A. Shallue, J. Webster *Res. Number Theory* (2022) 8:84
https://doi.org/10.1007/s40993-022-00384-z

Research in Number Theory

RESEARCH

Tabulating Carmichael numbers $n = Pqr$ with small P

Andrew Shallue[1] and Jonathan Webster[2*]

*Correspondence:
jewebste@butler.edu
[2]Butler University, Indianapolis, USA
Full list of author information is available at the end of the article

Abstract

We revisit the problem of tabulating Carmichael numbers. Carmichael numbers have been tabulated up to 10^{21} using an algorithm of Pinch (Math Comp 61(203):381–391, 1993). In finding all Carmichael numbers with d prime factors, the strategy is to first construct pre-products P with $d-2$ prime factors, then find primes q and r such that Pqr satisfies the Korselt condition. We follow the same general strategy, but propose an improvement that replaces an inner loop over all integers in a range with a loop over all divisors of an intermediate quantity. This gives an asymptotic improvement in the case where P is small and expands the number of cases that may be accounted as small. In head-to-head timings this new strategy is faster over all pre-products in a range, but is slower on prime pre-products. A hybrid approach is shown to improve even the case of prime pre-products.

Keywords: Carmichael numbers, Tabulation algorithms, Pseudoprimes

Mathematics Subject Classification: 11Y11, 11Y16, 11Y70

1 Introduction

Fermat's Little Theorem states that when p is prime that $a^p \equiv a \pmod{p}$ for any integer a. The converse of this theorem is a computationally efficient way to detect if an integer is composite. That is, for a given n we pick an $a < n$ and compute $a^n \pmod{n}$. If the result is not a, we may conclude that the number is composite. Unfortunately, there are composite numbers for which the computed congruence is a, e.g. $n = 341$ and $a = 2$. Even worse, there are composite numbers for which the congruence is a for any $a < n$. The least example is $n = 561$. These numbers are sometimes called absolute Fermat pseudoprimes but are more often called *Carmichael numbers* due to Carmichael who published the first examples [2].

For a background and survey on Carmichael numbers we refer to [13]. Our chief concern regards tabulation and the key source is Pinch's *The Carmichael numbers up to* 10^{15} [8]. This paper also gives a background of prior tabulations of Carmichael numbers. In a series of reports, he announced further tabulations using the same algorithm with the most recent being up to 10^{21} [9–12].

The tabulation method that Pinch employs uses a bifurcated approach. For a given number P, we search for primes q and r so that Pqr is a Carmichael number. Depending

on properties of P there are two different strategies for finding q and r. Roughly speaking, the strategies change when P goes from being "small" to "large." We follow in Pinch's footsteps and offer four significant contributions. These are

- an asymptotic analysis of his algorithm for when P is "small",
- a new asymptotically superior way of dealing with the "small" case,
- timing information supporting the theoretical claims, and
- a tabulation for all $P < 3 \cdot 10^6$.

In the "small" case, Pinch's method was only practical when P was a prime. In this case, the computation time was nearly linear in P. If P was composite, the computation could approach a time quadratic in P (see Theorem 4). Our improvement removes this dependence on the number of prime factors of P and remains nearly linear for all P (see Theorem 5). Due to the fact that the number of Carmichael numbers of the form Pqr with P fixed is finite, we do a new style of tabulation on P values rather than on the absolute size of Carmichael numbers. This tabulation includes Carmichael numbers not found in [12], e.g. with $P = 999983$,

$$1344\ 14285\ 88839\ 69679\ 08345\ 46298\ 33201 = P \cdot 1000709986897 \cdot 1343212046747951.$$

The rest of the paper is organized as follows. Section 2 establishes notation, relevant theorems, and the motivation for our tabulation. Section 3 analyzes the prior approaches and shows an asymptotically and practically improved algorithm. Section 4 discusses the implications in the context of a tabulation of all Carmichael numbers less than B. We conclude with comments on the implementation and other questions in Sect. 5.

2 Notation

The tabulation methods rely on Korselt's Criterion.

Theorem 1 (Korselt's Criterion) *A composite number n is a Carmichael number if and only if n is squarefree and $(p-1) \mid (n-1)$ for all prime divisors p of n.*

The tabulation methods construct n in factored form. Our goal will be to construct square-free odd numbers n which we will then test with Korselt's criterion. Let d be the number of factors in n and then $d > 2$. Let $P_k = \prod_{i=1}^{k} p_i$ and $Q_{d-k} = \prod_{i=k+1}^{d} p_i$ where $p_i < p_j$ iff $i < j$. The primary tabulation methods concern P_{d-2} and constructing or searching for p_{d-1} and p_d. Since these quantities are used so often, we will suppress subscripts and write these quantities as P, q, and r respectively.

Theorem 2 (Proposition 1 of [8]) *Let n be a Carmichael number less than B.*

(2.1) Let $k < d$. Then $p_{k+1} < (B/P_k)^{1/(d-k)}$ and $p_{k+1} - 1$ is relatively prime to p_i for all $i \le k$.

(2.2) Let $L = \operatorname{lcm}\{p_1 - 1, \ldots, p_{d-1} - 1\}$. Then $P_{d-1} r \equiv 1 \pmod{L}$ and $r - 1$ divides $P_{d-1} - 1$.

(2.3) Each p_i satisfies $p_i < \sqrt{n} < \sqrt{B}$.

Theorem 2.1 places significant restrictions on valid P. For example $P = 21$ is inadmissible because $3 \mid (7-1)$. In a tabulation of all Carmichael numbers less than B, bounds in 2.1 and 2.3 are important. The computational way to understand Theorem 2.2 is that we

may search for r in the residue class $(P_{d-1})^{-1} \pmod{L}$. This is a matter of sieving in the interval $(q, P_{d-1} - 1)$ with the residue class of $(P_{d-1})^{-1}$ with step size of L.

Beeger proved that if P is prime, then there are only a finite number of Carmichael numbers of the form Pqr for a given P [1]. In [4], Duparc generalized this result to composite P. While there is an implied algorithm in [4], Pinch restates the theorem and makes the algorithm explicit. We state his version of the theorem and follow his notation.

Theorem 3 (Proposition 2 of [8]) *There are integers $2 \leq D < P < C$ such that, putting $\Delta = CD - P^2$, we have*

$$q = \frac{(P-1)(P+D)}{\Delta} + 1, \tag{1}$$

$$r = \frac{(P-1)(P+C)}{\Delta} + 1, \tag{2}$$

$$P^2 < CD < P^2 \left(\frac{p_{d-2} + 3}{p_{d-2} + 1} \right). \tag{3}$$

Corollary 1 *There are only finitely many Carmichael numbers of the form Pqr for a given P.*

This motivates the present work. Since the number of Carmichael numbers with fixed P is finite, once the computation has been done, such a P never needs to be revisited (except possibly for independent verification). Thus, we want to tabulate all Carmichael numbers of the form Pqr for $P < X$.

The prior tabulations focused on finding all Carmichael numbers $n < B$. Pinch used Theorem 3 when P was "small" and used Theorem 2.2 when P was "large". Invoking Theorem 2.2 involves two things. First, one has to find q given P which may be done by standard prime sieving. Second, given q and P, one has to use Theorem 2.2 to find r. This means that if the first step (prime sieving for q) is more costly than invoking Theorem 3, then we may account P as "small."

Many of our results depend on $\tau(n) = \sum_{d|n} 1$, the function that counts the number of divisors of n. Dirichlet established

$$\sum_{n \leq x} \tau(n) = x \ln x + (2\gamma - 1)x + O(\sqrt{x}).$$

We now give a thorough explanation of the "small" case and show how it may be significantly improved which comprises the bulk of our new results.

3 The case for "small" *P*

In the reports [10–12], Pinch writes of Theorem 3 "in practice [the theorem] was useful only when $d = 3$." On a cursory reading of the theorem and resulting algorithm, it might not be clear why this is. We explain his result more thoroughly and provide justification for why using Theorem 3 when $d > 3$ becomes computationally expensive. Ideally, one would not want this dependence on d. If you have two different P but of comparable size then the timing results depend strongly on the number of prime factors of P. Indeed, we show that the cost per P can range from being nearly linear in P to being approaching quadratic in P. Ideally, an algorithm would only depend on the size of P. Intuitively, these two numbers have the same input size and we would want an algorithm that is dependent on the input size alone. We offer this improvement and remove the reliance on d. At

first, it might appear that the new approach offers no asymptotic improvement to the case $d = 3$. However, we show that we can use a hybrid method to improve even this case.

In the context of a tabulation of all Carmichael numbers $n < B$, this makes the ability to switch between "small" and "large" more intuitive because we do not have to deal with the added input of d.

3.1 Generating *CD* pairs in time $O(P^{2-\frac{1}{d-2}} \ln P)$

To use Theorem 3 in a computational way, Pinch suggests looping over $2 \le D < P$, then looping over all C as permitted by the inequality in Theorem 3.3. With C, D, we create Δ. Then the inner loop uses C, D, Δ to check that q and r are rational primes and that Pqr is a Carmichael numbers using Korselt's criterion. We ignore the asymptotic cost of these checks and only concern ourselves with the asymptotic count of the times the inner loop is entered.

Theorem 4 *Fix a pre-product P. Then all valid CD pairs may be created in time $O(L_P \ln P) = O(P^{2-\frac{1}{d-2}} \ln P)$.*

Proof The double nested loop used to create C and D is akin to using a sieve of Eratosthenes to find composite numbers on the interval $[P^2, P^2\left(\frac{p_{d-2}+3}{p_{d-2}+1}\right)]$. The asymptotic count of entering the inner loop is determined by the length of the interval.

Define L_P so that $P^2\left(\frac{p_{d-2}+3}{p_{d-2}+1}\right) = P^2 + L_P$. When P is prime, we have

$$P^2\frac{P+3}{P+1} = P^2 + 2P - 2 + \frac{2}{P+1}.$$

On the other hand, let $P = P_{d-3}p_{d-2}$ with $d > 3$. If p_{d-2} is minimal then P_{d-3} may be nearly as large as $P^{1-\frac{1}{d-2}}$. So,

$$P^2\left(\frac{p_{d-2}+3}{p_{d-2}+1}\right) = (P_{d-3}p_{d-2})^2\left(\frac{p_{d-2}+3}{p_{d-2}+1}\right) = P^2 + 2P_{d-3}^2(p_{d-2}-1) + \frac{2P_{d-3}^2}{p_{d-2}+1}.$$

Then $2P - 2 < L_P < 2PP_{d-3} < 2P^{2-\frac{1}{d-2}}$. In summary, L_P is nearly linear in P when P_{d-3} is small but L_P can approach quadratic in P as P_{d-3} becomes large with respect to P. In particular, L_P is linear in P when $d = 3$.

The cost of sieving an interval of length L_P by step size D is $O(L_P/D)$. Summing over D, gives

$$\sum_{D=2}^{P-1} O(L_P/D) = O(L_P \ln P) = O(P^{2-\frac{1}{d-2}} \ln P)$$

as claimed. □

So, when $d = 3$, $O(L_P \ln P) = O(P \ln P)$ and this helps explain why Pinch found the *CD* method only useful for the case $d = 3$. In the subsection below we will show how to restore linearity in P and remove the dependence on d.

3.2 Generating $D\Delta$ pairs in average time $O(P(\ln P)^2)$

Our approach is to change the order in which the various quantities are generated. In the above, *CD* pairs were created and then the integrality of q is checked. We follow Pinch looping over $2 \le D < P$. For a given D, loop over divisors δ of $(P-1)(P+D)$. Then $(P-1)(P+D)/\delta$ will be an integer. If $C = (P^2+\delta)/D$ is integral, then we have a *CD*

pair that would have been found from the prior subsection and δ would be as Δ above. We trade the guarantee of an integral C in favor of the guarantee of an integral q. We have to account for two things. First, that the number of times the inner loop is entered is asymptotically fewer. This is pretty simple to do as $\tau((P-1)(P+D))$ is expected to be smaller than L_P/D. Second, that the cost of obtaining the list of divisors is not expensive. This is also achieved by some variant of the sieve of Eratosthenes.

Through the use of a sieve of Eratosthenes, we can create factorizations of all numbers in the interval $[P+2, 2P-1]$ in time $O(P \ln \ln P)$. This represents a lower order cost in the computation for a given P. Depending on the nature of the implementation the storage space can be $O(P^{1/3+\epsilon})$ using [7], $O(P^{1/2+\epsilon})$ using standard segmenting or an incremental sieve [14], or a naive implementation requiring $O(P^{1+\epsilon})$ storage. Given factorizations of $(P-1)$ and $(P+D)$, it is easy to construct all possible divisors of $(P-1)(P+D)$. We should also note that having access to this information is not an unreasonable change from Pinch's own version. Generating primes is, after all, usually done with a sieve of Eratosthenes. Our implementation uses the incremental sieve described in [14], which costs $O(P^{1/2+\epsilon})$ to initialize and has average cost $O(\ln \ln P)$ to update. The initialization cost is cheaper than the cost of finding C when $D = 2$, and the updating cost is asymptotically smaller than the checks required by the inner loop, so it worth paying in order to minimize entry into the inner loop.

The new approach is to loop over all $2 \le D < P$. The inner loop is now over divisors of $(P-1)(P+D)$. In the inner loop, we check the integrality of C, the integrality of r, that Korselt's criteria holds, the primality of r, and finally the primality of q. When all checks pass, we output Pqr as a Carmichael number. While the ordering of these checks does not matter from an asymptotic point of view, we recommend this ordering. The checks done with a single division are easier to do and more likely to fail which allow us to avoid the more costly primality checks.

The average number of divisors of $(P-1)(P+D)$ will frequently be significantly less than L_P/D. The former has a count on average of $O((\ln P)^2)$ while the latter can be nearly quadratic in P when D is small. If we want to tabulate all Carmichael numbers of the form Pqr with $P < X$ then we may analyze the following sum

$$\begin{aligned}
&\textstyle\sum_{P<X} \sum_{D=2}^{P-1} \tau((P-1)(P+D)) \\
&< \left(\sum_{P<X} \tau(P-1)\right)\left(\sum_{D<X} \tau(P+D)\right) \\
&< \left(\sum_{P<X} \tau(P)\right)\left(\sum_{D<2X} \tau(D)\right) \\
&= 2X^2(\ln X)^2 + O(X^2 \ln X).
\end{aligned}$$

The above gives the following theorem.

Theorem 5 *The number of $D\Delta$ pairs used to tabulate all Carmichael number of the form Pqr for $P < X$ is $O(X^2(\ln X)^2)$.*

The average number of $D\Delta$ pairs per P is then on the order of $O(P(\ln P)^2)$. This removes the strong dependence on the number of divisors of P. In the case of $d = 3$, this seems to offer no improvement but we will show in the next subsection that this case can be improved by this method.

The full set of divisors of $(P-1)(P+D)$ need not be considered. One can discard large divisors because

$$p_{d-2} < q = \frac{(P-1)(P+D)}{\Delta} + 1 \Rightarrow \Delta < \frac{(P-1)(P+D)}{p_{d-2}-1}.$$

In the case of $d = 3$ this inequality is $\Delta < P + D$.

Example 1 Let $P = 5 \cdot 19 \cdot 23 \cdot 29 = 63365$, then there are four Carmichael numbers of the form Pqr. They are

(1) $P \cdot 683 \cdot 2545783 = 110177147679985$
(2) $P \cdot 2297 \cdot 36037 = 5245163907985$
(3) $P \cdot 37 \cdot 137 = 321197185$
(4) $P \cdot 70168253 \cdot 254447257 = 1131326282391998510665$.

The third number is the smallest Carmichael number with exactly six prime factors. Generating these four numbers requires checking about 9 million $D\Delta$ pairs or about 2.8 billion CD pairs.

3.3 An improvement to the $d = 3$ case

Given the prime factorization of $(P-1)$ and $(P+D)$, it is easy to compute how many total divisors we would have to consider prior to actually constructing the divisors. Supposing we have

$$(P-1)(P+D) = \prod_k p_i^{e_i} \quad \text{then} \quad \tau\left((P-1)(P+D)\right) = \prod_k (e_i+1).$$

Since we can easily compute both $\tau((P-1)(P+D))$ and L_P/D, we may choose to enter the corresponding innerloop based on which quantity is smaller: L_P/D or $\tau\left((P-1)(P+D)\right)$. Once L_P/D starts being smaller, one can entirely abandon the incremental sieve and assume L_P/D will continue to stay smaller.

This implies that the $d = 3$ case may be done asymptotically faster by this dynamic choice. In Sect. 3.1 and in the $d = 3$ case, L_P/D ranges from P to 2 as D ranges from 2 to $P-1$. That is, there are many CD pairs when D is small and this changes to having very few CD pairs as D approaches $P-1$. However, in the approach in Sect. 3.2, the number of divisors of $\tau((P-1)(P+D))$ remains relatively constant throughout the computation where the variance is determined by the number of divisors of $(P+D)$. The asymptotic cost for a given P is now

$$\sum_{D=2}^{P-1} \min\{2P/D, \tau\left((P-1)(P+D)\right)\}$$

Conjecture 1 *When P is a prime, all Carmichael numbers of the form Pqr may be found by considering $O(P \ln \ln P)$ many $D\Delta$ or CD pairs.*

This conjecture is quite believable because if we replace $\tau((P-1)(P+D))$ with its average value, then the conjecture is justified by the following. We may choose to compute Pqr with the $D\Delta$ method when D is small and switch to the CD method when D is large. Letting this cross-over occur at $D = \frac{P}{(\ln P)^2}$ and using the average value of $\tau((P-1)(P+D))$ which

is $(\ln P)^2$ we get:

$$\sum_{D=2}^{\frac{P}{(\ln P)^2}} (\ln P)^2 + \sum_{D=\frac{P}{(\ln P)^2}}^{P-1} 2P/D = O(P) + O(2P \ln\ln P) = O(P \ln\ln P).$$

Example 2 Let $P = 65003$ a prime, then there are ten Carmichael numbers of the form Pqr. They are

(1) $P \cdot 384226823 \cdot 1387549787527 = 346552994315684228595751 63$
(2) $P \cdot 260009 \cdot 149569603 = 2527930457246474281$
(3) $P \cdot 4485139 \cdot 1443304409 = 420791778351741348553$
(4) $P \cdot 4255030921 \cdot 605229266867 = 167400226720595416380338521$
(5) $P \cdot 2145067 \cdot 123503801 = 17220850085262054001$
(6) $P \cdot 11960369 \cdot 628504339 = 488636899246608538273$
(7) $P \cdot 845027 \cdot 27300841 = 1499615814744258121$
(8) $P \cdot 3073667 \cdot 36326833 = 7258013177193134833$
(9) $P \cdot 260009 \cdot 845027 = 14282109784670729$
(10) $P \cdot 845027 \cdot 1950061 = 107115466344644941$

The average value of $\tau((P-1)(P+D)/2)$ is around 45 and $\lfloor L_P/D \rfloor = 45$ when $D = 2827$. After some point, say $D = 6000$, the computation of $\tau((P-1)(P+D)/2)$ can cease entirely and the computation is finished using only generation of CD pairs as in Sect. 3.1.

In reality, the trade-off between the two methods is also determined by timing information. There are at least two timing issues that were ignored in the above analysis. First, the method of primality testing differs from the $D\Delta$ to the CD method (see Sect. 5.2). Second, the cost of constructing the CD pairs has almost no overhead cost compared to the overhead cost associated with the $D\Delta$ method. Once the CD method becomes dominant, we can cease the overhead with the $D\Delta$ method entirely and just safely assume that L_P/D will remain smaller. This means that the computation will likely go through three phases:

(1) D "small" - pay cost for factors - $D\Delta$ method used,
(2) D "moderate" - pay cost for factors - dynamic switching between CD and $D\Delta$,
(3) D "large" - stop paying cost for factors and default to CD method.

3.4 Back to the source

Pinch [8] gave an explicit version of an implied algorithm in Duparc [4]. There is a slightly different algorithm that is also implied which we explain here. Recall that the CD method creates Δ values. The values of the product CD lie in $[P^2+1, P^2+L_P]$ and this implies that $\Delta \in [1, L_P]$. Duparc suggests only creating such Δ values on the interval $[1, 2P-4]$. On this interval, we will construct $q > P$. We note that this interval is linear in P and such Carmichael numbers may be found in time $O(P \ln P)$. Having accounted for $q > P$ in this manner, we can now search for $p_{d-3} < q < P$ which can also be done for in time $O(P \ln P)$ by the use of a Sieve of Eratosthenes. The problem now is the cost of finding r given P and q. This is the exact same problem as P being large. We use Theorem 2 and we know that $(Pq)r \equiv 1 \pmod{L}$ and $(r-1)$ divides $Pq-1$. The former can be used to sieve in a residue class and the latter can be used by factoring $Pq-1$. In fact, the size of

the factors imply a bound on how far we have to sieve. One can now balance sieving with respect to L with the cost of finding factors of $Pq-1$. However, we know of no way to give an asymptotic estimate of this cost of finding q and r due to the nature of L. The quantity L can be as small as $p_{d-3}-1$ or nearly as large as P itself.

4 Tabulating all Carmichael numbers $n < B$ when P is "large"

Since both q and r may be found in time that is essentially linear in P, we want P to be large enough with respect to B so that the cost of the sieving/factoring approach is better. Since we will have to sieve for q regardless, this implies that $P > B^{1/3}$ and we sieve for q on $[p_{d-3}, P]$. If $q > P$ and $P > B^{1/3}$, then $Pqr > B$ and the corresponding Carmichael number would exceed the desired bound. As with Pinch's report, the case of $d = 3$ is entirely accounted as "small."

Theorem 6 *All Carmichael numbers less than B with exactly three prime factors may be tabulated by considering $O(B^{2/3})$ CD or DΔ pairs .*

Proof For each prime $p < B^{1/3}$, we use the results in Sect. 3.3 to consider $O(p \ln p)$ candidates for q, r. Since there are $O(B^{1/3}/\ln B)$ primes to consider, we consider the sum

$$\sum_{p<B^{1/3}} O(p \ln p) = O(B^{2/3}).$$

□

In [5], the authors conjecture that the number of Carmichael numbers less than B having exactly three prime factors is asymptotically equivalent to

$$c\frac{B^{1/3}}{(\ln B)^3}$$

and in [6] it was proved that this set is asymptotically bounded above by $B^{7/20+\epsilon}$. All of that is to say that an ideal tabulation algorithm would run in time linear in the output size and we are far from that.

Lower bounds on Δ may also be used to further reduce the number of divisors considered to avoid generating Carmichael numbers that exceed the desired bound. Consider

$$Pq^2 < P\left(\frac{(P-1)(P+D)}{\Delta}\right)^2 < B \Rightarrow \frac{P^{1/2}(P-1)(P+D)}{B^{1/2}} < \Delta.$$

This means that when P gets to be of size $B^{1/5}$ that divisors would be discarded due to being too small.

This means that the "large" case is as follows. We require $P > B^{1/3}$ and composite. This will have q satisfy $p_{d-3} < q < \sqrt{B/P}$. From here, we use Theorem 2 as described above to find r.

5 Implementation and conclusions

We conclude with three points. First, we mention some practical improvements to the DΔ method. Second, we address some issues regarding the primality testing and the approach we used. Here, we mention how we think primality testing ought to be done and what choices we made in order to compare the DΔ method to the *CD* method. Third, we present some timing information in order to show that our improved algorithms are practical. The timing information shows that it is reasonable to use the DΔ method on all

$P < X$ regardless of the number of prime factors in P. We also show that the $D\Delta$ method used in conjunction with the CD method (see Sect. 3.3) is faster than either method individually. This means that our novel approach gives an unconditional improvement for a tabulation of Carmichael numbers.

5.1 Improvements to $D\Delta$ method

For expositional purposes, we described a simple version of the $D\Delta$ method that worked for asymptotic purposes. The desired goal is to minimize the number of divisors that are created for entering the inner loop. There are two ways to accomplish this. First, is to use q is a prime and not merely the fact that it is an integer. The second is to use the fact that C is integral. We do not believe that either of the improvements we describe would change the asymptotic analysis of the algorithm.

Consider divisors of $(P-1)(P+D)/2$ instead of $(P-1)(P+D)$. Both guarantee q is integral but the former guarantees q is odd. It is possible to employ a similar trick to force $q \equiv 1, 2 \pmod 3$. This means that the divisor has to be multiple of three or the divisor has to be $1 \pmod 3$. In the latter case, this means prime divisors of $(P-1)(P+D)$ that are $2 \pmod 3$ have to appear with parity. Both of these can be incorporated into an odometer-style divisor generator to avoid generating unwanted divisors. While it is theoretically possible to scale this for other primes, the overhead seems like it would be worse than just doing the divisibility check. At that point, the "reduction" is equivalent to trial division.

We can further reduce the set of divisors we generate by examining the integrality condition on C. When $D \equiv 0 \pmod 2$, for $(P^2+\Delta)/D$ to be integral, Δ has to be odd. So, for all D even, we require $\nu_2(\Delta) = 0$. This generalizes as follows:

Proposition 1 *Let p be a prime with $p \mid D$, then*

(1) if $\nu_p(P) = 0$ then $\nu_p(\Delta) = 0$,
(2) if $\nu_p(P) = 1$ then $\nu_p(\Delta) \geq 1$.

Proof We require $(P^2+\Delta)/D$ to be an integer. So, for any prime p dividing D, we need $P^2 \equiv -\Delta \pmod p$. The valuations follow. □

At the time of writing, our implementation only considers $D \pmod 2$.

This also explains why most Carmichael numbers occur with D being relatively small. If we view $P^2+\Delta$ as a random residue modulo D, then we expect it to take the value 0 about $1/D$ times. The count of divisors determines how many "chances" we get of this happening.

5.2 Primality testing

The slowest operation in the inner loop will be the primality tests of q and r. In [8], Pinch writes "we note that testing candidates for p_i for primality is required at every state of the calculation. We found precomputing a list of prime numbers up to a suitable limit produced a considerable saving in time." He did not further address how primality was proved for his output. We address that from both a theoretical and practical point of view.

As with any primality proving algorithm, the strategy is generally to use some form of "fast" pseudoprimality test followed by a primality proof. On the former, one usually considers trial division up to some small bound followed by a base-2 strong Fermat pseu-

doprime test. On the latter, we note that in the $D\Delta$ method that $q-1$ is always generated in factored form. Further, $r-1$ is often generated in a partially factored form. These are ideal candidates for the Pocklington-style primality tests. In fact, q can always be tested with this method. If the factored part of $r-1$ exceeds $r^{3/10}$ then the Pocklington-style tests may be used (see Sect. 4.1 of [3] for an explanation).

If not enough of the partial factorization of $r-1$ is given or if q and r were generated with the *CD* method, we could use ECPP to establish primality of the output.

In Sect. 3, we compared the number of times the inner loop was entered for the *CD* method to the $D\Delta$ method. For timing purposes, we wanted the inner-loops to be comparable. That is, we did not want an entirely different primality testing algorithm in the inner loop of one method to compare to a different method. So, the data in the section below corresponds to a run that only does the "pseudoprimality" testing of trial division by small primes. This way, each method pays a comparable cost for entering the inner loop.

5.3 Timing data

Code for both the *CD* method and the $D\Delta$ method were implemented in C++ and run on a small cluster at Illinois Wesleyan University. The cluster has 4 nodes, each with a 12-core, 24-thread, 3.8 GHz processor (with boosting capability up to 4.6 GHz). The code that we used can be found on github.[1]

The following two sets of timings illustrate the improvements discussed in this paper. First, the $D\Delta$ and *CD* methods were applied to all pre-products up to a variety of bounds. The tuples (P, q, r) constructed may include non-Carmichael numbers due to a lack of primality proving, but the timings reflect a fair comparison since the same primality tests are applied in both cases. All timings are given in seconds. The timings found under the "Hybrid" label were obtained by running the $D\Delta$ algorithm unless $P_{d-3} < 20$ and then the algorithm from Sect. 3.3 was used.

Pre-product bound	*CD*	$D\Delta$	Hybrid
$1 \cdot 10^4$	81	21	10
$2 \cdot 10^4$	553	92	50
$3 \cdot 10^4$	1730	231	124
$4 \cdot 10^4$	3778	430	233
$5 \cdot 10^4$	7017	697	395
$6 \cdot 10^4$	11455	983	568
$7 \cdot 10^4$	17281	1425	795
$8 \cdot 10^4$	23806	1898	1072
$9 \cdot 10^4$	33288	2425	1386

Tabulations were also performed on exclusively prime pre-products. This is the best case for the *CD* method, and as one can see it outperforms the $D\Delta$ method under this restriction. A simple hybrid method was also implemented, where the tabulation switches from $D\Delta$ to *CD* when $D > P/(\ln P)^2$. As one can see, this combination of techniques provides an improvement over either individually.

[1] https://github.com/ashallue/tabulate_car.

Prime pre-product bound	$D\Delta$	CD	Hybrid
$1 \cdot 10^4$	9	1	1
$2 \cdot 10^4$	36	6	3
$3 \cdot 10^4$	83	15	8
$4 \cdot 10^4$	151	26	14
$5 \cdot 10^4$	237	41	22
$6 \cdot 10^4$	348	60	31
$7 \cdot 10^4$	470	80	41
$8 \cdot 10^4$	619	103	53
$9 \cdot 10^4$	738	125	64
$10 \cdot 10^4$	939	159	81
$11 \cdot 10^4$	1170	193	97
$12 \cdot 10^4$	1328	221	110

5.4 Future work

We intend to revisit the work here and expand on it in two ways. As with Pinch, we find that the case of $d = 3$ may be entirely accounted as small. For a tabulation of all Carmichael numbers less than B, we now would only have to consider composite P with $P > 3 \cdot 10^6$. We plan on letting the tabulation algorithm run longer and doing a tabulation of all Carmichael numbers $n < 10^{24}$. We would also like to revisit the implied algorithm of Duparc found in Sect. 3.4 and see if it gives improvements.

Data Sharing Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Author details
[1]Illinois Wesleyan University, Bloomington, USA, [2]Butler University, Indianapolis, USA.

Received: 6 September 2022 Accepted: 19 September 2022 Published online: 10 October 2022

References

1. Beeger, N.G.W.H.: On composite numbers n for which $a^{n-1} \equiv (\mathrm{mod}\, n)$ for every a prime to n. Scripta Math. **16**, 133–135 (1950)
2. Carmichael, R.D.: Note on a new number theory function. Bull. Am. Math. Soc. **16**(5), 232–238 (1910)
3. Crandall, R., Pomerance, C.: Prime Numbers. A Computational Perspective, 2nd edn. Springer, New York (2005)
4. Duparc, H.J.A.: On Carmichael numbers. Simon Stevin **29**, 21–24 (1952)
5. Granville, Andrew, Pomerance, Carl: Two contradictory conjectures concerning Carmichael numbers. Math. Comput. **71**(238), 883–908 (2002)
6. Heath-Brown, D.R.: Carmichael numbers with three prime factors. Hardy-Ramanujan J. **30**, 6–12 (2007)
7. Helfgott, H.A.: An improved sieve of Eratosthenes. Math. Comput. **89**(321), 333–350 (2020)
8. Pinch, R.G.E.: The Carmichael numbers up to 10^{15}. Math. Comput. **61**(203), 381–391 (1993)
9. Pinch, R.G.E.: The Carmichael numbers up to 10^{16}, (March, 1998). arXiv:math.NT/9803082
10. Pinch, R.G.E.: The Carmichael numbers up to 10^{17}, April (2005). arXiv:math.NT/0504119
11. Pinch, R.G.E.: The Carmichael numbers up to 10^{18}, (April, 2006). arXiv:math.NT/0604376
12. Pinch, R.G.E.: The Carmichael numbers up to 10^{21}, (May, 2007). `s369624816.websitehome.co.uk/rgep/p82.pdf`
13. Pomerance, C.: Carmichael numbers. Nieuw Arch. Wisk. (4) **11**(3), 199–209 (1993)
14. Sorenson, J.P.: Two compact incremental prime sieves. LMS J. Comput. Math. **18**(1), 675–683 (2015)

Publisher's Note