CALCULATING CYCLOTOMIC POLYNOMIALS

ANDREW ARNOLD AND MICHAEL MONAGAN

ABSTRACT. We present three algorithms to calculate $\Phi_n(z)$, the n_{th} cyclotomic polynomial. The first algorithm calculates $\Phi_n(z)$ by a series of polynomial divisions, which we perform using the fast Fourier transform. The second algorithm calculates $\Phi_n(z)$ as a quotient of products of sparse power series. These two algorithms, described in detail in the paper, were used to calculate cyclotomic polynomials of large height and length. In particular, we have found the least n for which the height of $\Phi_n(z)$ is greater than n, n^2, n^3, and n^4, respectively. The third algorithm, the big prime algorithm, generates the terms of $\Phi_n(z)$ sequentially, in a manner which reduces the memory cost. We use the big prime algorithm to find the minimal known height of cyclotomic polynomials of order five. We include these results as well as other examples of cyclotomic polynomials of unusually large height, and bounds on the coefficient of the term of degree k for all cyclotomic polynomials.

1. Introduction

The n_{th} **cyclotomic polynomial**, $\Phi_n(z)$, is the monic polynomial whose $\phi(n)$ distinct roots are exactly the n_{th} primitive roots of unity.

$$\Phi_n(z) = \prod_{\substack{j=1\\ gcd(j,n)=1}}^{n} \left(z - e^{2\pi ij/n}\right). \tag{1}$$

It is an irreducible polynomial over $\mathbb{Z}$ with degree $\phi(n)$, where $\phi(n)$ is Euler's totient function. The n_{th} **inverse cyclotomic polynomial**, $\Psi_n(z)$, is the polynomial whose roots are the n_{th} non-primitive roots of unity.

$$\Psi_n(z) = \prod_{\substack{j=1\\ gcd(j,n)>1}}^{n} \left(z - e^{2\pi ij/n}\right). \tag{2}$$

As the roots of $\Phi_n(z)$ and $\Psi_n(z)$ comprise all n_{th} roots of unity, we have

$$\Psi_n(z) = \frac{z^n - 1}{\Phi_n(z)}. \tag{3}$$

For more about inverse cyclotomic polynomials, see Moree [19].

Let the **order** of $\Phi_n(z)$ denote the number of distinct odd prime divisors of n. To make the distinction between the order of $\Phi_n(z)$ and n, we will refer to n as the **index** of $\Phi_n(z)$ in this paper.

2000 *Mathematics Subject Classification.* Primary 11Y16, Secondary 12-04

We write

$$\Phi_n(z) = \sum_{k=0}^{\phi(n)} a_n(k) z^k \quad \text{and} \quad \Psi_n(z) = \sum_{k=0}^{n-\phi(n)} c_n(k) z^k$$

and define $a_n(k) = 0$ for $k > \phi(n)$. We let $A(n)$ and $S(n)$ be the **height** and **length**, respectively, of $\Phi_n(z)$. That is,

$$A(n) = \|\Phi_n(z)\|_\infty = \max_{0 \le k \le \phi(n)} |a_n(k)| \quad \text{and} \quad S(n) = \|\Phi_n(z)\|_1 = \sum_{k=0}^{\phi(n)} |a_n(k)|.$$

For $n < 105$, all the coefficients of $\Phi_n(z)$ are $-1, 0$, or 1; however, for $n = 105$ we find that $A(105) = 2$. Paul Erdős [7] proved that $A(n)$ is not bounded above by any polynomial in n. That is, for any constant $c > 0$, there exists n such that $A(n) > n^c$. There is a wealth of material on the behaviour of $A(n)$ and the size of cyclotomic polynomial coefficients [4, 5, 17, 25]; however, computation may yet provide more insight. Koshiba [15, 16] calculated $A(4849845) = 669606$ and found the coefficients of $\Phi_n(z)$ with degree less than $\phi(n)/10$ for $n = 111546435 = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$ and in particular, to our knowledge, the first computed cyclotomic coefficient $a_n(k)$ such that $a_n(k) > n$. T.D. Noe has made a wealth of data about cyclotomic polynomials and their coefficients available at the Sloane On-Line Encyclopedia of Integer Sequences (see [20, 21, 22, 23, 24], for instance).

A cyclotomic polynomial $\Phi_n(z)$ is said to be **flat** if $A(n) = 1$. We say that $\Phi_n(z)$ is flatter than $\Phi_m(z)$ if $A(m) < A(n)$. It is currently unknown whether there are flat cyclotomic polynomials of order five or greater. We find many examples of flat cyclotomic polynomials of orders three and four, and many examples of $\Phi_n(z)$ of order five and height 2.

A related problem of interest is the behaviour of the z^k coefficient $a_n(k)$ of $\Phi_n(z)$ over $n \in \mathbb{N}$. To that end let $a(k) = \max_n |a_n(k)|$. Bachman [1] developed an asymptotic formula for $\log a(k)$ that improved on a result of Montgomery and Vaughan [18]. Noe [20] computed $a(k)$ for $k \le 1000$. We independently verify $a(k)$ for $k \le 172$ by way of a brute-force approach. We also extend results of Bosma [6], who computed the least value of k for which $b \in \mathbb{Z}$ occurs as $a_n(k)$ for some n, for $|b| \le 50$.

Our motivation for developing algorithms to calculate cyclotomic polynomials was to further the study the coefficients of $\Phi_n(z)$. In particular, we were interested in studying $A(n)$. We found, for instance, that $n = 1181895$ is the smallest n for which $A(n) > n$. Until recently, it was impractical to use modern computer algebra systems to calculate $\Phi_n(z)$ for n in the millions. With the development and implementation of the algorithms in this paper, we are able to compute $\Phi_n(z)$ for n in the billions and, for some specific, non-trivial cases, well beyond that.

1.1. Organization of paper.

We present three algorithms to calculate cyclotomic polynomials. Our first algorithm calculates $\Phi_n(z)$ for squarefree n via a series of polynomial divisions. We use the discrete fast Fourier transform to perform these divisions quickly. The second algorithm calculates $\Phi_n(z)$ as a quotient of products of sparse power series. The third algorithm outputs the coefficients of $\Phi_n(z)$ sequentially, in a manner which requires less memory than it takes to store $\Phi_n(z)$, a limitation of the first two algorithms.

Using these three algorithms, we have produced a library of data on the coefficients of cyclotomic polynomials. Amongst our results, we have computed:

[I] $A(n)$ and $S(n)$ for odd, squarefree n amongst the following:
- $n < 10^8$ with three prime factors.
- $n < 3 \cdot 10^8$ with four prime factors.
- $n < 6.5 \cdot 10^8$ with five prime factors.
- $n < 10^9$ with six prime factors.
- $n < 2.36 \cdot 10^9$ with seven prime factors.
- $n < 5.6 \cdot 10^9$ with eight prime factors.
- $n < 1.50 \cdot 10^{10}$ with nine prime factors.

[II] The height and length of $\Psi_n(z)$ for odd, squarefree n amongst the following:
- $n < 10^8$ with three prime factors.
- $n < 10^8$ with four prime factors.
- $n < 5 \cdot 10^8$ with five prime factors.
- $n < 5 \cdot 10^8$ with six prime factors.
- $n < 10^9$ with seven prime factors.
- $n < 2 \cdot 10^9$ with eight prime factors.

[III] The least n for which $A(n) > n, n^2, n^3$, and n^4 respectively, and other values of unusually large height (Tables 6,7).

[IV] The least n such that $A(n) > 2^{64}$ (Table 6).

[V] $\max_n a_n(k)$, $\min_n a_n(k)$, and $a(k) = \max_n |a_n(k)|$ for $k \leq 172$ (partially verifying a result of Noe [20]).

[VI] The least integer k for which, given b, there exists n such that $a_n(k) = b$ for $|b| \leq 927$ (Table 10).

[VII] Many examples of $\Phi_n(z)$ of order 5 with height 2 (Table 9).

We include some of the more noteworthy results in this paper; however, all of the data listed above is available on the web at

`http://www.cecm.sfu.ca/~ada26/cyclotomic/`

1.2. **Preliminaries.** We are interested in computing cyclotomic polynomials of large height. The following two identities are useful:

Lemma 1. *Let $n > 1$ be odd. Then $\Phi_{2n}(z) = \Phi_n(-z)$.*

Lemma 2. *Let p be a prime that divides n. Then $\Phi_{np}(z) = \Phi_n(z^p)$.*

Lemma 1 tells us $A(2n) = A(n)$ and $S(2n) = S(n)$ for odd n. Lemma 2 implies that $A(np) = A(n)$ and $S(np) = S(n)$ for p dividing n. For the remainder of the paper, we will be strictly concerned with the calculation of cyclotomic polynomials of squarefree, odd index. Lemmas 1 and 2 provide an easy means of calculating $\Phi_n(z)$ for n even or nonsquarefree, provided we can calculate $\Phi_m(z)$, where m is the greatest squarefree, odd divisor of n.

2. Calculating $\Phi_n(z)$ as a polynomial quotient via the fast Fourier transform

Our first algorithm calculates $\Phi_n(z)$ by a series of polynomial divisions. Our algorithm uses the following identity, which is easy to verify by showing that both sides have the same roots.

Lemma 3. *Let p be prime that does not divide m. Then* $\Phi_{mp}(z) = \dfrac{\Phi_m(z^p)}{\Phi_m(z)}$.

We thus are able to calculate $\Phi_n(z)$, for n of the form

$$n = p_1 p_2 \cdots p_k = mp,$$

with largest prime divisor $p = p_k$, by repeated polynomial division, as detailed in algorithm 1.

Algorithm 1: Calculating $\Phi_n(z)$ by repeated division

```
Input: $n = p_1 p_2 \cdots p_k$, a product of $k$ distinct primes.
Output: $\Phi_n(z)$, the $n_{th}$ cyclotomic polynomial.
$\eta \longleftarrow 1$, $\Phi_\eta(z) \longleftarrow z - 1$
for $j = 1$ to $k$ do
    $\eta^* \longleftarrow \eta p_j$
    $\Phi_{\eta^*}(z) \longleftarrow \frac{\Phi_\eta(z^{p_j})}{\Phi_\eta(z)}$
    $\eta \longleftarrow \eta^*$
return $\Phi_\eta(z)$
```

This algorithm (see [26]) is well-known. It is used in many computer algebra systems. For example, in Maple 13 and in prior versions, it is used by the `numtheory[cyclotomic]` command. Using classical, quadratic polynomial division, however, is much too slow to do any extensive search on cyclotomic polynomials of index n over one million. For example, Maple 13 takes over five minutes to find $\Phi_{255255}(z)$ (see table 3 for timings). We implemented algorithm 1 with machine-precision arithmetic and optimized the polynomial division in algorithm 1 by way of the discrete fast Fourier transform (FFT)[9, 26]. Our implementation of algorithm 1 can calculate $\Phi_{255255}(z)$ modulo two 32-bit primes in under one second. Algorithm 2 gives a high-level description of the division calculation via the FFT.

Using the fast Fourier transform, we can calculate $\Phi_n(z) = \frac{\Phi_m(z^p)}{\Phi_m(z)}$ modulo a prime q in $\mathcal{O}(N \log(N))$ arithmetic operations, where N is the smallest power of 2 greater than $\phi(n)$, the degree of the quotient. We need not interpolate using $N > \phi(m)p$, the degree of the numerator, as we know division of $\Phi_m(z^p)$ by $\Phi_m(z)$ is exact.

Algorithm 2: Polynomial division via the discrete fast Fourier transform

```
Input:
    • $\Phi_m(z)$, for some odd, squarefree $m$
    • $p$, an odd prime not dividing $m$
    • $N = 2^s$, a power of 2 greater than $\phi(n)$
    • $q$, a prime of the form $q = r \cdot N + 1$
    • $\omega$, a primitive $N_{th}$ root of unity modulo $q$
Output: $\Phi_{mp}(z) \mod q$
Calculate $A_i = \Phi_m(\omega^{ip}) \bmod q$ and $B_i = \Phi_m(\omega^i) \bmod q$ for $i = 0, 1, \ldots, N-1$
using the discrete FFT.
for $i = 0$ to $N - 1$ do $C_i \longleftarrow \frac{A_i}{B_i} \mod q$        //$C_i = \Phi_{mp}(\omega^i)$
Interpolate $C(z) = \Phi_{mp}(z) \mod q$ by the inverse discrete FFT.
```

Observe that algorithm 2 requires that B_i is nonzero modulo q for ω^i, $0 \leq i < N$. This does not pose a problem for odd indices m, however, by the following lemma.

Lemma 4. *Let $m > 1$ be an odd, squarefree integer, and let p, N, q, and ω be as defined in algorithm 2. Then $\Phi_m(\omega^i) \neq 0 \bmod q$ for $0 \leq i < N$.*

Proof. Every power of ω is a $(2^k)_{th}$ root of unity modulo q for some $k \leq s$. For $m > 1$ it holds that $\Phi_m(z)$ divides

$$\frac{z^m - 1}{z - 1} = z^{m-1} + z^{m-1} + \cdots + z + 1.$$

Thus as q does not divide m, any root of $\Phi_m(z) \bmod q$ is necessarily an m_{th} root of unity not equal to 1. Given m is odd for our purposes, the only m_{th} root of unity that is also a $(2^k)_{th}$ root of unity modulo q for some k is exactly 1, and so $\Phi_m(\omega^i) \neq 0 \bmod q$ for $0 \leq i < N$. □

2.1. **Implementation details.** For primes $q < 2^{32}$ of the form $q = rN + 1$, N cannot exceed 2^{27}. Thus for cyclotomic polynomials of large degree, we require primes larger than 2^{32}. Choosing unnecessarily large primes for the FFT would require multiprecision arithmetic. Using 64-bit arithmetic, we are able to multiply modulo prime numbers as large as 42-bits (algorithm 3); however, multiplication modulo a 42-bit prime is roughly twice as slow as multiplication modulo a 32-bit prime.

Algorithm 3: Multiplication modulo a 42-bit prime

```
Input: $a = a_{41}a_{40} \cdots a_1a_0, b = b_{41}b_{40} \cdots b_1b_0$, two 42-bit integers modulo a
       42-bit prime $q$ (i.e. $q < 2^{42}$)
Output: $c = ab \bmod q$
$A_0$ and $B_0$ are obtained via bitmask operations; $A_1$ and $B_1$ are obtained by
bitshifts:

$A_0 \longleftarrow a_{20}a_{19} \cdots a_0$                       // $A_0 = a \mod 2^{21}$
$A_1 \longleftarrow a_{41}a_{40} \cdots a_{21}$                    // $A_1 = (a - A_0)/2^{21}$
$B_0 \longleftarrow b_{20}b_{19} \cdots b_0$                       // $B_0 = b \mod 2^{21}$
$B_1 \longleftarrow b_{41}b_{40} \cdots b_{21}$                    // $B_1 = (b - B_0)/2^{21}$
$c \longleftarrow A_1B_12^{21} \bmod q$,   $c \longleftarrow c + A_1B_0$,   if $c > q$ then $c \longleftarrow c - q$
$c \longleftarrow c + A_0B_1$,   if $c > q$ then $c \longleftarrow c - q$
$c \longleftarrow c \cdot 2^{21} \bmod q$,   $c \longleftarrow c + A_0B_0$,   if $c > q$ then $c \longleftarrow c - q$
```

The FFT only gives us the coefficients of $\Phi_n(z)$ modulo a prime q_1. Our resulting polynomial, call it $H_n(z)$, will not equal $\Phi_n(z)$ if $A(n) > \frac{q_1}{2}$. We calculate $\Phi_n(z)$ modulo another prime q_2. We then reconstruct $H_n(z) \equiv \Phi_n(z) \bmod q_1q_2$ by Chinese remaindering. As such, we call this implementation the **FFT-CRT** algorithm. We do this with primes $q_1, q_2 \ldots q_l$ until $\|H_n(z)\|_\infty \cdot 2^{20} < \frac{q_1q_2 \cdots q_l}{2}$. We then take our solution $H_n(z)$ and use the FFT to test that

$$H_n(\omega^j) \cdot \Phi_m(\omega^j) - \Phi_m(\omega^{jp}) \equiv 0 \bmod q_{l+1} \quad (0 \leq j < N), \tag{4}$$

for some new prime q_{l+1} with N_{th} primitive root ω, where here N is a power of two greater than $\phi(m)p$. This is because $H_n(z)\Phi_m(z) - \Phi_m(z^p)$ is potentially a polynomial of degree $\phi(m)p$. If equation (4) holds for all j, $H_n(z) \equiv \Phi_n(z)$ (mod $Q = q_1 q_2 \cdots q_l \cdot q_{l+1}$). For $q_{l+1} > 2^{40}$, it follows that all of the coefficients of $\Phi_n(z)$, modulo Q, lie in the interval $(\frac{-Q}{2^{60}}, \frac{Q}{2^{60}})$. We consider 60 redundant bits sufficient. As we know the cyclotomic coefficients roughly have a flat-bell distribution, it is very improbable that the height of $\Phi_n(z)$ were greater than $\frac{Q}{2}$ with all the coefficients strictly in the range $(\frac{-Q}{2^{60}}, \frac{Q}{2^{60}})$ modulo Q. Indeed, all our results obtained by this method thus far have been consistent with results we have obtained by non-modular algorithms. Table 1 lists the primes and the primitive roots we used in our computations.

	q	$=$	$r \cdot N + 1$	size of q	ω
q_1	2748779069441	$=$	$5 \cdot 2^{39} + 1$	42 bits	243
q_2	4123168604161	$=$	$15 \cdot 2^{38} + 1$	42 bits	624392905782
q_3	2061584302081	$=$	$15 \cdot 2^{37} + 1$	41 bits	624392905781
q_4	206158430209	$=$	$3 \cdot 2^{36} + 1$	38 bits	10648
q_5	2027224563713	$=$	$59 \cdot 2^{35} + 1$	41 bits	1609812002788

Table 1: Primes and the primitive roots used in our FFT calculations

The brunt of the computation in our implementation of 1 takes place in the last division, as each successive division effectively increases the degree of the resulting intermediate polynomial by another factor. For squarefree n, we can compute $\Phi_n(z)$ mod q_i in $\mathcal{O}(N \cdot \log N)$ operations in $\mathbb{Z}_{q_i}$.

3. Calculating $\Phi_n(z)$ as a quotient of sparse power series

Every n_{th} root of unity is a primitive d_{th} root of unity for some unique $d|n$, and every d_{th} primitive root of unity is an n_{th} root of unity. Thus $z^n - 1 = \prod_{d|n} \Phi_d(z)$. Applying the Mőbius inversion formula to this, we obtain the well-known identity:

$$\Phi_n(z) = \prod_{d|n, d>0} \left(z^d - 1\right)^{\mu\left(\frac{n}{d}\right)}, \tag{5}$$

where μ is the Mőbius function. For $n > 1$, the number of squarefree divisors of n is even, in which case

$$\Phi_n(z) = \prod_{d|n, d>0} \left(1 - z^d\right)^{\mu\left(\frac{n}{d}\right)}. \tag{6}$$

For example, for $n = 105 = 3 \cdot 5 \cdot 7$,

$$\Phi_{105}(z) = \frac{(1 - z^3)(1 - z^5)(1 - z^7)(1 - z^{105})}{(1 - z)(1 - z^{15})(1 - z^{21})(1 - z^{35})}.$$

The sparseness of each term in this quotient lends itself to fast power series arithmetic. For the purposes of our algorithm, we treat $\Phi_n(z)$ as a truncated power series. Multiplying a power series $B(z) = \sum_{i=0}^{\infty} b(i) z^i$ by $1 - z^d$ is easy:

$$\left(\sum_{i=0}^{\infty} b(i) z^i\right)\left(1 - z^d\right) = \sum_{i=0}^{d-1} b(i) z^i + \sum_{i=d}^{\infty} (b(i) - b(i-d)) z^i.$$

To divide by $1 - z^d$ we merely multiply by the power series expansion of $\frac{1}{1-z^d}$:

$$\left(\sum_{i=0}^{\infty} b(i)z^i\right)\left(1 + z^d + z^{2d} + \cdots\right) = \sum_{i=0}^{\infty}\big(b(i) + b(i-d) + b(i-2d) + \cdots\big)z^i. \tag{7}$$

Observe that the terms of $B(z)(1+z^d)$ and $B(z)(1+z^d)^{-1}$ depend strictly on terms of equal or lesser degree in $B(z)$. In addition, for $n > 1$, the coefficients of $\Phi_n(z)$ are **palindromic**. That is, $a_n(k) = a_n(\phi(n) - k)$. So, to calculate the $\Phi_n(z)$ as a power series, we only need compute and store the first $\frac{\phi(n)}{2} + 1$ terms of $\Phi_n(z)$ and of any intermediate power series in our computation.

Division by $(1 - z^d)$ done naively could be quadratic-time, ruining the efficiency of the algorithm. Suppose then that we have the coefficients $b(0)$, $b(1), \ldots, b(D)$ of some power series $B(z)$ up to degree $D = \phi(n)/2$, and we want to calculate $c(0), c(1), \ldots, c(D)$, the first $D + 1$ coefficients of the power series $C(z) = B(z) \cdot (1 - z^d)^{-1} = \sum_{i=0}^{\infty} c(i)$. We can calculate all the $c(i)$ in linear time, without using additional memory to store intermediate results. By (7), $c(i) = b(i)$ for $0 \leq i < d$. For $i > d$, where $i = qd + r$ and $0 \leq r < i$,

$$c(i) = b(i) + b(i-d) + \cdots + b(i-qd).$$

Since $c(i-d) = b(i-d) + b(i-2d) + \cdots + b(i-qd)$, we have that

$$c(i) = c(i-d) + b(i).$$

If we compute $c(i) = c(i-d) + b(i)$ for $i = d, d+1, \ldots, D$, we will have calculated all the $c(i)$ using at most one addition operation per term. Once we compute $c(i)$ we write that value over $b(i)$; we discard $B(z)$ as we compute $C(z)$. Algorithm 4 describes how we calculate $\Phi_n(z)$ using these techniques.

Algorithm 4: Computing $\Phi_n(z)$ as a quotient of sparse power series

The sparse power series (SPS) algorithm

```
Input: $n = p_1 p_2 \cdots p_k$, a product of $k$ distinct primes
Output: $a_n(0), \ldots, a_n(\phi(n)/2)$, the first half of the coefficients of $\Phi_n(z)$
$D \longleftarrow \phi(n)/2$, $a_n(0) \longleftarrow 1$
for $1 \leq i \leq D$ do $a_n(i) \longleftarrow 0$
for $d|n$ such that $d > 0$ do
    if $\mu(\frac{n}{d}) = 1$ then                              // multiply by $(1 - z^d)$
        for $i = D$ down to $d$ by $-1$ do $a_n(i) \longleftarrow a_n(i) - a_n(i-d)$
    else                                                      // divide by $(1 - z^d)$
        for $i = d$ to $D$ do $a_n(i) \longleftarrow a_n(i) + a_n(i-d)$
```

We call the algorithm the sparse power series (or **SPS**) algorithm. The SPS algorithm requires $\mathcal{O}(2^k \cdot \phi(n))$ operations in $\mathbb{Z}$ to calculate $\Phi_n(z)$ of order k.

We have implemented 8, 32, 64, and 128-bit versions of the SPS algorithm. A variant of our 32 and 64-bit versions are now used in Maple 14 and Sage. We do not use the GNU Multi-Precision library for multiprecision arithmetic. We hand-coded our own 64 and 128-bit integer arithmetic that checks for overflow. This can be done without using redundant bits. We also have modular implementations of the SPS algorithm that calculates $\Phi_n(z)$ modulo 8, 16 or 32-bit primes, and reconstructs $\Phi_n(z)$ by Chinese remaindering. This version is particularly useful for calculating cyclotomic polynomials that we cannot completely store in main memory with large precision.

3.1. The sparse power series algorithm for inverse cyclotomic polynomials.

Recall that $\Psi_n(z)$, the n_{th} inverse cyclotomic polynomial, is the monic polynomial whose roots are the n_{th} non-primitive roots of unity. By (3), $\Phi_n(z) \cdot \Psi_n(z) = z^n - 1$. Thus we see from (6) that, for $n > 1$,

$$\Psi_n(z) = \frac{z^n - 1}{\Phi_n(z)} = (z^n - 1) \cdot \prod_{d|n} \big(1 - z^d\big)^{-\mu(\frac{n}{d})} = - \prod_{d|n, d<n} \big(1 - z^d\big)^{-\mu(\frac{n}{d})}.$$

It becomes immediate that we can calculate $\Psi_n(z)$ in a manner ever-similar to the sparse power series algorithm for $\Phi_n(z)$. As with $\Phi_n(z)$, we need only compute the first half of the terms of $\Psi_n(z)$ because the coefficients of $\Psi_n(z)$ for $n > 1$ are **antipalindromic**: given $\Psi_n(z) = c_n(0) + c_n(1)z + \cdots + c_n(n - \phi(n))z^{n-\phi(n)}, n > 1$, it holds that $c_n(k) = -c_n(n - \phi(n) - k)$. For squarefree $n = p_1 p_2 \dots p_k$ the algorithm requires $\mathcal{O}\big(2^k(n - \phi(n))\big) = \mathcal{O}(2^k n)$ operations in $\mathbb{Z}$.

Algorithm 5: Calculating $\Psi_n(z)$ as a quotient of sparse power series

The sparse power series algorithm for $\Psi_n(z)$

```
Input: $n = p_1 p_2 \cdots p_k$, a product of $k$ distinct primes
Output: $c_n(0), \dots, c_n(\lfloor \frac{n-\phi(n)}{2} \rfloor)$, the first half of the coefficients of $\Psi_n(z)$
$D \longleftarrow \lfloor \frac{n-\phi(n)}{2} \rfloor$,      $c_n(0) \longleftarrow -1$
for $1 \le i \le D$ do $c_n(i) \longleftarrow 0$
for $d|n$ such that $0 < d < n$ do
    if $\mu(\frac{n}{d}) = -1$ then                      // multiply by $(1 - z^d)$
        for $i = D$ down to $d$ by $-1$ do $c_n(i) \longleftarrow c_n(i) - c_n(i-d)$
    else                                               // divide by $(1 - z^d)$
        for $i = d$ to $D$ do $c_n(i) \longleftarrow c_n(i) + c_n(i-d)$
```

4. A low-memory algorithm to calculate $A(n)$

Calculating cyclotomic polynomials of very large degree using algorithm 4 can bode problematic, as oftentimes $\Phi_n(z)$ will not fit in main memory. In such a case, there are a variety of approaches to calculate $\Phi_n(z)$.

One approach is to calculate $\Phi_n(z)$ modulo primes p_i sufficiently small that we can fit $\Phi_n(z)$ in memory and write the images to hard disk. We then reconstruct the coefficients of $\Phi_n(z)$ sequentially from the images of $\Phi_n(z)$ mod p_i. This minimizes the amount of computation we have to do on the hard disk.

For yet larger cyclotomic polynomials, we may not even be able to store the coefficients modulo a prime in memory. In which case we may be forced to write $\Phi_n(z)$ and our intermediate work to disk. This proves most costly, as the speed of the hard disk bottlenecks the sparse power series algorithm.
As a motivating example, consider $\Phi_n(z)$ for

$$n = 2576062979535 = 3 \cdot 5 \cdot 29 \cdot 2609 \cdot 2269829.$$

This is smallest $n = p_1 p_2 p_3 p_4 p_5$, a product of five distinct odd primes, such that

$$p_k \equiv -1 \mod \prod_{i=1}^{k-1} p_i \ \text{ for } k = 2, 3, 4, 5. \tag{8}$$

Nathan Kaplan [12] asked whether this cyclotomic polynomial is flat. To our knowledge, no one has yet found a flat cyclotomic polynomial of order 5. This was a natural candidate to test for flatness. Kaplan [14] proved for $n = p_1p_2p_3$ satisfying (8) for $k = 2, 3$ that $A(n) = 1$. In addition, for every odd $n < 3 \cdot 10^8$ of the form $n = p_1p_2p_3p_4$ satisfying (8) for $k = 2, 3, 4$ is flat.

For our purposes, it is not necessary that we retrieve all the coefficients of $\Phi_n(z)$ at once, as we are mostly concerned with the height of $\Phi_n(z)$. Indeed, for a cyclotomic polynomial with degree in the tens of billions or beyond, there is very little we can feasibly do with $\Phi_n(z)$, so there may be no purpose to store it in memory for further computation.

Let $n = mp$ be a squarefree, odd integer with largest prime divisor p. We can compute $A(n)$ by inspecting some of the coefficients of $\Phi_n(z)$ sequentially such that we only have to store m coefficients of $\Phi_n(z)$ at any one time. This algorithm takes $\mathcal{O}(m^2) = \mathcal{O}(\frac{n^2}{p^2})$ integer operations, provided we have $\Phi_m(z)$ and $\Psi_m(z)$. Clearly, such an algorithm works best for n with a large prime divisor, hence we affectionally call it the **big prime algorithm**.

Recall from (3) that $\Psi_n(z) = (z^n - 1)/\Phi_n(z)$. By lemma 3,

$$\begin{aligned}\Phi_n(z) = \Phi_{mp}(z) = \frac{\Phi_m(z^p)}{\Phi_m(z)} &= \Phi_m(z^p) \cdot \Psi_m(z) \cdot (z^m - 1)^{-1}, \\ &= \Phi_m(z) \cdot \Psi_m(z) \cdot (-1 - z^m - z^{2m} - \cdots).\end{aligned} \tag{9}$$

Write $\Phi_m(z) = \sum_{i=0}^{\phi(m)} b_i z^i$ and $\Psi_m(z) = \sum_{j=0}^{m-\phi(m)} c_j z^j$. From equation (9), we can express coefficients of $\Phi_n(z) = \sum_{k=0}^{\phi(n)} a_n(k) z^k$ in terms of the b_i and c_j:

$$a_n(k) = -\sum_{\substack{ip+j \equiv k \bmod m \\ ip+j \leq k}} b_i c_j.$$

This leads to the recurrence

$$a_n(k) = a_n(k-m) - \sum_{ip+j=k} b_i c_j, \tag{10}$$

which is key to our algorithm. We formally describe the big prime algorithm hereafter:

```
Algorithm 6: A low-memory algorithm to obtain the height of $\Phi_n(z)$
The big prime algorithm for calculating $A(n)$
Input: $n = mp$, a squarefree odd integer with largest prime divisor $p$
$b_0, b_1, \dots, b_{\phi(m)/2}$, the first half of the coefficients of $\Phi_m(z)$,
$c_0, c_1, \dots, c_{m-\phi(m)}$, the coefficients of $\Psi_m(z)$
Output: $H$, the height of $\Phi_n(z)$
$\bar{a}(0), \bar{a}(1) \dots, \bar{a}(m-1) \leftarrow 0, 0, \dots, 0$
for $i = 0$ to $\lfloor \frac{\phi(n)}{2p} \rfloor$ do
    for $j = 0$ to $m - \phi(m)$ do
        $k \leftarrow ip + j \bmod m$
        $\bar{a}(k) \leftarrow \bar{a}(k) - b_i c_j$,
        if $j < p$ and $|\bar{a}(k)| > H$ then $H \leftarrow |\bar{a}(k)|$
return $H$
```

The algorithm iterates through pairs (i,j) such that $0 \leq j \leq m - \phi(m)$, and $ip \leq \phi(n)/2$. Since there are only $\mathcal{O}(m \cdot \phi(m))$ such pairs, it follows that the big prime algorithm takes $\mathcal{O}(m{\cdot}\phi(m)) \in \mathcal{O}(m^2)$ arithmetic operations. We only require the first half of the terms of $\Phi_m(z)$, as $\frac{\phi(m)}{2} > \frac{\phi(m)(p-1)}{2p} = \frac{\phi(n)}{2p}$.

We store the value of $a_n(k)$ in $\bar{a}(k \bmod m)$, and discard that value when computing $a_n(k+m)$. If $p > m - \phi(m)$, the degree of $\Psi_m(z)$, then algorithm 6 does not consider every term of $\Phi_n(z)$ with degree less than $\phi(n)/2$. In particular, if there is no pair (i,j) such that $0 \leq i \leq \phi(m)$, $0 \leq j \leq m - \phi(n)$, and $ip + j = k$, then the big prime algorithm will not consider the term of the degree k. It follows from (10) that for such $k \geq m$, $a_n(k) = a_n(k-m)$, and for such $k < m$, $a_n(k) = 0$. Thus we need not consider these terms to obtain the height $A(n)$.

It is easy to modify algorithm 6 to generate all (or half) of the terms of $\Phi_n(z)$ (see algorithm 7). In such case the number of comparisons and arithmetic operations in $\mathbb{Z}$ increases from $\mathcal{O}(m{\cdot}\phi(m))$ to $\mathcal{O}(m{\cdot}\phi(m){+}\phi(n))$. The computation space remains effectively the same; however, there is the additional $\mathcal{O}(\phi(n))$ space required to store the coefficients $a_n(k)$, whether to memory or disk.

Algorithm 7: Computing $\Phi_{mp}(z)$ from $\Phi_m(z)$ and $\Psi_m(z)$

The big prime algorithm for generating $\Phi_n(z)$

```
Input: $n = mp$, a squarefree odd integer with largest prime divisor $p$,
$b_0, b_1, \dots, b_{\phi(m)/2} \leftarrow$ the first half of the coefficients of $\Phi_m(z)$,
$c_0, c_1, \dots, c_{m-\phi(m)} \leftarrow$ the coefficients of $\Psi_m(z)$
Output: $a_n(0), \dots, a_n(\phi(n)/2)$, the first half of the coefficients of $\Phi_n(z)$
$\bar{a}(0), \bar{a}(1) \dots, \bar{a}(m-1) \leftarrow 0, 0, \dots, 0$
for $i = 0$ to $\lfloor \frac{\phi(n)}{2p} \rfloor$ do
    for $j = 0$ to $m - \phi(m)$ do
        $k \longleftarrow ip + j \bmod m$
        $\bar{a}(k) \leftarrow \bar{a}(k) - b_i c_j$
    for $k = 0$ to $p - 1$ do  $a_n(k) \leftarrow \bar{a}(k \bmod m)$
return $a_n(0), \dots, a_n(\phi(n)/2)$
```

4.1. **The big prime algorithm for $\Psi_n(z)$.** Provided we have $\Phi_m(z)$ and $\Psi_m(z)$, we can generate the terms of $\Psi_n(z)$ for $n = mp$, in $\mathcal{O}(m{\cdot}\phi(m))$ arithmetic operations in $\mathbb{Z}$. To do this we calculate $\Psi_n(z)$ as a product of two polynomials. One can show that

$$\Psi_{mp}(z) = \Phi_m(z)\Psi_m(z^p) \tag{11}$$

by showing that both sides of (11) have the same roots. Thus if we again let $\Phi_m(z) = \sum_{i=0}^{\phi(m)} b_i z^i$ and $\Psi_m(z) = \sum_{j=0}^{m-\phi(m)} c_j z^j$, it is immediate that

$$\Psi_n(z) = \sum_{i+pj=k} b_i c_j z^k,$$

where the sum is taken over $0 \leq i \leq \phi(m)$ and $0 \leq j \leq m - \phi(m)$. If $p > \phi(m)$, the implementation is especially simple, as we have at most one solution to $i + jp = k$ for a given k.

4.2. **Implementation details and observations.** The big prime algorithm and its variants were developed to calculate $A(n)$ for large squarefree n with a large prime divisor. The n for which we first implemented this algorithm were of the form $n = p_1p_2p_3p_4p_5$, a product of five distinct primes, such that $p_k > \prod_{i=0}^{k-1} p_i$ for $k = 2, 3, 4, 5$.

For such cases, to calculate $A(n)$ it is often advantageous to use a sparse representation for $\Phi_m(z)$ and $\Psi_m(z)$. For example, for $n = mp$ where $n = 2576062979535$, $m = 1134915$ and $p = 2269829$, $\Phi_m(z)$ has degree 584192 but only 31679 terms, and $\Psi_m(z)$ has degree 550723 but only 2982 terms.

For yet larger examples of n, we cannot fit an array of m integers in main memory. For example, we wanted to see if $\Phi_n(z)$ was flat, where

$$n = 2876941641794034669918155 = 5 \cdot 29 \cdot 2029 \cdot 2353639 \cdot 4154714171969.$$

This is the smallest $n = p_1p_2p_3p_4p_5$ with $p_1 = 5$ that satisfies the set of congruences in (8). To test if $A(n) = 1$ we first used the sparse power series algorithm to calculate $\Phi_{p_1p_2p_3}(z)$ and $\Psi_{p_1p_2p_3}(z)$, we then used the big prime algorithm algorithm to generate sparse representations of $\Psi_m(z)$ and $\Phi_m(z)$, where $m = p_1p_2p_3p_4$. We then calculated the terms of $\Phi_n(z)$ whose degress were in a range modulo m sufficiently small that we could fit in memory. We found that $\Phi_n(z)$ is not flat, as $|a_n(k)| = 2$ for

$$k = 266298073621 \cdot 4154714171969 + 109596 = 1843987300735798525434 91,$$

at which point we stopped the calculation.

5. A comparison of the efficiency of the algorithms

Table 2 gives a comparison of the three algorithms to calculate $\Phi_n(z)$. For the purposes of this section, let n be of the form

$$n = p_1p_2 \cdots p_k = mp,$$

a product of k distinct primes with largest prime divisor p.

Table 2: Time and space complexities of the algorithms for $\Phi_n(z)$.

Algorithm	# of operations in $\mathbb{Z}$	space complexity
FFT-CRT (mod 1 prime, section 2)	$\mathcal{O}(n \log n)$	$\mathcal{O}(\phi(n))$
Sparse power series (algorithm 4)	$\mathcal{O}(2^k \cdot \phi(n))$	$\mathcal{O}(\phi(n))$
Big prime (for $A(n)$ only, algorithm 6)	$\mathcal{O}(m \cdot \phi(m))$*	$\mathcal{O}(m)$

* Excludes cost of computing $\Phi_m(z)$ and $\Psi_m(z)$.

The time and space complexities alone do not completely illuminate the advantages and disadvantages of each algorithm. Our implementation of the SPS algorithm (alg. 4) has several advantages over the FFT-CRT method (alg. 1). First, we can perform the calculations with little memory overhead; effectively all the memory used in the power series algorithm is to store the coefficients. The power series algorithm makes better use of the memory used to store terms. Using 64-bits of storage for one term gives us exactly 64-bit precision using the SPS method, whereas the FFT-CRT algorithm uses 64-bits to store a 42-bit terms. Most significantly, the arithmetic operations used in algorithm 4 are strictly additions and subtractions, which take fewer CPU cycles than multiplication and division operations. In addition, the SPS algorithm exhibits better locality than the discrete fast Fourier transform. In practice, our implementation of the SPS algorithm is appreciably faster than that of the FFT-CRT algorithm (see table 3).

Table 3: A comparison of running times of cyclotomic polynomial algorithms on a 3 GHz Intel Xeon system

Algorithm	Running time to compute $\Phi_n(z)$(seconds)		
	$n = 255255$	$n = 1181895$	$n = 43730115$
Maple 13 `cyclotomic` command	429.14	-	-
Maple using an array of machine precision integers	13.78	197.37	-
FFT-CRT using 2 42-bit primes, plus a check prime (section 2)	1.12	4.52	371.93
FFT-CRT using 2 32-bit primes, plus a check prime	0.47	2.07	163.68
SPS, 64-bit (algorithm 4)	0.01	0.08	6.04
Big prime, 64-bit (algorithm 6)*	0.06	0.50	727.44

*Includes time to compute $\Phi_m(z)$ and $\Psi_m(z)$ via the 64-bit sparse power series algorithm

Table 4: Running time for the big prime algorithm (calculating height only, 8-bit version) on a 3 GHz Intel Xeon system

n	factorization of n	user time (seconds)
2576062979535	$3 \cdot 5 \cdot 29 \cdot 2609 \cdot 2269829$	1.23
36654908721735	$3 \cdot 5 \cdot 29 \cdot 6959 \cdot 12108659$	3.29
117714212390685	$3 \cdot 5 \cdot 59 \cdot 3539 \cdot 37584179$	3.44
1349266102959585	$3 \cdot 5 \cdot 59 \cdot 8849 \cdot 172290029$	8.61
16628239064490285	$3 \cdot 5 \cdot 179 \cdot 10739 \cdot 576684299$	31.90

We see that the big prime algorithm is relatively fast for $n = 255255$ and 1181895 but considerably slower for $n = 43730115$. This is what we expect, as the algorithm is quadratic-time. These examples, however, are not what the big prime algorithm was designed for. Table 4 shows timings used to calculate heights of $\Phi_n(z)$ for n of the form discussed in section 4.2. With the big prime algorithm, we can calculate heights of cyclotomic polynomials that may otherwise be infeasible to compute.

6. Results

All of our numerical results can be found in the appendix. All of the data we have computed is made available at

`http://www.cecm.sfu.ca/~ada26/cyclotomic/`

6.1. **Big heights and lengths of cyclotomic polynomials.** To find cyclotomic polynomials with large heights, we needed bounds on $A(n)$. Bang [3] showed that for $n = pq$, a product of two primes, that $A(n) = 1$; and for $n = pqr$, a product of three primes that $A(n) < p$. Bloom [5] later proved for $n = pqrs$, a product of four primes with $p < q < r < s$, that $A(n) < p(p-1)(pq-1)$. Bateman, Pomerance, and Vaughan [4] proved a generalized albeit slightly weaker result: for $n = p_1p_2\cdots p_k$, a product of k distinct primes with $p_1 < p_2 < \cdots < p_k$,

$$A(n) \leq A(p_1p_2\cdots p_{k-1})\prod_{j=0}^{k-2} S(p_1p_2\cdots p_j) \tag{12}$$

Using $S(p_1p_2\cdots p_j) \leq A(p_1p_2\cdots p_j)\cdot p_1p_2\cdots p_j$ and Bang's results, they inductively obtain that

$$A(p_1p_2\cdots p_k) \leq \prod_{i=1}^{k-2} p_i^{2^{k-i-1}-1} \tag{13}$$

For example, $A(p_1p_2p_3p_4p_5p_6) \leq p_1^{15}p_2^7p_3^3p_4^1$. We use this bound to narrow our search for large values of $A(n)$, more specifically, $A(n)$ for which $A(n) > A(m)$ for $n < m$. Consider, for instance, $n = p_1p_2p_3p_4p_5$, a product of five primes where $p_1 < p_2 < p_3 < p_4 < p_5$. We have that $A(n) < p_1^7p_2^3p_3 < n^{2.2}$. Given that $\max_{1\leq k\leq n} A(k)$ exceeds $n^{2.2}$ for $n > 43730115$, we can ignore products of five primes greater than 43730115 when searching for increasingly large heights.

Using the algorithms of sections 2 and 3, we have created a library of data on $A(n)$ and $S(n)$. We include here our more noteworthy results. Table 6 (page 18) shows those cyclotomic polynomials we have found whose height is greater than all those of smaller index. Excluding n less than roughly 10000, those n for which we obtain the largest heights also typically yield the largest lengths. Table 6 also gives $\log_n(A(n))$, which was of interest to us. Our results include the smallest n such that $A(n) > n$, $A(n) > n^2, A(n) > n^3$, and $A(n) > n^4$. Table 5 (page 17) shows $A(n)$ for n, a product of the s smallest odd primes, for $1 \leq s \leq 9$.

Equation (12) suggests that if $A(n)$ is large, then $A(np)$ is potentially large as well. This appears the case in table 7 (page 19), which shows computed values of $A(n)$ for which $A(n) > n^4$. Every n in table 7 is divisible by $m = 40755 = 3\cdot5\cdot11\cdot13\cdot19$, which appears in table 6 and has at least one of 29 or 37 as a prime factor. $40755\cdot29 = 1181895$ also appears in table 6. In addition, every n in table 7 is neither divisible by 17, 23, nor (with the exception of $n = 13162764615$) 7.

Many of the examples in table 7 (page 19) are multiples of $m = 43730115$, the first case such that $A(m) > m^2$. We mark these n in table 7 accordingly.

6.2. Computing $\Phi_n(z)$ of higher order.

We currently cannot calculate $\Phi_n(z)$ of order greater than 9. Calculating $\Phi_n(z)$ where $n = 3\cdot5\cdot7\cdot11\cdot13\cdot17\cdot19\cdot23\cdot29\cdot31$, the product of the smallest ten odd primes, would require over 120 GB of memory to compute the polynomial using the SPS algorithm and 64-bit integers, and yet more memory using the FFT-CRT algorithm.

We attempted to compute the cyclotomic polynomials for

$$n = 99660932085 = 3\cdot5\cdot11\cdot13\cdot19\cdot29\cdot37\cdot43\cdot53,$$

which we expect to have a very large height. We computed $\Phi_n(z)$ modulo 32-bit primes by algorithm 4. $\Phi_n(z)$ has degree 38041436160. Storing half the coefficients of $\Phi_n(z)$ with 32-bit precision requires roughly 76 GB, and so the computation had to be done directly to disk. As such, the computation was particularly slow. We computed five images of $\Phi_n(z)$, after which the hard disk crashed. We will not be redoubling our efforts using this approach.

6.3. **Flat cyclotomic polynomials.** If p is a prime, then $\Phi_p(z) = 1+z+\cdots+z^{p-1}$ is trivially flat. All cyclotomic polynomials of order 2 are also flat. This is easy to verify using the following identity (see Lenstra, [11]): Given primes p, q, let u, v be the integers such that $0 < u < p$, $0 < v < q$, and $uq + vp = pq + 1$. Then

$$\Phi_{pq}(z) = \sum_{i=0}^{u-1}\sum_{j=0}^{v-1} z^{iq+jp} - \sum_{i=u}^{p-1}\sum_{j=v}^{q-1} z^{iq-jp-pq}.$$

One can check that the degrees of the terms in each sum are distinct. Cyclotomic polynomials of order three and greater are not, in general, flat.

6.3.1. *Flat cyclotomic polynomials of order 3.* There are 1566382 $n < 10^8$ of the form $n = pqr$, a product of three distinct odd primes, such that $A(n) = 1$. Bachman [2] proved that $A(pqr) = 1$ if $q \equiv -1 \bmod p$ and $r \equiv -1 \bmod pq$. Kaplan [14] proved a more general result, that $A(pqr) = 1$ when $r \equiv \pm 1 \bmod pq$. For $n = pqr < 10^8$, we find that $A(n) = 1$ if $q \equiv 1 \bmod p$ and $r \equiv \pm 2 \bmod pq$. The aforementioned families account for 414832 of these flat cyclotomic polynomials of order 3.

6.3.2. *Flat cyclotomic polynomials of order 4.* Noe [22] has calculated flat cyclotomic polynomials of order 4, for index $n < 5 \cdot 10^6$. We extend his result to $n < 3 \cdot 10^8$. There are 1389 such n, and each of these $n = p_1p_2p_3p_4$ satisfies

$$p_2 \equiv -1 \bmod p_1, \qquad p_3 \equiv \pm 1 \bmod p_1p_2, \qquad p_4 \equiv \pm 1 \bmod p_1p_2p_3. \tag{14}$$

In addition, any cyclotomic polynomial $\Phi_n(z)$ of order four with $n = p_1p_2p_3p_4 < 3 \cdot 10^8$ satisfying (14) is flat.

6.3.3. *Are there flat cyclotomic polynomials of order 5 or greater?* For $n < 6.5 \cdot 10^8$, there is no cyclotomic polynomial $\Phi_n(z)$ of order 5 with height less than 4. Table 8 shows the cyclotomic polynomials of order 5 and index n that are flatter than any cyclotomic polynomial of order 5 and smaller index, for $n < 6.5 \cdot 10^8$. In an attempt to find a flat cyclotomic polynomial of order 5, we computed $A(n)$ for n, a product of 5 distinct primes such that for p dividing n, n/p satisfies the set of congruences (14). That is, we computed $A(n)$ for $n = p_1p_2p_3p_4p_5$ satisfying

$$\begin{aligned} p_2 &\equiv -1 \bmod p_1, & p_3 &\equiv -1 \bmod p_1p_2, \\ p_4 &\equiv \pm 1 \bmod p_1p_2p_3, & p_5 &\equiv \pm 1 \bmod p_1p_2p_3p_4. \end{aligned} \tag{15}$$

We only consider n satisfying (15) for which, given (p_1, p_2, p_3, p_4), p_5 is minimal for its congruence class modulo $p_1p_2p_3p_4$, because of the following theorem from Kaplan:

Theorem 5 (Kaplan, [13]). *Let $m > 0$ and let p, q be primes such that $m < p < q$ and $p \equiv q \pmod m$. Then $A(mp) = A(mq)$.*

We have calculated $A(n)$ for all such $n < 2^{63}$. There are 5349 such n. Of these, $A(n) = 2$ for 5212 cases and $A(n) = 3$ for the remaining ones.

We list the smallest indices n of this form for which we have computed $A(n) = 2$ in table 9. The data for all 5349 cases can be found at our website.

6.4. **Extrema of the k_{th} cyclotomic polynomial coefficient $a_n(k)$.** Let $a(k) = \max_n |a_n(k)|$, and let $a^*(k) = \max_n a_n(k)$ and $a_*(k) = \min_n a_n(k)$ be the one-sided bounds. For notational convenience, we also define

$$a^{**}(k) = \max_{\text{squarefree } n} a_n(k) \qquad \text{and} \qquad a_{**}(k) = \min_{\text{squarefree } n} a_n(k).$$

It is clear that $a^{**}(k) \leq a^*(k)$ and $a_{**}(k) \geq a_*(k)$.

Bachman [1] showed that for a constant A_0, and for sufficiently large k,

$$\log a(k) = A_0 \frac{\sqrt{k}}{(\log k)^{1/4}} \left(1 + \mathcal{O}\left(\frac{\log\log k}{\sqrt{\log k}}\right)\right).$$

Gallot et al. [8] computed $a(k)$ for $k \leq 30$. We calculated $a(k)$ for $k \leq 172$ using a brute-force approach we detail below. Noe [21] calculated $a(k)$, for $k \leq 1000$ using a brute-force approach for $k \leq 128$, and a superior, fast method due to Grytczuk and Tropak [10] for larger k.

We verify his computation up to $k \leq 172$, and for those k we find the smallest index n for which we obtain $|a_n(k)| = a(k)$. It is immediate from algorithm 4 that $a_n(k)$ depends on the divisors of n that are less than or equal to k. In particular, if p and q are distinct primes that are greater than k, then $a_n(k) = a_{npq}(k)$ and $a_{np}(k) = a_{nq}(k)$. Thus to calculate $a^{**}(k)$, we need only consider $a_n(k)$ for n of the form $n = m$ and $n = mq$, where m is a product of distinct primes less than or equal to k, and q is first prime greater than k.

We used this brute-force approach to calculate $a^{**}(k)$ and $a_{**}(k)$ for $0 \leq k \leq 172$. This entailed inspection of $\Phi_n(z)$ for every squarefree n that is a product of primes less than or equal to 173, the 40_{th} prime. There are $2^{40} > 10^{12}$ such n. We used a variant of algorithm 4 to obtain the first 211 terms of $\Phi_n(z)$; instead of truncating the power series of $\Phi_n(z)$ to degree $\phi(n)/2$, we truncate the power series to degree 210. In addition, given odd n, we use lemma 1 to obtain the truncated power series of $\Phi_{2n}(z)$. Since $\Phi_{2n}(z) = \Phi_n(-z)$ for odd $n > 1$, it follows that

$$a_{2n}(2k) = a_n(2k), \qquad \text{and} \qquad a_{2n}(2k+1) = -a_n(2k+1).$$

Given $a_n^{**}(d)$, for $0 \leq d \leq k$, one can obtain $a^*(k)$ by inspection. Suppose that $a^*(k) > a^{**}(k)$, then there exists some non-squarefree n for which $a_n(k) > a^{**}(k)$. Write $n = md$, where m is the squarefree part of n. By lemma 2, $\Phi_n(z) = \Phi_m(z^d)$, and so if $d|k$, $a_n(k) = a_m(k/d)$, otherwise $a_n(k) = 0$. Thus $a^*(k) = \max_{d|k} a^{**}(k/d)$. Similarly, $a_*(k) = \min_{d|k} a_{**}(k/d)$. Typically we find that $a^*(k) = a^{**}(k)$. $k = 118$ is the least $k > 0$ such that $a(k) > k$, as $a^*(118) = 124$.

Another related problem is, given $b \in \mathbb{Z}$, find minimal k such that there exists n such that $a_n(k) = b$. Define

$$\alpha(b) = \min_{a_n(k)=b} k \qquad (\text{for } b \in \mathbb{Z}) \qquad \text{and}$$
$$\bar{\alpha}(b) = \min_{|a_n(k)|=b} k = \min(\alpha(b), \alpha(-b)) \qquad (\text{for } b \geq 0),$$

where the minima are taken over all pairs (n, k) such that $n > 0, k \geq 0$.

In our computation of $a(k)$ we have simultaneously computed $\alpha(b)$, for $-927 \leq b \leq 927$, and the smallest n for which $a_n(\alpha(b)) = b$. Again by lemma 2, we need only consider squarefree n to computer $\alpha(b)$. Suppose $b \neq 0$ and $a_{np^2}(k) = b$. Then, as $a_{np^2}(k) \neq 0$, p must divide k by lemma 2, and $a_{np}(k/p) = b$. Thus $\alpha(b) \leq k/p < k$.

Given that we know the maxima and minima of $a_n(k)$ for fixed $k \leq 172$, if the minimum k we have found for which $\exists n \ni a_n(k) = b$ is less than or equal to 172, then we know we have the exact value of $\alpha(b)$. The same holds if the minimum such k equals 173; however, we cannot be certain that we have the smallest n for which $a_n(173) = b$.

Table 10 shows $\bar{\alpha}(b)$ and least n such that $|a_n(\bar{\alpha}(b))| = b$ for select values of b. We extend results by Bosma [6], and by Grytczuk and Tropak [10]. Grytczuk and Tropak found results for $|b| \leq 10$. Bosma calculated results for $|b| \leq 50$. We have in fact calculated the smallest k such that $a_n(k) = b$ for $-927 \leq b \leq 927$. All of these results can be found at our website.

References

[1] G. Bachman. On the coefficients of cyclotomic polynomials. *Mem. Amer. Math. Soc.*, 106(510):vi+80, 1993.

[2] G. Bachman. Flat cyclotomic polynomials of order three. *Bull. London Math. Soc.*, 38(1):53–60, 2006.

[3] A. S. Bang. Om ligningen $\phi_n(x) = 0$. *Nyt Tidsskrift for Mathematik*, (6):6–12, 1895.

[4] P.T. Bateman, C. Pomerance, and R.C. Vaughan. On the size of the coefficients of the cyclotomic polynomial. In *Topics in classical number theory, Vol. I, II (Budapest, 1981)*, volume 34 of *Colloq. Math. Soc. János Bolyai*, pages 171–202. North-Holland, Amsterdam, 1984.

[5] D.M. Bloom. On the coefficients of the cyclotomic polynomials. *Amer. Math. Monthly*, 75:372–377, 1968.

[6] W. Bosma. Computation of cyclotomic polynomials with Magma. In *Computational algebra and number theory (Sydney, 1992)*, volume 325 of *Math. Appl.*, pages 213–225. Kluwer Acad. Publ., Dordrecht, 1995.

[7] Paul Erdös. On the coefficients of the cyclotomic polynomial. *Bull. Amer. Math. Soc.*, 52:179–184, 1946.

[8] Y. Gallot, P. Moree, and H. Hommerson. Value distribution of cyclotomic polynomial coefficients. Available at `http://arxiv.org/abs/0803.2483`.

[9] K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for Computer Algebra.* Kluwer Academic Publishers, Boston, 1992.

[10] A. Grytczuk and B. Tropak. A numerical method for the determination of the cyclotomic polynomial coefficients. In *Computational number theory (Debrecen, 1989)*, pages 15–19. de Gruyter, Berlin, 1991.

[11] Jr H.W. Lenstra. Vanishing sums of roots of unity. In *Proceedings, Bicentennial Congress Wiskundig Genootschap (Vrije Univ., Amsterdam, 1978), Part II*, volume 101 of *Math. Centre Tracts*, pages 249–268, Amsterdam, 1979. Math. Centrum.

[12] N. Kaplan. Personal correspondence.

[13] N. Kaplan. Cyclotomic polynomials of order four or higher. To appear in *Integers*.

[14] N. Kaplan. Flat cyclotomic polynomials of order three. *J. Number Theory*, 127(1):118–126, 2007.

[15] Y. Koshiba. On the calculations of the coefficients of the cyclotomic polynomials. *Rep. Fac. Sci. Kagoshima Univ.*, (31):31–44, 1998.

[16] Y. Koshiba. On the calculations of the coefficients of the cyclotomic polynomials. II. *Rep. Fac. Sci. Kagoshima Univ.*, (33):55–59, 2000.

[17] H. Maier. The size of the coefficients of cyclotomic polynomials. In *Analytic number theory, Vol. 2 (Allerton Park, IL, 1995)*, volume 139 of *Progr. Math.*, pages 633–639. Birkhäuser Boston, Boston, MA, 1996.

[18] H.L. Montgomery and R. C. Vaughan. The order of magnitude of the mth coefficients of cyclotomic polynomials. *Glasgow Math. J.*, 27:143–159, 1985.

[19] P. Moree. Inverse cyclotomic polynomials. *J. Number Theory*, 129(3):667–680, 2009.

[20] T.D. Noe. Least k such that the x^n coefficient of cyclotomic polynomial phi(k,x) has the largest possible magnitude. Sequence A138475 in N. J. A. Sloane (Ed.), The On-Line Encyclopedia of Integer Sequences (2008), `http://www.research.att.com/~njas/sequences/A138475`.

[21] T.D. Noe. Maximum possible magnitude of the x^n coefficient of a cyclotomic polynomial. Sequence A138474 in N. J. A. Sloane (Ed.), The On-Line Encyclopedia of Integer Sequences (2008), published electronically at `http://www.research.att.com/~njas/sequences/A138474`.

[22] T.D. Noe. Numbers n such that phi(n,x) is a flat cyclotomic polynomial of order four. Sequence A117318 in N. J. A. Sloane (Ed.), The On-Line Encyclopedia of Integer Sequences (2008), `http://www.research.att.com/~njas/sequences/A117318`.

[23] T.D. Noe. Numbers n such that phi(n,x) is a flat cyclotomic polynomial of order three. Sequence A117223 in N. J. A. Sloane (Ed.), The On-Line Encyclopedia of Integer Sequences (2008), `http://www.research.att.com/~njas/sequences/A117223`.

[24] T.D. Noe. Odd squarefree numbers n such that the cyclotomic polynomial phi(n,x) is not coefficient convex. Sequence A146960 in N. J. A. Sloane (Ed.), The On-Line Encyclopedia of Integer Sequences (2008), `http://www.research.att.com/~njas/sequences/A146960`.

[25] R. Thangadurai. On the coefficients of cyclotomic polynomials. In *Cyclotomic fields and related topics (Pune, 1999)*, pages 311–322. Bhaskaracharya Pratishthana, Pune, 2000.

[26] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

7. Appendix

Table 5: $A(n)$ for n a product of the smallest odd primes

n	factorization of n	$A(n)$	$\log_n A(n)$
105	$3 \cdot 5 \cdot 7$	2	0.148937
1155	$3 \cdot 5 \cdot 7 \cdot 11$	3	0.155791
15015	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$	23	0.326043
255255	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17$	532	0.504147
4849845	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$	*669606	0.871381
111546435	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$	8161018310	1.231662
3234846615	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29$	2888582082500892851	1.941216

*(Koshiba, 2002 [15]).

Centre for Experimental and Constructive Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

E-mail address: `ada26@sfu.ca`

E-mail address: `mmonagan@cecm.sfu.ca`

Table 6: n such that $A(n) > A(m)$ for $m < n$

n	$A(n)$	$\log_n A(n)$
1	1	-
105	2	0.148937
385	3	0.184540
1365	4	0.192037
1785	5	0.214959
2805	6	0.225686
3135	7	0.241716
6545	9	0.250069
10465	14	0.285125
11305	23	0.335958
17255	25	0.329943
20615	27	0.331781
26565	59	0.400255
40755	359	0.554229
106743	397	0.516829
171717	434	0.503836
255255	532	0.504147
279565	1182	0.564147
327845	31010	0.814317
707455	35111	0.777039
886445	44125	0.780927
983535	59815	0.797093
1181895	14102773	1.177309
1752465	14703509	1.147954
3949491	56938657	1.175678
8070699	74989473	1.140162
10163195	1376877780831	1.732388
13441645	1475674234751	1.707101
15069565	1666495909761	1.702652
30489585	2201904353336	1.649191
37495115	2286541988726	1.631796
40324935	2699208408726	1.634490
43730115	862550638890874931	2.347376
169828113	*31484567640915734941	2.369146
185626077	42337944402802720258	2.373635
416690995	80103182105128365570406901971	3.353164
437017385	86711753206816303264095919005	3.349121
712407185	111859370951526698803198257925	3.281324
1250072985	137565800042644454188531306886	3.203113
1311052155	192892314415997583551731009410	3.211947
1880394945	64540997036010911566826446181523888971563	4.400339
2317696095	67075962666923019823602030663153118803367	4.359458
13162764615	**5465808676670557863536977958031695430428633	4.223361

[*] First instance such that $A(n) > 2^{64}$

[**] We have yet to show that $A(13162764615) > A(m)$ for all $m < 13162764615$.

Table 7: $A(n)$ such that $A(n) > n^4$

n	$A(n)$	$\log_n A(n)$
*1880394945	64540997036010911566826446181523888971563	4.400339
*2317696095	67075962666923019823602030663153118803367	4.359458
*2580076785	44178992295210157476612718521873077815356	4.318615
*2667537015	50978590999382303149290759045486931349075	4.318577
2693538705	41311315644168801150352807111933599181273	4.306965
*2929917705	38986521325602066434123587685733770107831	4.287687
2998467615	50703798374791014119206172479361263858737	4.295185
3100110585	37023124240370602498540690923261030138801	4.274245
3405039495	41344572168478125565953675666880875630891	4.261005
3436583865	29420710476505598428925342851511436269885	4.243720
*3629599545	31146417203818726557508589352280096726851	4.235775
3695785665	27544659617064926303551956412846199778051	4.226722
3821066535	31128120202581145792970998453196458758768	4.225879
3825631095	40071361165347422625605753503633105870103	4.237096
3955313505	27667056853522527517062927200210380189261	4.213942
4196909145	24175660733238789429230720957313695678993	4.196578
4218183255	31889907167996287439150048606465589433551	4.208117
4253640105	24726477535222303105776318525136248038167	4.195053
4344360735	32507457200647115397398921638675751619003	4.203392
*4416741615	30381795842095831309232742046965986663843	4.197220
4672030935	22760422406095676210121193519495242514072	4.173655
*4679122305	24693521216952698497060423547227286104011	4.177032
4715312745	18998617883089513487054213546426519137567	4.163816
*4766582535	27232450597992733695751246276504205431221	4.177952
*13162764615	5465808676670557863536977958031695430428633	4.223361

*multiple of 43730115

Table 8: $\Phi_n(z)$ of order 5 that are flatter than all $\Phi_m(z)$ of order 5 for $m < n$, for $n \leq 6.5 \cdot 10^8$

n	factorization of n	$A(n)$
15015	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$	23
23205	$3 \cdot 5 \cdot 7 \cdot 13 \cdot 17$	21
31395	$3 \cdot 5 \cdot 7 \cdot 13 \cdot 23$	15
574665	$3 \cdot 5 \cdot 7 \cdot 13 \cdot 421$	14
774795	$3 \cdot 5 \cdot 7 \cdot 47 \cdot 157$	13
1331715	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 1153$	12
2666895	$3 \cdot 5 \cdot 7 \cdot 11 \cdot 2309$	9
3725085	$3 \cdot 5 \cdot 7 \cdot 13 \cdot 2729$	7
40765935	$3 \cdot 5 \cdot 7 \cdot 43 \cdot 9029$	6
48713385	$3 \cdot 5 \cdot 7 \cdot 47 \cdot 9871$	5
76762245	$3 \cdot 5 \cdot 7 \cdot 59 \cdot 12391$	4

Table 9: $\Phi_n(z)$ of order 5 such that $A(n) = 2$

n	factorization of n
1147113361785	$3 \cdot 5 \cdot 29 \cdot 1741 \cdot 1514671$
2294224451565	$3 \cdot 5 \cdot 29 \cdot 1741 \cdot 3029339$
2576062979535	$3 \cdot 5 \cdot 29 \cdot 2609 \cdot 2269829$
7157926096635	$3 \cdot 5 \cdot 29 \cdot 4349 \cdot 3783629$
7157929880265	$3 \cdot 5 \cdot 29 \cdot 4349 \cdot 3783631$
14031384951165	$3 \cdot 5 \cdot 29 \cdot 6089 \cdot 5297431$
15456385821615	$3 \cdot 5 \cdot 29 \cdot 2609 \cdot 13618981$
36654908721735	$3 \cdot 5 \cdot 29 \cdot 6959 \cdot 12108659$
39282436838685	$3 \cdot 5 \cdot 59 \cdot 3541 \cdot 12535141$
44151142013985	$3 \cdot 5 \cdot 59 \cdot 5309 \cdot 9396929$
44151151410915	$3 \cdot 5 \cdot 59 \cdot 5309 \cdot 9396931$
46392857518515	$3 \cdot 5 \cdot 29 \cdot 7829 \cdot 13622461$

Table 10: $\bar{\alpha}(b)$, the least k for which b occurs as $|a_n(k)|$; and the least n for which $|a_n\big(\bar{\alpha}(k)\big)| = b$; for select $b \leq 927$

a	$\bar{\alpha}(b)$	n
0	1	4
1	0	1
2	7	105
3	17	323323
4	23	1062347
5	30	37182145
6	36	215656441
7	43	65552121635
8	46	845904650955
9	47	75145115045
10	52	30704573184285
20	70	152125131763605
30	82	307444891294245705
40	89	1352450076803386856295
50	95	3929160775540133527939545
60	99	194825753248734022710250308207855
70	100	163957329252276946718326137628485
80	106	4281025919618354440889355
90	106	1200321465765450313917852148868594655
100	112	23806785138997669045785703155
200	132	162938425981534060763635083977029188109663335
300	143	2326975571029326286598990252532796074781464457755
400	153	5412131370764127757636390017210695
500	158	5921057432644596149106845426292101971454108035
600	163	50967866897743398269290966947741571435
700	167	41211036991280460777846257111155083155
800	170	523848308647668419809505921108741216619845
900	173	*480644324429304014052020896687401734836765
927	173	*1269140374116844321897058519227927779943780451272073121291475705

*There may exist smaller n for which $|a_n(k)| = b$.