A fast recursive algorithm for computing cyclotomic polynomials

Andrew Arnold, Michael Monagan

Centre for Experimental and Constructive Mathematics
Simon Fraser University

PASCO 2010, Grenoble, France

Organization of talk

- An introduction to cyclotomic polynomials and the sparse power series (SPS) algorithm
- Improving the SPS algorithm
- A challenge problem: computing giant cyclotomic polynomials
- A look at cyclotomic coefficients

What are cyclotomic polynomials?

Definition

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

$$\Phi_n(z) = \prod_{\substack{0 \le j < n \\ \gcd(j,n)=1}} \left(z - e^{\frac{2\pi i j}{n}}\right).$$

We let the **order** *of $\Phi_n(z)$ denote the number of distinct odd primes dividing its index n.*

Definition

The n_{th} **inverse cyclotomic polynomial**, *$\Psi_n(z)$, is defined as*

$$\Psi_n(z) = (z^n - 1)/\Phi_n(z).$$

A motivating problem

We let $A(n)$ denote the **height** of $\Phi_n(z)$, that is, the absolute value of the largest coefficient of $\Phi_n(z)$.

Theorem (Erdős, 1946)

Fix $c > 0$, then there exists infinitely many n such that $A(n) > n^c$.

Question

Given c, what is the least n such that $A(n) > n^c$?

A motivating problem

We let $A(n)$ denote the **height** of $\Phi_n(z)$, that is, the absolute value of the largest coefficient of $\Phi_n(z)$.

Theorem (Erdős, 1946)

Fix $c > 0$, then there exists infinitely many n such that $A(n) > n^c$.

Question

Given c, what is the least n such that $A(n) > n^c$?

Table: The least n such that $A(n) > n^c$, for $c = 1, 2, 3, 4$

c	n	$A(n)$
1	1181895	14102773
2	43730115	862550638890874931
3	416690995	80103182105128365570406901971
4	1880394945	64540997036010911566826446181523888971563

Some basic identities of cyclotomic polynomials

Lemma

Let $p \mid m$, then $\Phi_{mp}(z) = \Phi_m(z^p)$.

Lemma

Let m be odd, then $\Phi_{2m}(z) = \Phi_m(-z)$.

These two lemmas give an easy means of computing $\Phi_n(z)$ from $\Phi_m(z)$, where m is the largest squarefree odd divisor of n.

Lemma

Let $p \nmid m$, then
$$\Phi_{mp}(z) = \Phi_m(z^p)/\Phi_m(z) = \Phi_m(z^p)\Psi_m(z)/(z^m - 1).$$

This lemma outlines a method of computing $\Phi_n(z)$ for odd, squarefree n by way of a series of polynomial divisions.

The sparse power series (SPS) algorithm

For $n > 1$, we compute $\Phi_n(z)$ as

$$\Phi_n(z) = \prod_{d|n} (1 - z^d)^{\mu(n/d)}.$$

We call the $(1 - z^d)^{\pm 1}$ (alternatively $(z^d - 1)^{\pm 1}$) comprising $\Phi_n(z)$ the **subterms** of $\Phi_n(z)$.

Example

For $n = 105 = 3 \cdot 5 \cdot 7$:

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

We can compute the truncated power series of this product. Multiplying a truncated power series of degree D by either $(1 - z^d)$ or $(1 - z^d)^{-1} = (1 + z^d + z^{2d} + \dots)$ requires $\mathcal{O}(D)$ arithmetic operations in the coefficient domain.

The sparse power series algorithm

Input: $n = p_1 p_2 \cdots p_k$, a product of k distinct primes
Output: the coefficients of $\Phi_n(z) = \sum_{i=0}^{\phi(n)} a(i) z^i$

```
D ⟵ φ(n),                                // truncate to degree φ(n)
a(0), a(1), a(2), ..., a(D) ⟵ 1, 0, 0, ..., 0

for d|n such that d > 0 do
    if μ(n/d) = 1 then                     // multiply by 1 − z^d
        for i = D down to d by −1 do a(i) ⟵ a(i) − a(i − d)
    else                                   // divide by 1 − z^d
        for i = d to D do a(i) ⟵ a(i) + a(i − d)
    end
end
return a(0), a(1), ..., a(D)
```

$\#$ of operations in $\mathbb{Z}$: $\mathcal{O}(2^k \cdot \phi(n))$

Improving the sparse power series algorithm

Let $d_1, d_2, \ldots, d_{2^k}$ be the divisors of n in the order the SPS algorithm iterates through them all. The SPS algorithm computes 2^k truncated power series,

$$f_s(z) = \prod_{i=1}^{s} (1 - z^d)^{\mu(n/d)} \bmod z^{\phi(n)+1}, \text{ for } 1 \leq s \leq 2^k.$$

If, however, $f_t(z)$ is a polynomial of degree $D_t < \phi(n)$, then for $s < t$, we need only compute the terms of $f_s(z)$ to degree D_t.

Aim

Order the divisors of n in an order which minimizes the degree bound at every stage of the computation of $\Phi_n(z)$.

The palindromic property of cyclotomic coefficients

Lemma

Let

$$f(z) = \sum_{k=0}^{D} a(k) z^k = \Phi_{n_1}(z) \cdots \Phi_{n_r}(z)$$

be a degree-D product of cyclotomic polynomials such that $n_1, \ldots, n_r$ are all odd, then

$$a(k) = (-1)^D a(D-k).$$

- For odd n if $f_t = \prod_{j=0}^{t} (1 - z^{d_j})^{n/d_j}$ is a polynomial of degree D_t, then it is exactly a product of cyclotomic polynomials of the form in lemma above. Thus for $s \leq t$, we actually only have to truncate to degree $\lfloor D_t/2 \rfloor$. If the degree bound increases from f_t to f_{t+1}, we can apply this lemma to trivially obtain the higher-degree terms of $f_t(z)$.

SPS2: A first improvement

For a prime $n = mp$,

$$\Phi_n(z) = \Psi_m(z)\Phi_m(z^p)(z^m - 1)^{-1}.$$

We can reexpress this equation in terms of the subterms of $\Phi_n(z)$:

$$\Phi_n(z) = \left(\prod_{d|m, d<m} (z^d - 1)^{\mu(n/d)}\right)\left(\prod_{d|n, p|m} (z^d - 1)^{\mu(n/d)}\right)(z^m - 1)^{-1}$$

- We multiply by the $2^{k-1} - 1$ subterms appearing in the left product first, truncating to degree $\lfloor (m - \phi(m))/2 \rfloor$. This produces the first half of the terms of $\Psi_m(z)$.
- We then apply the palindromic property to yield the higher-degree terms of $\Psi_m(z)$ (up to degree at most $\phi(n)/2$), and multiply by the remaining subterms, truncating to degree $\phi(n)/2$.

SPS2: A first improvement

Example

For $n = 3 \cdot 5 \cdot 7$,

$$\Phi_{105}(z) = \Psi_{15}(z)\Phi_{15}(z^7)(z^{15} - 1)^{-1}$$

Table: A comparison of the degree bound using SPS and SPS2

method	$d\|n$							
	1	3	5	7	15	21	35	105
SPS1	24	24	24	24	24	24	24	24
SPS2	3	3	3	24	24	24	24	24

- This change does not improve the complexity of the algorithm; however, it saves us a factor of 2 in practice.

SPS3: The iterative SPS algorithm

We can apply the identity $\Phi_{mp}(z) = \Psi_m(z)\Phi_m(z^p)(z^m - 1)^{-1}$ iteratively. Let $n = p_1 p_2 \cdots p_k$, a product of k distinct odd primes. For $1 \le i \le k$, let $m_i = p_1 p_2 \cdots p_{i-1}$ and $e_i = p_{i+1} \cdots p_k$. We set $m_1 = e_k = 1$. Note that $e_i p_i m_i = n$ for $1 \le i \le k$. By repeated appliation of the SPS2 identity, we can show that

$$\Phi_n(z) = \left(\prod_{j=2}^{k} \Psi_{m_j}(z^{e_j})\right)\left(\prod_{j=1}^{k}(z^{n/p_j} - 1)^{-1}\right)(z^n - 1)$$

We compute $\Phi_n(z)$ as

$$\Psi_{m_k}(z^{e_k}) \cdots \Psi_{m_2}(z^{e_2}) \cdot \left(\prod_{j=1}^{k}(z^{n/p_j} - 1)^{-1}\right)(z^n - 1)$$

from left to right.

SPS3: The iterative SPS algorithm

Example

$$\Phi_{105}(z) = \Psi_{15}(z)\Psi_3(z^7)(z^{15}-1)^{-1}(z^{21}-1)^{-1}(z^{35}-1)^{-1}(z^{105}-1)$$

Table: A comparison of the degree bound using SPS1-3

method	$d \mid n$							
	1	3	5	7	15	21	35	105
SPS1	24	24	24	24	24	24	24	24
SPS2	3	3	3	24	24	24	24	24
SPS3	3	3	3	7	24	24	24	24

The speedup we see from SPS2-3 is small for $\Phi_n(z)$ of low order; however, these are exactly the $\Phi_n(z)$ that are easy to compute. For $\Phi_n(z)$ or order $k > 2$, the degree bound lowers (over SPS2) for $2^k - 2^{k-1} - k$ subterms of $\Phi_n(z)$. We see considerably speedup (3-5x) for $\Phi_n(z)$ of order ≥ 6.

SPS4: The recursive SPS algorithm

For SPS3 with express $\Phi_n(z)$ as a product of inverse cyclotomic polynomials (plus some additional subterms). We derive an analog for $\Psi_n(z)$. Given $n = p_1 p_2 \cdots p_k$, again let $m_i = p_1 p_2 \cdots p_{i-1}$ and $e_i = p_{i+1} \cdots p_k$, $m_1 = e_k = 1$ then:

$$\Psi_n(z) = \Phi_{m_k}(z^{e_k}) \cdots \Phi_{m_1}(z^{e_1}).$$

Thus we can break $\Phi_n(z)$ into products of $\Psi_m(z)$ of smaller index, and in turn break $\Psi_m(z)$ into products of cyclotomic polynomials of yet smaller index. We recurse until we have a cyclotomic or inverse cyclotomic polynomial of order 1.

SPS4: An example

Consider the case of $\Phi_{105}(z)$ again. SPS3 computes $\Phi_n(z)$ as

$$\Phi_{105}(z) = \Psi_{15}(z)\Psi_3(z^7)(z^{15}-1)^{-1}(z^{21}-1)^{-1}(z^{35}-1)^{-1}(z^{105}-1)$$

However, in light of our new identity for $\Psi_n(z)$, we know SPS3 computes $\Psi_{15}(z)$ in a wasteful manner.

$$= \left(\Phi_3(z)\Phi_1(z^5)\right)\Psi_3(z^7)(z^{15}-1)^{-1}(z^{21}-1)^{-1}(z^{35}-1)^{-1}(z^{105}-1)$$

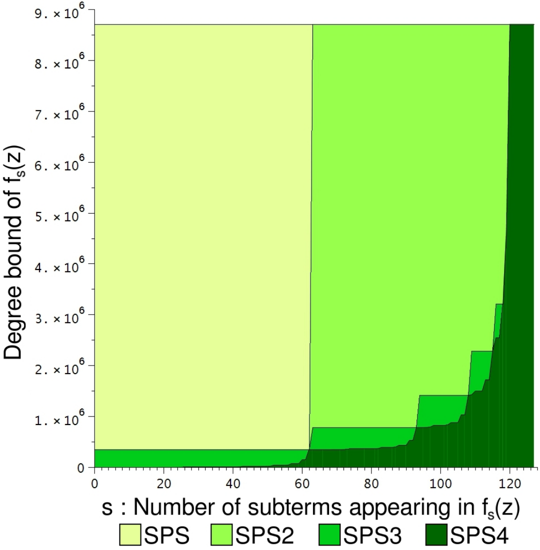
- For the two subterms appearing in $\Phi_3(z) = (z^3-1)/(z-1)$, we truncate to degree $\phi(3)/2 = 1$.
- Then for the remaining subterm in $\Psi_{15}(z)$, $\Phi_1(z^5) = z^5 - 1$, we truncate to half the degree of $\Psi_{15}(z)$, as before with SPS3 and SPS2.

Comparing SPS1-4 on $\Phi_{105}(z)$

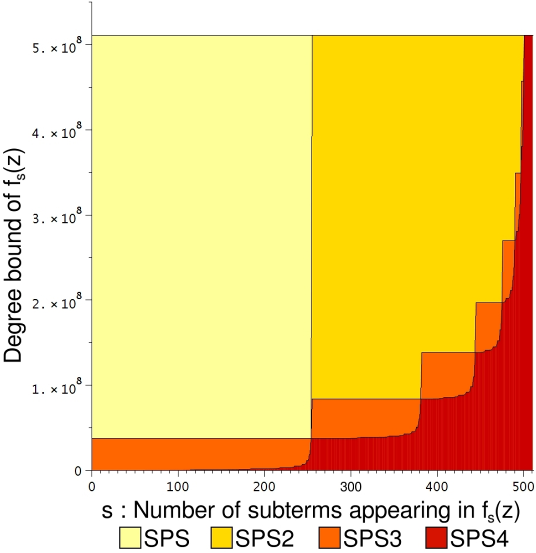
Table: degree bound using SPS1-4 on $\Phi_{105}(z)$

method	$d \mid n$							
	1	3	5	7	15	21	35	105
SPS1	24	24	24	24	24	24	24	24
SPS2	3	3	3	24	24	24	24	24
SPS3	3	3	3	7	24	24	24	24
SPS4	1	1	3	7	24	24	24	24

Comparing SPS1-4 on $\Phi_{43730115}(z)$



Comparing SPS1-4 on $\Phi_{3234846615}(z)$



Timings

Table: Time to calculate $\Phi_n(z)$ (in seconds*) on a Intel 2.7 GHz Core i7

	order	algorithm				
n	of $\Phi_n(z)$	FFT	SPS	SPS2	SPS3	SPS4
255255	6	0.40	0.00	0.00	0.00	0.00
1181895	6	1.76	0.01	0.00	0.00	0.00
4849845	7	7.74	0.12	0.06	0.02	0.01
37182145	7	142.37	1.75	0.95	0.23	0.19
43730115	7	140.62	1.69	0.93	0.23	0.19
111546435	8	295.19	6.94	3.88	1.45	0.94
1078282205	8	-	105.61	58.25	12.34	9.29
3234846615	9	-	432.28	244.44	81.32	49.18

*times are rounded to the nearest hundredth of a second

A challenge problem

T.D. Noe asked us to compute $\Phi_n(z)$, where

$$n = 99660932085 = 3 \cdot 5 \cdot 11 \cdot 13 \cdot 19 \cdot 29 \cdot 37 \cdot 43 \cdot 53.$$

- This is the lcm of the least two integers m satisfying $A(m) > m^4$.
- This polynomial requires a large amount of space; storing the coefficients of $\Phi_n(z)$ as 320-bit integers requires 760 GB. Moreover, we didn't have an array of disks to expedite the computation.

A first attempt

Our first attempt (before we thought of SPS2-4) was to compute $\Phi_n(z)$ modulo 32-bit primes using SPS. Each image is roughly 76 GB. As such, our computation was disk-based. This technique unfortunately was not an effective approach to compute $\Phi_n(z)$. The computation was very slow. Each image of $\Phi_n(z)$ took over two weeks to compute.

It was, however, an effective means of destroying hard disks.

A new approach

We computed $\Psi_m(z)$, where $m = n/53 = 1880394945$, modulo five $64-bit$ primes. We then computed $g_j(z)$, where, given

$$\sum_{k=0}^{\phi(n)/2} c(k)z^k = \Psi_m(z)\cdot(z^m-1)^{-1} \bmod z^{\phi(n)/2+1}.$$

we set $g_j(z)$ as

$$g_j(z) = \sum_{0\le j+53k\le\phi(n)/2} c(j+53k)z^k,$$

in which case,

$$\sum_{j=0}^{52} z^j g_j(z^{53}) = \Psi_m(z)\cdot(z^m-1)^{-1} \bmod z^{\phi(n)/2+1}.$$

A new approach (continued)

$$\sum_{j=0}^{52} z^j g_j(z^{53}) = \Psi_m(z) \cdot (z^m - 1)^{-1} \bmod z^{\phi(n)/2+1}.$$

Thus, as $\Phi_n(z) = \Psi_m(z)(z^m - 1)^{-1}\Phi_m(z^{53})$,

$$\sum_{j=0}^{52} z^j g_j(z^{53})\Phi_m(z^{53}) \equiv \Phi_n(z) \pmod{z^{\phi(n)/2+1}}.$$

Thus to compute all the coefficients of $\Phi_n(z)$, we need only compute $g_j(z)\Phi_m(z)$ for $0 \leq j < 53$. We can compute $g_j(z)\Phi_m(z)$ modulo a 64-bit prime in memory with $> 5GB$ of RAM. We can then reconstruct $g_j(z)\Phi_m(z)$ by way of Chinese remaindering.

A new approach (continued)

$$\sum_{j=0}^{52} z^j g_j(z^{53}) = \Psi_m(z) \cdot (z^m - 1)^{-1} \bmod z^{\phi(n)/2+1}.$$

Thus, as $\Phi_n(z) = \Psi_m(z)(z^m - 1)^{-1}\Phi_m(z^{53})$,

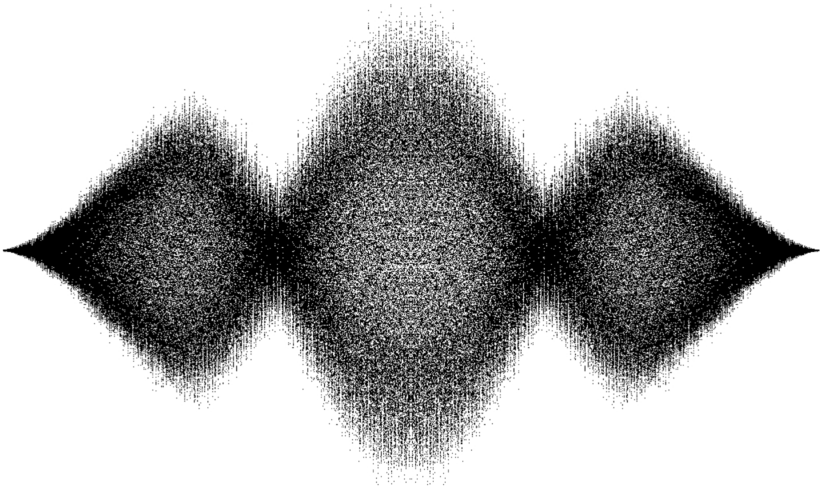
$$\sum_{j=0}^{52} z^j g_j(z^{53})\Phi_m(z^{53}) \equiv \Phi_n(z) \pmod{z^{\phi(n)/2+1}}.$$

Thus to compute all the coefficients of $\Phi_n(z)$, we need only compute $g_j(z)\Phi_m(z)$ for $0 \le j < 53$. We can compute $g_j(z)\Phi_m(z)$ modulo a 64-bit prime in memory with $> 5GB$ of RAM. We can then reconstruct $g_j(z)\Phi_m(z)$ by way of Chinese remaindering. We found that

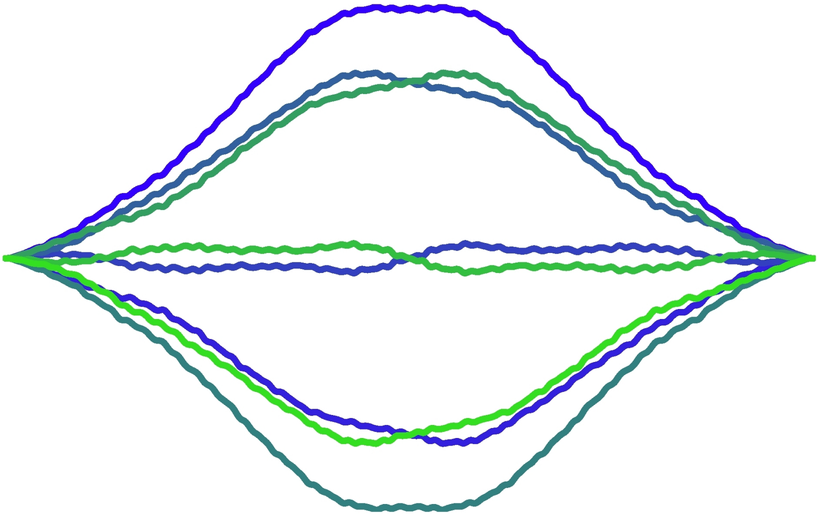
$$A(99660932085) = 61267208717407836670896202324395260$$
$$124725254733381530786789617551493787739155364471853 70,$$

which is roughly $2^{291.6}$ or $n^{7.98}$. The computation took roughly 2 days, distributed over 3 desktop computers.

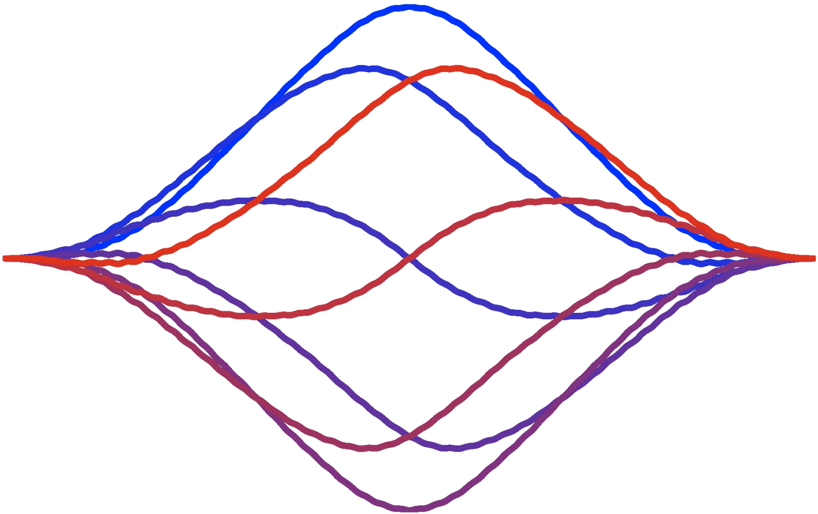
The coefficients of $\Phi_{4849845}(z)$



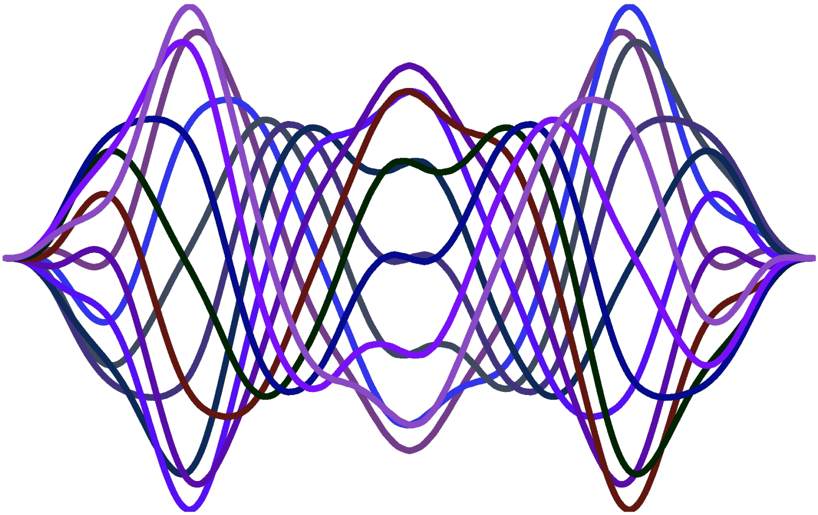
Plots of $\Phi_{1181895}(z)$



Plots of $\Phi_{43730115}(z)$



The coefficients of $\Phi_{40324935}(z)$



Questions?