of the sieve of Eratosthenes finds these in $O(B^{1/2} \log\log B)$ arithmetic operations. There are other sieving methods that can improve the time by a factor of $(\log\log B)^3$ [19]. This component of the computation contributes only a lower-order term in the overall asymptotic cost of tabulation. Henceforth, we assume that there are always $k > 2$ prime factors in our construction.

5.2. CD Method

The first approach follows Pinch's method of constructing CD pairs. To do so, a double nested for loop creates D satisfying $1 \le D < P$. The inequality (4.3) sets the bounds for C for the second for loop. In the inner loop, we check that the number n is an absolute Lucas pseudoprime. That is, we check that q and r are integral. Second, we check that the divisibility statements in Theorem 2 hold for all primes. Lastly, we check that both q and r are primes. The purpose of ordering of these checks is to delay the most expensive checks until last.

Theorem 5. *The number of CD pairs used to tabulate all absolute Lucas pseudoprimes of the form Pqr is $\Theta(P_{k-3}P\log P) \subset O(P^{2-\frac{1}{k-2}}\log P)$.*

Proof. We start with the inequality in the proof of Theorem 4 that bounds the length of the interval around P^2:

$$|CD - P^2| < \frac{2P(P+1)}{p+1} < 2P_{k-3}(P+1).$$

So, the interval length is bounded by $4P_{k-3}(P+1)$. Now, the total number of C values created for each D is given by

$$\sum_{D=1}^{P-1} \left\lfloor \frac{4P_{k-3}(P+1)}{D} \right\rfloor = \Theta(P_{k-3}P\log P).$$

[3]This example could be seen as the simplest example of a Chernick-like class of absolute Lucas pseudoprimes and for two prime factors that is all there is. It is not hard to create other Chernick-like families, e.g., $n = p_1p_2p_3 = (6k-1)(6k+1)(18k-1)$, where $-\delta_{p_1} = \delta_{p_2} = -\delta_{p_3} = 1$, will be an absolute Lucas pseudoprime.

Since P_{k-3} may be bounded by $P^{1-\frac{1}{k-2}}$ (see Theorem 3(i)), this gives a bound of $O(P^{2-\frac{1}{k-2}} \log P)$. □

Due to the absolute value on the inequality above, double the work is required. For each CD pair, two cases are considered. This implies that this should be about four times slower than the CD method for the Carmichael case. Since this constant is ignored in the asymptotic analysis, the result is the same as Theorem 4 from [18].

5.3. $D\Delta$ Method

The second method is to construct the divisors of $(P-\delta_P)(\delta_q P+\delta_r D)$. The symbol δ_r allows $|(\delta_q P + \delta_r D)|$ to be any integer in $[1, 2P-1]$ (except P). The symbol δ_q allows these divisors to be positive or negative. So, there are a total of four different cases to consider. For each integer in $[1, 2P-1]$, the algorithm considers its negation, too. Thus, we account for all four possible choices of Jacobi symbols. For each of the four separate cases, we constructed C by first checking it is an integer. Next, q and r are created using the symbols from the four choices. Lastly, the divisibility criteria of Theorem 2 is checked before testing whether q and r are primes.

Theorem 6. *The number of $D\Delta$ pairs used to tabulate all absolute Lucas pseudoprimes of the form Pqr is $O(\tau(P-\delta_P)\,(P\log P))$.*

Proof. For every P, we consider all D in the interval $[1, P-1]$. Then count the number of divisors of $(P-\delta_P)(\delta_q P+\delta_r D)$.

$$\begin{aligned}
\sum_{D<P} \tau\left((P-\delta_P)(\delta_q P+\delta_r D)\right) &< \tau(P-\delta_P)\left(\sum_{D<P} \tau(\delta_q P+\delta_r D)\right) \\
&< 2\tau(P-\delta_P)\left(\sum_{n<2P} \tau(n)\right) \\
&= 2\tau(P-\delta_P)\,(2P\log 2P + O(P)) \\
&= O(\tau(P-\delta_P)\,(P\log P))
\end{aligned}$$

The second inequality follows from two facts. Since $(\delta_q P + \delta_r D)$ can be either positive or negative, this accounts for the appearance of the 2 in third line. Since, $(\delta_q P + \delta_r D)$ ranges in values from 1 to $2P-1$, this accounts for the change in the bounds on the summation. □

As with the CD method, this is the same asymptotic result as Theorem 5 from [18] but with an implied constant that is 4 times larger.

Example 1. Let $P = 11 \cdot 13 \cdot 17 \cdot 19 = 46189$ and $d = 5$, then there are eight absolute Lucas pseudoprimes for $d = 5$ of the form Pqr.

1. $P \cdot 57349 \cdot 331111621 = 877079242172199781$
2. $P \cdot 709 \cdot 4093501 = 134053974841501$
3. $P \cdot 1009 \cdot 378901 = 17658567813601$
4. $P \cdot 230941 \cdot 29144629 = 310883829596647021$
5. $P \cdot 2161 \cdot 231589 = 23115923797681$
6. $P \cdot 23 \cdot 83 = 88174801$
7. $P \cdot 161659 \cdot 577351 = 4311003447437401$
8. $P \cdot 1459 \cdot 2251 = 983368161419501$

The divisor method requires constructing about 7.8 million $D\Delta$ pairs. The CD method requires the construction of about 4.8 billion CD pairs. By prioritizing all other checks first, the $D\Delta$ method used only 68 primality checks (and 16 were required to get the above output).

6. Algorithms for Large Preproducts

6.1. Distinguishing "Large" from "Small"

So far, the only approach to find $n < B$ has been to construct a preproduct $P = P_{k-2}$ and use Theorem 4 to find the remaining two primes in time that is essentially linear in P. This approach has the benefit that it is not dependent on k or $\lambda_d(P)$. However, as P grows in size (with respect to B) it is more and more likely to create absolute Lucas pseudoprimes outside the tabulation bound. We may discard these but there is no obvious way to improve the asymptotic cost and only generate the $q = p_{k-1}$ and $r = p_k$ of the correct sizes. At some point it will be more efficient to exhaustively generate the candidate q values via a look-up table or with a sieve. In either case, the cost will be roughly linear in the length of the interval the primes lie in (differing by $\log B$ factors depending on the method used).

Since the algorithms in Section 5 allows us to create all q in time roughly linear in P, the bound $q < 2(P+1)^2$ is not helpful in figuring out when to switch to an exhaustive search because this implies a search cost that is roughly quadratic in P. The bound $Pq^2 < B$ implies $q < (B/P)^{1/2}$. So, P and $(B/P)^{1/2}$ equalize around $P = B^{1/3}$. For the "large" case, we will assume that $P > X > B^{1/3}$ where X is some chosen cross-over point. We will construct q by exhaustive search for primes in the interval $(p_{k-2}, \sqrt{B/P}) \subset (p_{k-2}, \sqrt{B/X}) \subset [1, B^{1/3})$. With q, we know $P_{k-1} = Pq$ and $\lambda_d(P_{k-1})$, and will use this information to analyze the cost of finding $r = p_k$. The difficulty with getting an asymptotic estimate of the total

cost of the tabulation of the "large case" is that not much is known about the asymptotic behavior of $\lambda_d(P_{k-1})$. For example, if $\lambda_d(P_{k-1})$ were within a fixed constant multiple ℓ of P_{k-1}, then there would only be 2ℓ candidate values of p_k to check. However, there is no reason to believe that this could happen. Since $\lambda_1(n)$ can be very small with respect to n, it would be reasonable to believe that $\lambda_d(n)$ has the same property.

6.2. Finding p_k Given P_{k-1}

There are a few approaches for finding p_k given P_{k-1}. We describe what we did and discuss some valid options that were not implemented.

We use congruence in Theorem 3(ii) to describe p_k up to a sign:

$$p_k \equiv \delta_{P_{k-1}} \delta_{p_k} P_{k-1}^{-1} \pmod{\lambda_d(P_{k-1})}.$$

This means that there are two residue classes r_1, r_2 modulo $\lambda_d(P_{k-1})$ to consider and the number of candidates to be considered in this arithmetic progression is

$$\min\left\{ \left\lceil \frac{P_{k-1} - \delta_{P_{k-1}}}{\lambda_d(P_{k-1})} \right\rceil, \left\lceil \frac{B}{P_{k-1}\lambda_d(P_{k-1})} \right\rceil \right\}.$$

The first term comes from $(p_k - \delta_{p_k}) \mid (P_{k-1} - \delta_{P_{k-1}})$ implying $p_k - \delta_{p_k} \leq P_{k-1} - \delta_{P_{k-1}}$. The second term comes from the fact that $P_{k-1}p_k < B$ and we compute the greatest multiple of $\lambda_d(P_{k-1})$ for which the inequality holds. This is all we implemented; creating candidates in arithmetic progression is "fast," memory efficient, and easy to program.

However, there is an asymptotically superior choice that we did not implement. This is because the worst-case arises when $\lambda_d(P_{k-1})$ is really small. For these cases, one should view the problem as integer factorization rather than sieving in an arithmetic progression. That is, we want to find factors of $P_{k-1} - \delta_{P_{k-1}}$. On this view, the congruence

$$p_k \equiv \delta_{P_{k-1}} \delta_{p_k} P_{k-1}^{-1} \pmod{\lambda_d(P_{k-1})}$$

can happen to make the factoring problem easier. This happens whenever $\lambda_d(P_{k-1})$ is large enough (see results on *divisors in residue classes* in Section 4.2.3 of [3]). When $\lambda_d(P_{k-1})$ is particularly small, then testing candidates in arithmetic progression could be worse than trial division because there could be $O(P_{k-1}/\lambda_d(P_{k-1})) = O(P_{k-1})$ candidates to check. Trial division would only check $O(\sqrt{P_{k-1}})$ candidates and this is among the slowest of factoring algorithms. Any asymptotically faster integer factorization algorithm will find candidates for p_k in an asymptotically superior way.

7. Implementation, Statistics, and Questions

7.1. Implementation Details

In Section 5.3, we required divisors of integers in the interval $[1, 2P-1]$. One option was a large look-up table with prime factorizations of every integer in $[1, 2X]$ to be used for every $P < X$. This table could be used to easily check the admissibility of P and find all divisors of $(P - \delta_P)(\delta_q P - \delta_r D)$. However, this table would be very space intensive. Instead, we opted for two incremental sieves. One sieve was used to find admissible P and it always stored the factors of $P-1$ and $P+1$ so that the factors of $P - \delta_P$ would be accessible. For any admissible P, another incremental sieve was instantiated to factor integers in $[1, 2P-1]$ for the $\delta_q P - \delta_r D$ term. This approach uses only $O(\sqrt{X})$ space. If X is chosen as suggested in Section 6.1, this is $O(B^{1/6})$ space.

For the four tabulations with $d = 5, -7, -11$ and 13, we chose $X = 6 \cdot 10^6 > 2^{64/3}$. For every $P < X$, we used either the CD method or the $D\Delta$ method. The choice was made on a per D basis by choosing which inner loop would create fewer candidates. The program computed all possible $n = Pqr$, and we used post-processing to eliminate $n > 2^{64}$. Our choice of X means that there are no cases for $k = 3$ that need to be accounted as large. We wrote nine distinct programs for the large case (one for each $3 < k < 13$). For $k \geq 13$, tabulations would have been empty for $B = 2^{64}$. We used a precomputed list of primes in the interval $[1, \sqrt{B/X})$. If $X > B^{1/3}$, this requires $O(B^{1/3})$ storage. For each $k > 3$, we keep track of $k-1$ pointers in the array. At each level, we make sure that the implied product is bounds admissible. And at the $k-2$ level, we also insure that the product exceeds X.

The code and other supplementary information may be found at
`https://github.com/Chelmreich/Absolute-Lucas-Pseudoprimes`.

7.2. Statistics and Comparison to Carmichael Numbers

We let $C_d(k, B)$ be the function that counts the number of absolute Lucas pseudoprimes less than B, where d is the discriminant of the family of Lucas sequences and k is the number of prime factors. There seems to be more absolute Lucas pseudoprimes than Carmichael numbers. The presence of the product of twin primes plays a significant role in this count. Letting $\alpha = \log_B(C_d(B))$, then the least order of magnitude for which $\alpha > 1/3$ is 15 for Carmichael numbers. But for the other discriminants this threshold is crossed at $13, 8, 9$, and 11 (ordered by discriminant).

We are not entirely sure why this is. Our expectation was that the exclusion of primes dividing d from admissible preproducts would cause there to be fewer of these numbers. Since the actual asymptotic behavior of Carmichael numbers is still subject to many open questions (e.g., [11]), we believe the asymptotic counts of these

numbers would be subject to the same problems. See Tables 1-8 in the Appendix for information on the tabulations organized by the count of prime divisors.

One curious feature of absolute Lucas pseudoprimes is that they do not exhibit uniform distribution of Jacobi symbols by their prime factorizations. For example, with $k = 3$ and $d = 5$, there are eight possible ways the Jacobi symbols may appear. The case $\delta_{p_1} = \delta_{p_2} = \delta_{p_3} = 1$, which are also Carmichael numbers, had 32227 numbers in it. While the case corresponding to $\delta_{p_1} = \delta_{p_2} = -\delta_{p_3} = 1$ only had 1146 numbers in it.

7.3. Questions

In [1], the authors revisited the Baillie-PSW primality test with an aim of strengthening it. Could the unbalanced nature of the distribution of Jacobi symbols in absolute Lucas pseudoprimes imply that there is a better choice of families of Lucas sequences for this test?

It is always desirable to have sharper asymptotic estimates for algorithms. In our case, this would require better bounds on $\lambda_d(n)$. Does this generalized Carmichael function have the same asymptotic behavior as $\lambda_1(n)$? See Theorem 2 and Theorem 3 of [7] or Theorem 5 of [8] for asymptotic results on the Carmichael function.

Acknowledgements. We both thank Anthony Gurovski for his initial contributions, which included a tabulation up to 10^{17} for $d = 5$. We are grateful to Hugh Williams' encouragement and his comments on a preliminary draft of this work. We are also thankful to the anonymous referee.

References

[1] R. Baillie, A. Fiori, and S. Wagstaff, Strengthening the Baillie-PSW primality test, *Math. Comp.* **90** (330) (2021), 1931-1955.

[2] R. Baillie, and S. Wagstaff, Lucas pseudoprimes, *Math. Comp.* **35** (152) (1980), 1391-1417.

[3] R. Crandall and C. Pomerance, *Prime Numbers: A Computational Perspective*, Springer, New York, 2005.

[4] H.G.W.H. Beeger, On composite numbers n for which $a^{n-1} \equiv 1 \pmod{n}$ for every a prime to n, *Scripta Math.* **16** (1950), 133-135.

[5] H.J.A. Duparc, On Carmichael numbers, *Simon Stevin* **29** (1952), 21-24.

[6] P. Erdős, Some asymptotic formulas in number theory, *J. Indian Math. Soc.* **12** (1948), 75-78.

[7] P. Erdős, C. Pomerance, and E. Schmutz, Carmichael's lambda function, *Acta Arithmetica* **58** (4) (1991), 365-385.

[8] J. Friedlander, C. Pomerance, and I. Shparlinski, Period of the power generator and small values of the Carmichael function, *Math. Comp.* **70** (236) (2001), 1591–1605.

[9] T. Granlund and the GMP development team, *GNU MP: The GNU Multiple Precision Arithmetic Library*. http://gmplib.org/.

[10] J. Grantham, Frobenius pseudoprimes, *Math. Comp.* **70** (234) (2000), 873–891.

[11] A. Granville and C. Pomerance, Two contradictory conjectures concerning Carmichael numbers, *Math. Comp.* **71** (238) (2001), 883-908.

[12] G. Jaeschke, On strong pseudoprimes to several bases, *Math. Comp.* **61** (204) (1993), 915–926.

[13] M. Joye and J.-J. Quisquater, Efficient computation of full Lucas sequences, *Electron. Lett.* **32** (6) (1996), 537-538.

[14] D.H. Lehmer, An extended theory of Lucas' functions, *Ann. of Math.* (2) **31** (1930), no. 3, 419-448.

[15] R. G. E. Pinch, The Carmichael numbers up to 10^{15}, *Math. Comp.* **61** (203) (1993), 381-391.

[16] R. G. E. Pinch, The Carmichael numbers up to 10^{21}, https://tinyurl.com/45w4ec2w.

[17] M. Rabin, Probabilistic algorithm for testing primality, *J. Number Theory* **12** (1980), 128-138.

[18] A. Shallue and J. Webster, Tabulating Carmichael numbers $n = Pqr$ with small P, *Res. Number Theory* **8** (2022) (4), Paper no. 84, 11 pp.

[19] J. Sorenson and J. Webster, Two algorithms to find primes in patterns, *Math. Comp.* **89** (324) (2020), 1953-1968.

[20] H. C. Williams, *Édouard Lucas and Primality Testing*, Wiley, New York, 1998.

[21] H. C. Williams, On numbers analogous to the Carmichael numbers, *Canad. Math. Bull.* **20** (1) (1977), 133-143.

Appendix

B	$k=2$	3	4	5	6	7	$C_5(B)$	α
10^3	1	0	0	0	0	0	1	0
10^4	1	2	0	0	0	0	3	0.1193
10^5	1	7	0	0	0	0	8	0.1806
10^6	9	22	3	0	0	0	34	0.2552
10^7	24	50	24	2	0	0	100	0.2857
10^8	64	102	89	18	1	0	274	0.3047
10^9	159	189	249	106	7	0	710	0.3168
10^{10}	414	356	512	358	71	0	1711	0.3233
10^{11}	1053	633	1008	1040	316	17	4067	0.3281
10^{12}	2734	1110	1857	2703	1268	180	9855	0.3328
10^{13}	7301	2038	3344	6226	4174	966	24108	0.3371
10^{14}	19674	3737	5649	13287	12078	4288	59209	0.3409
10^{15}	53561	6754	9462	26821	31472	15721	146774	0.3444
10^{16}	146953	12215	15639	51121	76397	50690	367518	0.3478
10^{17}	407779	22004	25186	94748	173721	148482	933074	0.3512
10^{18}	1142128	39974	40155	169243	376784	404815	2404810	0.3545
10^{19}	3220913	73298	62991	293565	783905	1033279	6272286	0.3578
2^{64}	4247414	86227	70917	338435	946862	1313728	8111918	0.3586

Table 1: Values of $C_5(k, B)$ for $2 \le k \le 7$ and $C_5(B)$

B	$k=8$	9	10	11	12
10^{12}	3	0	0	0	0
10^{13}	59	0	0	0	0
10^{14}	490	6	0	0	0
10^{15}	2844	138	1	0	0
10^{16}	13280	1201	22	0	0
10^{17}	53529	7338	287	0	0
10^{18}	191645	37528	2501	37	0
10^{19}	621182	165609	17013	526	5
2^{64}	839626	240258	27437	1004	10

Table 2: Values $C_5(k, B)$ for $8 \le k \le 12$

B	2	3	4	5	6	7	$C_{-7}(B)$	α
10^3	0	0	0	0	0	0	0	0
10^4	1	4	0	0	0	0	5	0.1747
10^5	1	17	0	0	0	0	18	0.2511
10^6	4	53	10	0	0	0	67	0.3043
10^7	15	115	74	8	0	0	212	0.3323
10^8	37	249	267	61	0	0	614	0.3485
10^9	94	509	746	316	16	0	1684	0.3585
10^{10}	239	965	1770	1272	168	0	4414	0.3645
10^{11}	623	1773	3777	4565	1128	70	11936	0.3706
10^{12}	1595	3248	7458	14516	5260	602	32684	0.3762
10^{13}	4320	5863	14052	41215	19405	3696	88689	0.3806
10^{14}	11756	10490	25389	99562	61541	18690	229020	0.3828
10^{15}	32071	19211	44127	207979	175819	79626	570903	0.3838
10^{16}	88111	34589	75146	390112	459693	291488	1410927	0.3843
10^{17}	243992	62833	124996	684936	1127659	958164	3559042	0.3854
10^{18}	684583	115274	203560	1154665	2609781	2870274	9183044	0.3868
10^{19}	1930996	211336	326436	1902266	5763746	7969591	24136647	0.3686
2^{64}	2546823	248473	369654	2167587	7063176	10340609	31283689	0.3890

Table 3: Values of $C_{-7}(k, B)$ for $2 \leq k \leq 7$ and $C_{-7}(B)$

B	8	9	10	11	12
10^{12}	5	0	0	0	0
10^{13}	138	0	0	0	0
10^{14}	1568	24	0	0	0
10^{15}	11676	392	2	0	0
10^{16}	67197	4544	47	0	0
10^{17}	318930	36532	999	1	0
10^{18}	1304962	227523	12309	113	0
10^{19}	4752342	1173485	104118	2324	7
2^{64}	6595966	1770773	175859	4753	16

Table 4: Values of $C_{-7}(k, B)$ for $8 \leq k \leq 12$

B	2	3	4	5	6	7	$C_{-11}(B)$	α
10^3	1	1	0	0	0	0	2	0.1003
10^4	3	3	0	0	0	0	6	0.1945
10^5	6	6	3	0	0	0	15	0.2352
10^6	8	25	14	1	0	0	48	0.2802
10^7	15	63	51	7	1	0	137	0.3052
10^8	41	157	156	27	1	0	382	0.3228
10^9	108	317	421	155	15	0	1016	0.3341
10^{10}	276	617	990	693	80	0	2656	0.3424
10^{11}	694	1215	2157	2452	516	16	7050	0.3498
10^{12}	1795	2292	4373	7798	2493	230	18981	0.3565
10^{13}	4899	4171	8535	22623	9547	1575	51381	0.3624
10^{14}	13183	7514	15701	56048	31758	8307	133036	0.3660
10^{15}	35654	13667	27741	119135	92145	37187	329925	0.3679
10^{16}	97750	24427	47899	226410	247963	143068	816901	0.3695
10^{17}	271562	44398	80166	402574	618647	485980	2059048	0.3714
10^{18}	760653	80786	131666	684709	1456771	1498393	5321599	0.3737
10^{19}	2147345	149154	212589	1141736	3259866	4257511	14044816	0.3762
2^{64}	2831298	175235	241194	1305333	4007294	5554172	18225511	0.3767

Table 5: Values of $C_{-11}(k, B)$ for $2 \leq k \leq 7$ and $C_{-11}(B)$

B	8	9	10	11	12
10^{13}	31	0	0	0	0
10^{14}	520	5	0	0	0
10^{15}	4299	97	0	0	0
10^{16}	27953	1426	5	0	0
10^{17}	142421	13079	221	0	0
10^{18}	614240	90839	3524	18	0
10^{19}	2332830	507737	35554	494	0
2^{64}	3268653	778688	62510	1132	2

Table 6: Values of $C_{-11}(k, B)$ for $8 \leq k \leq 12$

B	2	3	4	5	6	7	$C_{13}(B)$	α
10^3	2	0	0	0	0	0	2	0.1003
10^4	2	0	0	0	0	0	2	0.0753
10^5	5	4	1	0	0	0	10	0.2
10^6	10	16	8	0	0	0	34	0.2552
10^7	18	39	30	1	0	0	88	0.2778
10^8	59	87	95	20	0	0	261	0.3021
10^9	135	182	230	106	8	0	661	0.3134
10^{10}	335	360	494	359	121	4	1673	0.3223
10^{11}	861	669	1012	1175	2115	22	5854	0.3425
10^{12}	2218	1213	1892	3358	12776	133	21590	0.3612
10^{13}	5972	2190	3349	8860	44394	727	65492	0.3705
10^{14}	15996	3921	5722	20351	116366	3207	165573	0.3728
10^{15}	43387	7065	9512	40233	252535	12101	364973	0.3708
10^{16}	119760	12767	15772	71695	483640	40779	745471	0.3670
10^{17}	333122	22825	25616	119960	857358	125490	1491078	0.3631
10^{18}	933600	41533	40764	191781	1434794	351699	3029748	0.3601
10^{19}	2634300	76327	63903	300111	2326807	920304	6487389	0.3585
2^{64}	3473895	89688	71861	337321	2633462	1177125	8028538	0.3584

Table 7: Values of $C_{13}(k, B)$ for $2 \leq k \leq 7$ and $C_{13}(B)$

B	8	9	10	11	12
10^{14}	10	0	0	0	0
10^{15}	106	34	0	0	0
10^{16}	587	469	2	0	0
10^{17}	2782	3858	67	0	0
10^{18}	11463	23179	932	3	0
10^{19}	43598	113433	8493	113	0
2^{64}	61566	168929	14437	254	0

Table 8: Values of $C_{13}(k, B)$ for $8 \leq k \leq 12$