Sparse and scalable Residue Number Systems from polynomial point of view

Eugene Zima
Joint work with Natalya Ter-Saakov, Robert Dougherty-Bliss and Owen West
Dedicated to 80-th birthday of Sergei Abramov

SCG, University of Waterloo

April 9, 2026

Brief intro

Residue number systems (RNS) based on pairwise relatively prime moduli are a powerful tool for accelerating integer computations via the Chinese Remainder Theorem.

Brief intro

Residue number systems (RNS) based on pairwise relatively prime moduli are a powerful tool for accelerating integer computations via the Chinese Remainder Theorem. When deciding a particular approach to the choice of moduli m_i one can try to satisfy the following natural requirements:

- $\gcd(m_i, m_j) = 1$ for $i \neq j$;
- reduction modulo m_i is "simpler" than division with remainder;
- products of moduli and their inverses have a sparse bit pattern;
- the moduli m_i all have roughly the same bit length.

Brief intro

Residue number systems (RNS) based on pairwise relatively prime moduli are a powerful tool for accelerating integer computations via the Chinese Remainder Theorem. When deciding a particular approach to the choice of moduli m_i one can try to satisfy the following natural requirements:

- $\gcd(m_i, m_j) = 1$ for $i \neq j$;
- reduction modulo m_i is "simpler" than division with remainder;
- products of moduli and their inverses have a sparse bit pattern;
- the moduli m_i all have roughly the same bit length.

We add another condition:

- the moduli should be scalable.

We add another condition:

- the moduli should be scalable.

100101

We add another condition:

- the moduli should be scalable.

100101

$2- > 2^2$

We add another condition:

- the moduli should be scalable.

100101

$2 -> 2^2$

10000010001

We add another condition:

- the moduli should be scalable.

100101

$2 -> 2^2$

10000010001

or

10000000101

We add another condition:

- the moduli should be scalable.

100101
$2 -> 2^2$
10000010001
or
10000000101
Scalability of moduli means that a fixed set of moduli can be adjusted in size to any desired precision for integer arithmetic. We will give a precise definition later...

We add another condition:

- the moduli should be scalable.

100101
$2 -> 2^2$
10000010001
or
10000000101
Scalability of moduli means that a fixed set of moduli can be adjusted in size to any desired precision for integer arithmetic. We will give a precise definition later...

We start with several examples of different choice of moduli. In these examples we write the $2n$ bit integer u as $u_l 2^n + u_r$, where $u_l, u_r < 2^n$.

Example (1)

Consider moduli set $\{2^n, 2^n - 1, 2^n + 1\}$ for $n > 0$. This set of moduli satisfies every requirement. They are relatively prime, have roughly the same bitlength, reduction modulo m_i is division free (using $u \equiv u_l \pm u_r \pmod{2^n \mp 1}$ and $u \bmod 2^n = u_r$), products of any two have a sparse bit pattern, and the relevant modular inverses are simple:

$$(2^n + 1)^{-1} \bmod (2^n - 1) = 2^{n-1}$$
$$(2^n - 1)^{-1} \bmod (2^n) = 2^n - 1$$
$$(2^n(2^n + 1))^{-1} \bmod (2^n - 1) = 2^{n-1}$$
$$\dots$$

These formulas and properties hold up under scaling ($2 \to 2^c$).

Let $m_1 = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^n$, $M = m_1 m_2 m_3$, $M_i = M/m_i$, and $v_i = M_i^{-1} \bmod m_i$. Then

$$M_1 = 2^{2n} + 2^n, M_2 = 2^{2n} - 2^n, M_3 = 2^{2n} - 1,$$

$$v_1 = 2^{n-1}, v_2 = 2^{n-1} + 1, v_3 = 2^n - 1.$$

Let $m_1 = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^n$, $M = m_1 m_2 m_3$, $M_i = M/m_i$, and $v_i = M_i^{-1} \bmod m_i$. Then

$$M_1 = 2^{2n} + 2^n, M_2 = 2^{2n} - 2^n, M_3 = 2^{2n} - 1,$$

$$v_1 = 2^{n-1}, v_2 = 2^{n-1} + 1, v_3 = 2^n - 1.$$

Given residues $a_i = a \bmod m_i$ the reconstruction of unique $0 \leq a < M$ is multiplication and division free:

$$\left[\sum_i (a_i v_i \bmod m_i) M_i \right] \bmod M.$$

However, there are only three moduli.

Let $m_1 = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^n$, $M = m_1 m_2 m_3$, $M_i = M/m_i$, and $v_i = M_i^{-1} \bmod m_i$. Then

$$M_1 = 2^{2n} + 2^n, M_2 = 2^{2n} - 2^n, M_3 = 2^{2n} - 1,$$

$$v_1 = 2^{n-1}, v_2 = 2^{n-1} + 1, v_3 = 2^n - 1.$$

Given residues $a_i = a \bmod m_i$ the reconstruction of unique $0 \leq a < M$ is multiplication and division free:

$$\left[\sum_i (a_i v_i \bmod m_i) M_i \right] \bmod M.$$

However, there are only three moduli.

$$T(n) = 3T(2n/3) + f(n)...$$

Example (2)

Another well studied set of moduli are those of the form $2^n - 1$ with relatively prime exponents. Schönhage [11, Ch. 4.3] showed how to generate arbitrarily large sets of relatively prime moduli of this type with similar bitlengths. All requirements are met except for scalability. For example, replacing 2 with 2^c can destroy the relative primality of $2^n - 1$ and $2^m - 1$. The inverses have a closed form bit pattern, but the pattern is dense and not preserved with the growth of moduli length.

For moduli of the form $m_i = 2^{a2^i} + 1, \quad i = 0, 1, \ldots, k$, ($a$ is an arbitrary positive integer)

$$M_i^{-1} \bmod m_i = 2^{a2^i - 1} - 2^{a-1} + 1, \quad i = 1, 2, ..., k,$$

where $M_i = \prod_{j=0}^{i-1} m_j, \quad i = 0, 1, \ldots, k-1.$

Example (3)

Consider moduli of the form $2^n + 1$ with all exponents having distinct 2-adic valuation $\nu_2(n)$. It was shown in [14, 1] that these moduli are scalable and required inverses have scalable bit patterns. For example,

$$(2^{24c} + 1)^{-1} \bmod (2^{16c} + 1) = 2^{16c-1} + 2^{8c-1} + 1$$
$$(2^{16c} + 1)^{-1} \bmod (2^{24c} + 1) = 2^{24c-1} - 2^{16c-1} - 2^{8c-1} + 1$$

for any natural $c > 0$. This choice of moduli satisfies almost all requirements except for balanced bitlengths; the largest modulus may have twice as many bits as the smallest.

Example (3)

Consider moduli of the form $2^n + 1$ with all exponents having distinct 2-adic valuation $\nu_2(n)$. It was shown in [14, 1] that these moduli are scalable and required inverses have scalable bit patterns. For example,

$$(2^{24c} + 1)^{-1} \bmod (2^{16c} + 1) = 2^{16c-1} + 2^{8c-1} + 1$$
$$(2^{16c} + 1)^{-1} \bmod (2^{24c} + 1) = 2^{24c-1} - 2^{16c-1} - 2^{8c-1} + 1$$

for any natural $c > 0$. This choice of moduli satisfies almost all requirements except for balanced bitlengths; the largest modulus may have twice as many bits as the smallest.
Another problem is that moduli bit-length grows exponentially with the size of moduli set: k moduli with exponents of distinct 2-adic valuation need to be at least 2^k bits long.

Example (4)

Families of moduli of the form $2^n - 2^j + 1, 0 < j < n$ with fixed n were considered in [3, 5]. These moduli are perfectly balanced in bit-length, sometimes preserve relative primality under simple scaling, and also sometimes have scalable inverses. For example,

$$(2^{20c} - 2^{12c} + 1)^{-1} \bmod (2^{20c} - 2^{4c} + 1) = 2^{8c} + 1$$

for any natural $c > 0$. Reduction modulo m_i is division free using $u \equiv u_r + (2^j - 1)u_l (\bmod\ 2^n - 2^j + 1)$. If $j \leq \alpha n$ with $0 < \alpha < 1$ then the number of division-free steps in the reduction is bounded by $\lceil (1 - \alpha)^{-1} \rceil + 1$ and uses only shifts, additions and subtractions (i.e. has running-time in $\Theta(n)$).

Small sets of these trinomial moduli were successfully used to improve the performance of FFLAS-FFPACK [4] on a standard matrix multiplication benchmark with very large entries (2^{18} or more bits).

Small sets of these trinomial moduli were successfully used to improve the performance of FFLAS-FFPACK [4] on a standard matrix multiplication benchmark with very large entries (2^{18} or more bits). The matrices were reduced by a "top layer" of trinomial moduli, and the residues were handed off to FFLAS-FFPACK. The final answer was reconstructed in a way that exploited the sparsity of these trinomials, as described in [3].

Small sets of these trinomial moduli were successfully used to improve the performance of FFLAS-FFPACK [4] on a standard matrix multiplication benchmark with very large entries (2^{18} or more bits). The matrices were reduced by a "top layer" of trinomial moduli, and the residues were handed off to FFLAS-FFPACK. The final answer was reconstructed in a way that exploited the sparsity of these trinomials, as described in [3].

Unfortunately, it was shown in [5] that a suitable set of k trinomial moduli must have a large degree n as k grows. For $k = 11$ the only known set of scalable moduli has $n \approx 10^{2774}$.

What is common for scalable examples above?

What is common for scalable examples above?

We would like to be able to answer reasonable questions ... For example, given two relatively prime integers with sparse bit pattern $(1001001)_2, (1010001)_2$ will they preserve co-primality and other properties under scaling?

What is common for scalable examples above?

We would like to be able to answer reasonable questions ... For example, given two relatively prime integers with sparse bit pattern $(1001001)_2, (1010001)_2$ will they preserve co-primality and other properties under scaling?
In what follows we offer a "unified" approach to the selection of good moduli.

What is common for scalable examples above?

We would like to be able to answer reasonable questions ... For example, given two relatively prime integers with sparse bit pattern $(1001001)_2, (1010001)_2$ will they preserve co-primality and other properties under scaling?
In what follows we offer a "unified" approach to the selection of good moduli.
Pick monic $f(x), g(x) \in \mathbb{Z}[x]$ with f_0, g_0 odd.

What is common for scalable examples above?

We would like to be able to answer reasonable questions ... For example, given two relatively prime integers with sparse bit pattern $(1001001)_2, (1010001)_2$ will they preserve co-primality and other properties under scaling?
In what follows we offer a "unified" approach to the selection of good moduli.
Pick monic $f(x), g(x) \in \mathbb{Z}[x]$ with f_0, g_0 odd.
Consider integers $f(2^\ell), g(2^\ell)$ as candidates for moduli.
Note that scaling preserves sparsity ...

What properties $f(x), g(x)$ must have to produce "good" pair of integers?

What properties $f(x), g(x)$ must have to produce "good" pair of integers?
Obviously they must be relatively prime... but this is not sufficient.

What properties $f(x), g(x)$ must have to produce "good" pair of integers?
Obviously they must be relatively prime... but this is not sufficient.
Need $f(2^\ell), g(2^\ell)$ to be relatively prime integers for all large enough ℓ.

What properties $f(x), g(x)$ must have to produce "good" pair of integers?
Obviously they must be relatively prime... but this is not sufficient.
Need $f(2^\ell), g(2^\ell)$ to be relatively prime integers for all large enough ℓ.
But this is also not sufficient...

Elementary preliminaries

We assume that integers are represented in *Sparse Balanced Binary* (SBB) form. This is also known as non-adjacent form [8] or canonical signed digit representation [9]. SBB uniquely represents integers with signed bits: $x \in \mathbb{Z}$ is written as

$$x = \sum_{i=0}^{n} b_i 2^i \text{ with } b_i \in \{-1, 0, 1\} \tag{1}$$

under the restriction that no two adjacent bits are both set. (In other words, $b_i \cdot b_{i+1} = 0$ for $i = 0, 1, \ldots, n-1$.)

Definition

For any integer u, let $l(u)$ be the bit-length of u and $s(u)$ the number of set bits when u is written in SBB form.

Definition

A rational a/b is dyadic if $\gcd(a, b) = 1$ and b is a power of 2. A polynomial $f \in \mathbb{Q}[x]$ is dyadic if its coefficients are dyadic. The dyadic depth, $B(f)$, of a dyadic polynomial is the smallest nonnegative integer m such that $2^m f(x)$ has only integer coefficients.

For any polynomial $f \in \mathbb{Q}[x]$, $\deg f$ is its degree and $\mathrm{lc}(f)$ is its leading coefficient. The resultant of two polynomials is denoted by $\mathrm{res}(f, g)$.

Elementary observation

Evaluating dyadic polynomials at powers of two gives integers in SBB form (provided that coefficients are written in SBB form). In particular, scaling these integers ($2 \to 2^c$) preserves the bit structure.

Elementary observation

Evaluating dyadic polynomials at powers of two gives integers in SBB form (provided that coefficients are written in SBB form). In particular, scaling these integers ($2 \to 2^c$) preserves the bit structure.

Proposition

If $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$ *is dyadic with* $a_0 \in \mathbb{Z}$ *and* $\ell > B(f) + \max_j l(\mathrm{numer}(a_j))$, *then*

$$f(2^\ell) = a_n 2^{n\ell} + a_{n-1} 2^{(n-1)\ell} + \cdots + a_0$$

is an integer in SBB form such that

$$s(f(2^\ell)) = \sum_{i=0}^{n} s(\mathrm{numer}(a_i)).$$

Definition

A pair of polynomials with integer coefficients dyadically resolve if their resultant is a signed power of 2, meaning it is of the form $\pm 2^k$ for some nonnegative integer k.

Lemma

If $f, g \in \mathbb{Z}[x]$ *dyadically resolve, then there are dyadic* $s, t \in \mathbb{Q}[x]$ *such that* $s(x)f(x) + t(x)g(x) = 1$ *and* $\deg(s(x)) < \deg(g(x))$, $\deg(t(x)) < \deg(f(x))$.

Scalability

Definition

Given two coprime integers a and b, we say that a and b are **scalable moduli** if there exist monic polynomials $f(x), g(x) \in \mathbb{Z}[x]$, polynomials $p(x), q(x) \in \mathbb{Q}(x)$, and natural c such that:

1. $f(2) = a$ and $g(2) = b$
2. $\gcd(f(2^\ell), g(2^\ell)) = 1$
3. $p(2^\ell)$ and $q(2^\ell)$ are integers such that $p(2^\ell) = f(2^\ell)^{-1} \bmod g(2^\ell)$ and $q(2^\ell) = g(2^\ell)^{-1} \bmod f(2^\ell)$
4. $\mathrm{lc}(p), \mathrm{lc}(q) > 0$, $\deg p(x) \leq \deg g(x)$ and $\deg q(x) \leq \deg f(x)$

for $\ell \geq c$.

Theorem

Let f and g be monic polynomials over $\mathbb{Z}$ such that $f(0), g(0) \in \{-1, 1\}$. If $f(x)$ and $g(x)$ dyadically resolve, then $f(2)$ and $g(2)$ are scalable moduli.

Since $f(x)$ and $g(x)$ dyadically resolve, there are $s, t \in \mathbb{Q}[x]$ with dyadic coefficients such that

$$s(x)f(x) + t(x)g(x) = 1 \tag{2}$$

and $\deg(s(x)) < \deg(g(x))$, $\deg(t(x)) < \deg(f(x))$.

If $s_0 \in \mathbb{Z}$ (thus $t_0 \in \mathbb{Z}$) and $\mathrm{lc}(s) > 0$ we are done: set $p(x) = s(x)$.
For large enough ℓ both $s(2^\ell)$ and $t(2^\ell)$ are integers such that $s(2^\ell) = f(2^\ell)^{-1} \bmod g(2^\ell)$ (directly follows from (2)).

If $s_0 \in \mathbb{Z}$ (thus $t_0 \in \mathbb{Z}$) and $\mathrm{lc}(s) > 0$ we are done: set $p(x) = s(x)$. For large enough ℓ both $s(2^\ell)$ and $t(2^\ell)$ are integers such that $s(2^\ell) = f(2^\ell)^{-1} \bmod g(2^\ell)$ (directly follows from (2)).

If $s_0 \in \mathbb{Z}$ and $\mathrm{lc}(s) < 0$ set $p(x) = g(x) + s(x)$. For large enough ℓ the values $p(2^\ell)$ are integers such that $0 < p(2^\ell) < g(2^\ell)$ and $p(2^\ell) = f(2^\ell)^{-1} \bmod g(2^\ell)$. Follows from

$$(s(x) + g(x))f(x) + (t(x) - f(x))g(x) = 1$$

If $s_0 \notin \mathbb{Z}$ set $\delta = \lceil |s_0| \rceil - |s_0|$,

$$\gamma = \begin{cases} \delta, & sign(s_0 g_0) > 0 \\ 1 - \delta, & \text{otherwise} \end{cases}$$

and $p(x) = \gamma g(x) + s(x)$.
Note, that $0 < \mathrm{lc}(p) < 1$. For large enough ℓ the values $p(2^\ell)$ are integers such that $0 < p(2^\ell) < g(2^\ell)$ and $p(2^\ell) = f(2^\ell)^{-1} \bmod g(2^\ell)$.
Follows from

$$(s(x) + \gamma g(x))f(x) + (t(x) - \gamma f(x))g(x) = 1$$

Example (1)

The polynomials $\{x, x-1, x+1\}$ pairwise dyadically resolve:

$$\begin{aligned} \text{res}(x, x-1) &= -1 \\ \text{res}(x, x+1) &= 1 \\ \text{res}(x-1, x+1) &= 2. \end{aligned}$$

Example (2)

It is well-known that $\gcd(2^a - 1, 2^b - 1) = 2^{\gcd(a,b)} - 1$, but the polynomials $x^a - 1$ and $x^b - 1$ never dyadically resolve if $a, b > 0$, because their resultant is 0.

Example (3)

Any pair of relatively prime Fermat polynomials $x^n + 1$ and $x^m + 1$ dyadically resolve. This can be seen by a direct resultant computation involving roots of unity, or by using Swan's formula for the resultant of two binomials [13]:

$$\begin{aligned} \operatorname{res}(x^n + 1, x^m + 1) &= (-1)^m((-1)^{m/d} - (-1)^{n/d})^d \\ &= \begin{cases} \pm 2^d, & \nu_2(n) \neq \nu_2(m) \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where $d = \gcd(n, m)$ and $\nu_2(n)$ is the 2-adic valuation of n.

Example (4)

The trinomials $x^n - x^j + 1$ and $x^n - x^\ell + 1$ dyadically resolve sometimes.

$$\begin{aligned}
\operatorname{res}(x^{30} - x + 1, x^{30} - x^7 + 1) &= 63 \\
\operatorname{res}(x^{30} - x^2 + 1, x^{30} - x^7 + 1) &= 31 \\
\operatorname{res}(x^{30} - x^3 + 1, x^{30} - x^7 + 1) &= 3 \\
\operatorname{res}(x^{30} - x^4 + 1, x^{30} - x^7 + 1) &= 7 \\
\operatorname{res}(x^{30} - x^5 + 1, x^{30} - x^7 + 1) &= 3 \\
\operatorname{res}(x^{30} - x^6 + 1, x^{30} - x^7 + 1) &= 1.
\end{aligned}$$

Of the 406 pairs of trinomials of the form $x^{30} - x^j + 1$ with $0 < j < 30$, 115 of them dyadically resolve. We are unaware of a way to characterize which pairs of trinomials dyadically resolve other than directly computing the resultant (although there are some shortcuts outlined in [5]).

Scalability of products

Consider polynomials $f_1(x), f_2(x), g(x)$ that pairwise dyadically resolve. Since

$$\mathrm{res}(f_1(x)f_2(x), g(x)) = \mathrm{res}(f_1(x), g(x))\,\mathrm{res}(f_2(x), g(x)),$$

$f_1(x)f_2(x)$ and $g(x)$ dyadically resolve. This means that it is easy to construct (as shown before) a polynomial $p(x)$ such that

$$p(2^\ell) = (f_1(2^\ell)f_2(2^\ell))^{-1} \bmod g(2^\ell)$$

for all large enough ℓ.

This can be simplified assuming we already have $p_1(x), p_2(x)$ that generate inverses $p_1(2^\ell)$ and $p_2(2^\ell)$ modulo $g(2^\ell)$.
Set $p(x) = (p_1(x)p_2(x)) \mod g(x)$. If $p(0) \in \mathbb{Z}$ and $\mathrm{lc}(p(x)) > 0$, then we are done, otherwise use correcting construction (as shown before) to produce $p(x)$ with integer constant term and positive leading coefficient.
It is important to note here, that $\deg p(x) \leq \deg g(x)$ and $B(p(x)) \leq B(p_1(x)) + B(p_2(x))$. The last equality means that the minimal scaling factor that guarantees the bit pattern can grow, if both $p_1(x)$ and $p_2(x)$ have positive dyadic depth.

The above argument can be repeated for inverses of products of arbitrary sets of moduli incrementally.

The above argument can be repeated for inverses of products of arbitrary sets of moduli incrementally.
Recall that Garner's reconstruction algorithm requires inverses $(m_0 \cdots m_i)^{-1} \bmod m_{i+1}$ for $i = 1, 2, \ldots, k-1$.

The above argument can be repeated for inverses of products of arbitrary sets of moduli incrementally.
Recall that Garner's reconstruction algorithm requires inverses $(m_0 \cdots m_i)^{-1} \bmod m_{i+1}$ for $i = 1, 2, \ldots, k-1$.
If m_{i+1} is generated by a polynomial $g(x)$ of degree n, then the polynomial $p(x)$ generating inverses $(m_0 \cdots m_i)^{-1} \bmod m_{i+1}$ will have degree $\leq n$. It follows that the inverses, scaled by large scaling factor L, will have bit-length in $\Theta(nL)$ but only $\Theta(n)$ bits set.

Looking for moduli

The ideal scenario for Chinese remaindering is to find a large set of polynomials which pairwise dyadically resolve.

Definition

A set of monic polynomials over $\mathbb{Z}$ that pairwise dyadically resolve is called a *clique*.

Looking for moduli

The ideal scenario for Chinese remaindering is to find a large set of polynomials which pairwise dyadically resolve.

Definition

A set of monic polynomials over $\mathbb{Z}$ that pairwise dyadically resolve is called a *clique*.

These sets directly correspond to cliques in certain graphs: If G is a graph whose vertices are monic polynomials in $\mathbb{Z}[x]$ where the edge $\{f, g\}$ exists if and only if f and g dyadically resolve, then the cliques of G are precisely the cliques defined above.
The natural questions are:

1. Are there arbitrarily large cliques?
2. Are large cliques easy to find?

Table: Growth rate of cliques of trinomials. Smallest degree n for cliques of size k consisting of only the given trinomials.

k	$x^n - x^j + 1$	$x^n - x^j - 1$	$x^n + x^j - 1$
2	3	3	3
3	5	4	4
4	5	6	6
5	10	8	8
6	11	12	12
7	22	20	20
8	41	36	36
9	82	48	48
10	1668	120	120
11	$< 10^{2774}$	120	120
12	$< 10^{10458}$	240	240
13	$< 10^{40968}$	720	720
14	$< 10^{166457}$	2160	2160

Side note

When are $x^n - x^k + 1$ and $x^n - x^j + 1$ relatively prime and what is their GCD if they are not?

Proposition

$\gcd(x^n - x^k + 1, x^n - x^j + 1) \neq 1$ *if and only if all of the following conditions hold:*

- $\nu_2(n) > \nu_2(k) = \nu_2(j)$
- $\nu_3(n) = \nu_3(k) = \nu_3(j)$
- $k/n \equiv j/n \equiv -1 (\mathrm{mod} 3)$

In this case

$$\gcd(x^n - x^k + 1, x^n - x^j + 1) = \Phi_6(x^{\gcd(n,k,j)}) = x^{2\gcd(n,k,j)} - x^{\gcd(n,k,j)} + 1.$$

Example ($x^n - x^k + 1$)

In [5], it was shown that there are arbitrarily large cliques consisting of polynomials of the form $x^n - x^k + 1$ with n fixed and $0 < k < n$.

Example ($x^n - x^k + 1$)

In [5], it was shown that there are arbitrarily large cliques consisting of polynomials of the form $x^n - x^k + 1$ with n fixed and $0 < k < n$. To be precise, if $\omega(n)$ is the size of the largest clique of these trinomials, then $\lim_{n\to\infty} \omega(n) = \infty$.

Example ($x^n - x^k + 1$)

In [5], it was shown that there are arbitrarily large cliques consisting of polynomials of the form $x^n - x^k + 1$ with n fixed and $0 < k < n$. To be precise, if $\omega(n)$ is the size of the largest clique of these trinomials, then $\lim_{n\to\infty} \omega(n) = \infty$.
The proof is constructive but extremely pessimistic; the resulting cliques are of no practical use.

Example ($x^n - x^k + 1$)

In [5], it was shown that there are arbitrarily large cliques consisting of polynomials of the form $x^n - x^k + 1$ with n fixed and $0 < k < n$.
To be precise, if $\omega(n)$ is the size of the largest clique of these trinomials, then $\lim_{n\to\infty} \omega(n) = \infty$.
The proof is constructive but extremely pessimistic; the resulting cliques are of no practical use.
It was also shown in [5] that $\omega(n) \leq 2 \log_2 n$, so there is not much hope that a better construction exists.

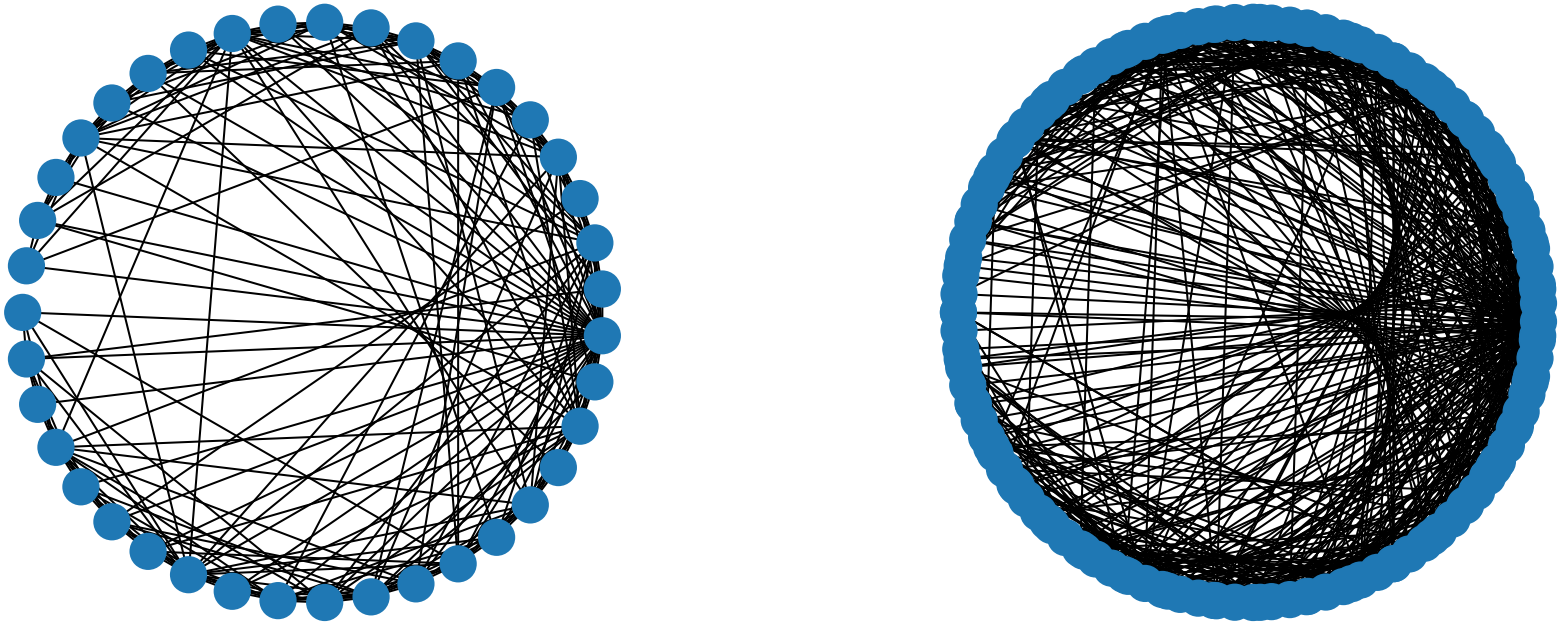


Figure: $T(40)$ and $T(100)$

Example ($x^n - x^k - 1$)

The trinomials $x^n - x^k - 1$ have larger cliques with smaller n compared to the previous family, but empirically demonstrate similar logarithmic behavior.

Similar proposition hods:
In this case if gcd is non-trivial then

$$\gcd(x^n - x^k - 1, x^n - x^j - 1) = \Phi_3(x^{\gcd(n,k,j)}) = x^{2\gcd(n,k,j)} + x^{\gcd(n,k,j)} + 1.$$

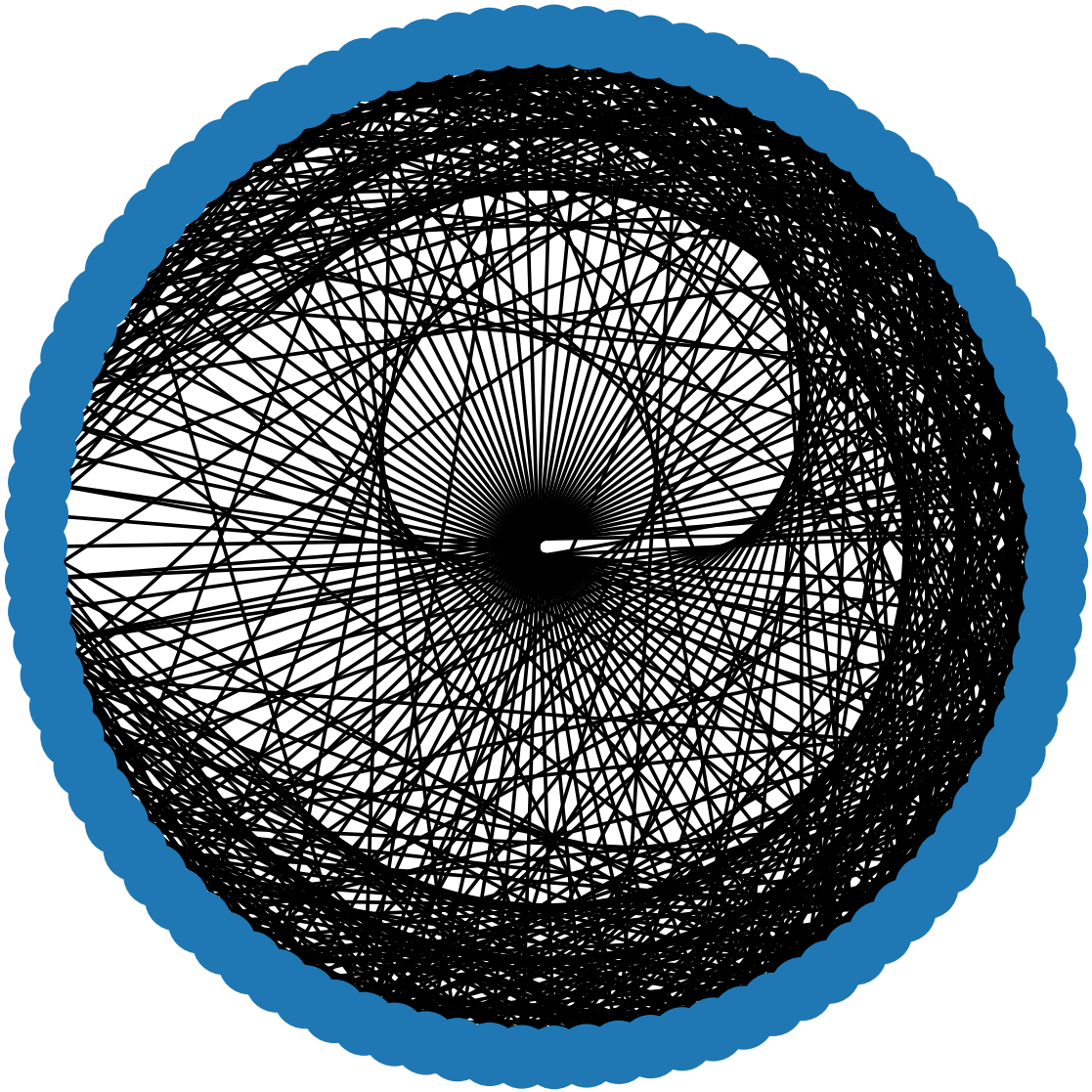


Figure: $T(100)$

Example ($x^n + x^k - 1$)

The reciprocal polynomial of $x^n + x^k - 1$ is $-(x^n - x^{n-k} - 1)$. This map is a bijection between this family and the previous family of trinomials $x^n - x^k - 1$ that preserves resultants, so the clique sizes are exactly the same. (In other words, this is a graph isomorphism.)

Example ($x^n + x^k + 1$)

No two trinomials of the form $x^n + x^k + 1$ dyadically resolve. The resultant of any two is always divisible by 3. This can be seen by reasoning as follows: If $a(x)(x^n + x^k + 1) + b(x)(x^n + x^j + 1) = d$ for some $a, b \in \mathbb{Z}[x]$ and $d \in \mathbb{Z}$, then setting $x = 1$ shows that d is a multiple of 3. The resultant of $x^n + x^k + 1$ and $x^n + x^j + 1$ is one such d.

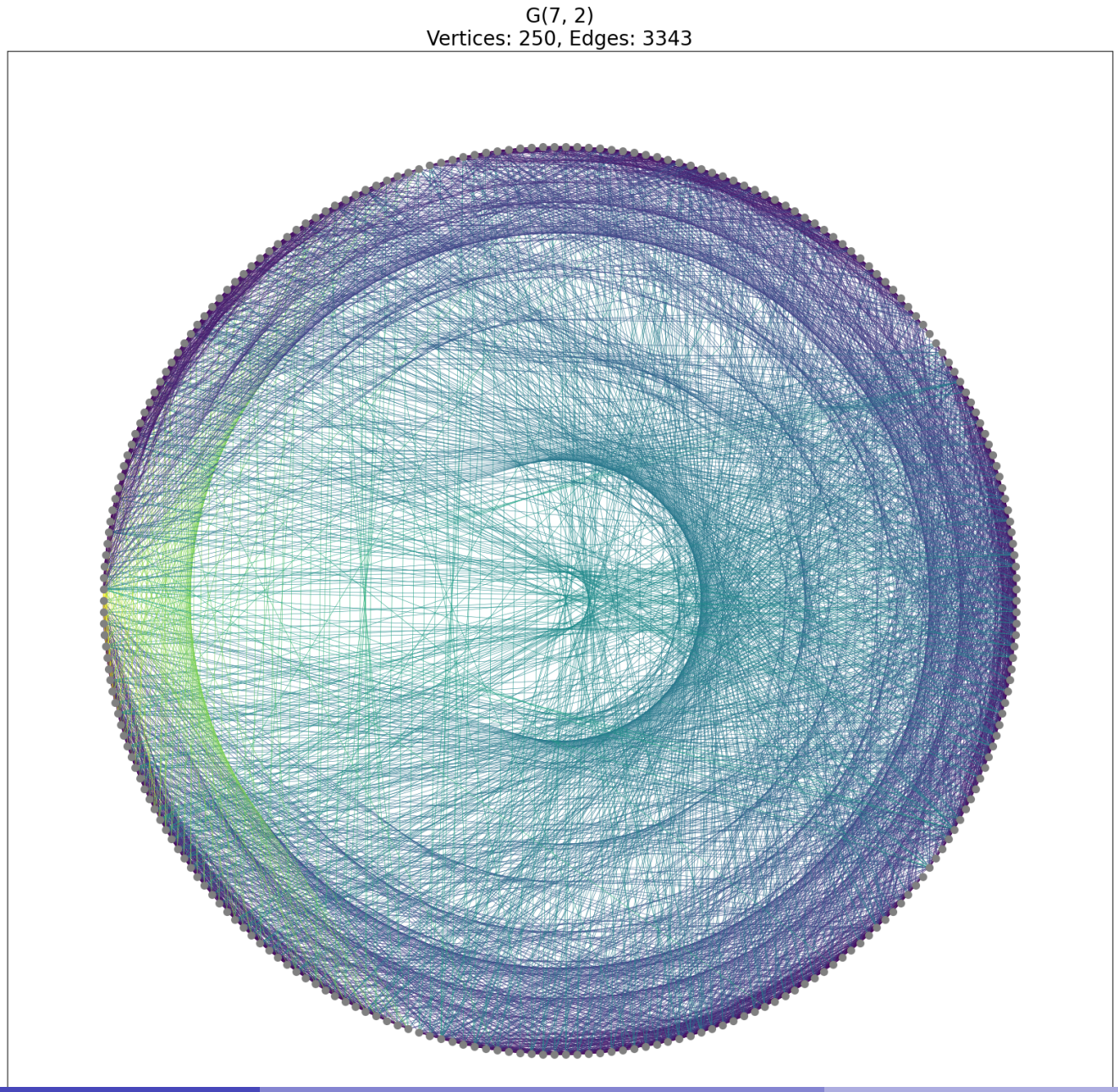
Other families

Definition

Let $G(n, m)$ be the graph whose vertices are all monic polynomials in $\mathbb{Z}[x]$ of degree n with coefficients bounded by m in absolute value. The edge $\{f, g\}$ exists if and only if f and g dyadically resolve.

Table: Largest cliques with polynomials in $\mathbb{Z}[x]$ by degree, coefficients bounded in absolute value.

n	m	size of largest clique in $G(n, m)$
2	5	6
3	5	11
4	5	13
5	3	15
6	2	17


G(7, 2)
Vertices: 250, Edges: 3343

Practical concerns

Some cliques are better than others. The most desirable properties are:

1. polynomials with the same degree, so the moduli are balanced
2. relatively sparse polynomials and inverse polynomials
3. limited dyadic depth of the inverse polynomials
4. second-highest power of x is small relative to the degree.

The last wish is to enable fast reduction.
Every clique can be quantitatively evaluated against the above requirements with simple symbolic computations. The sparsity and second-highest power of x is obvious, the inverses are essentially the Bézout cofactors, the SBB form can be read from the coefficients, and so on.

Consider $a = 1021 = 2^{10} - 2^2 + 1$ and $b = 1013 = 2^{10} - 2^4 + 2^2 + 1$.

Consider $a = 1021 = 2^{10} - 2^2 + 1$ and $b = 1013 = 2^{10} - 2^4 + 2^2 + 1$.

1. Construct $f(x)$ and $g(x)$ such that $a = f(2), b = g(2)$;
2. Compute $\mathrm{res}(f(x), g(x))$ or Bézout's cofactors;
3. See what you get.

Consider $a = 1021 = 2^{10} - 2^2 + 1$ and $b = 1013 = 2^{10} - 2^4 + 2^2 + 1$.

1. Construct $f(x)$ and $g(x)$ such that $a = f(2), b = g(2)$;
2. Compute $\mathrm{res}(f(x), g(x))$ or Bézout's cofactors;
3. See what you get.

First try: let $f(x) = x^{10} - x^2 + 1$ and $g(x) = x^{10} - x^4 + x^2 + 1$, then $\mathrm{res}(f(x), g(x)) = 961$. Bad luck! Bézout's cofactors are

$$\frac{15}{31}x^8 - \frac{1}{31}x^6 - \frac{2}{31}x^4 - \frac{19}{31}x^2 + \frac{8}{31}, -\frac{15}{31}x^8 + \frac{1}{31}x^6 + \frac{2}{31}x^4 + \frac{4}{31}x^2 + \frac{23}{31}$$

Consider $a = 1021 = 2^{10} - 2^2 + 1$ and $b = 1013 = 2^{10} - 2^4 + 2^2 + 1$.

1. Construct $f(x)$ and $g(x)$ such that $a = f(2), b = g(2)$;
2. Compute $\mathrm{res}(f(x), g(x))$ or Bézout's cofactors;
3. See what you get.

First try: let $f(x) = x^{10} - x^2 + 1$ and $g(x) = x^{10} - x^4 + x^2 + 1$, then $\mathrm{res}(f(x), g(x)) = 961$. Bad luck! Bézout's cofactors are

$$\frac{15}{31}x^8 - \frac{1}{31}x^6 - \frac{2}{31}x^4 - \frac{19}{31}x^2 + \frac{8}{31}, -\frac{15}{31}x^8 + \frac{1}{31}x^6 + \frac{2}{31}x^4 + \frac{4}{31}x^2 + \frac{23}{31}$$

Another try: let $f(x) = x^{10} - x^2 + 1$ and $g(x) = x^{10} - 3x^2 + 1$, then $\mathrm{res}(f(x), g(x)) = 2^{10}$; Bézout's cofactors are

$$\frac{3}{2} - \frac{x^8}{2}, -\frac{1}{2} + \frac{x^8}{2},$$

which guarantees both – relative primality of $f(2^\ell), g(2^\ell)$ and existence of polynomials $p(x), q(x)$ from definition of scalability for any $\ell \geq 1$ (run Maple).

There are numerous open questions from this project. We list three, in no particular order:

1. Is there a formula for the resultant of two trinomials similar to Swan's formula for the resultant of two binomials [13]? $\text{res}(x^n - x^k + 1, x^n - x^j + 1) = \ldots?$

There are numerous open questions from this project. We list three, in no particular order:

1. Is there a formula for the resultant of two trinomials similar to Swan's formula for the resultant of two binomials [13]? $\mathrm{res}(x^n - x^k + 1, x^n - x^j + 1) = \ldots$?
2. ~~If n is fixed, is there an absolute upper bound on the size of a clique in $G(n, m)$? For example, is there a largest number of cubic polynomials in $\mathbb{Z}[x]$ that pairwise dyadically resolve, no matter how big their coefficients are?~~ This is not open ... If polynomials dyadically resolve, they must be co-prime mod p for any odd p.

There are numerous open questions from this project. We list three, in no particular order:

1. Is there a formula for the resultant of two trinomials similar to Swan's formula for the resultant of two binomials [13]? $\mathrm{res}(x^n - x^k + 1, x^n - x^j + 1) = \ldots$?
2. ~~If n is fixed, is there an absolute upper bound on the size of a clique in $G(n, m)$? For example, is there a largest number of cubic polynomials in $\mathbb{Z}[x]$ that pairwise dyadically resolve, no matter how big their coefficients are?~~ This is not open ... If polynomials dyadically resolve, they must be co-prime mod p for any odd p.
3. In [5] it was shown that a clique of trinomials of the form $x^n - x^k + 1$ with n fixed has size no more than $2 \log_2 n$. Is this a tight upper bound?

There are numerous open questions from this project. We list three, in no particular order:

1. Is there a formula for the resultant of two trinomials similar to Swan's formula for the resultant of two binomials [13]? $\mathrm{res}(x^n - x^k + 1, x^n - x^j + 1) = \ldots$?
2. ~~If n is fixed, is there an absolute upper bound on the size of a clique in $G(n, m)$? For example, is there a largest number of cubic polynomials in $\mathbb{Z}[x]$ that pairwise dyadically resolve, no matter how big their coefficients are?~~ This is not open ... If polynomials dyadically resolve, they must be co-prime mod p for any odd p.
3. In [5] it was shown that a clique of trinomials of the form $x^n - x^k + 1$ with n fixed has size no more than $2 \log_2 n$. Is this a tight upper bound?
4. ... Your question

[1] Chen B. and Zima E.
Block fermat numbers in modular arithmetic.
Programming and Computer Software, 52(2), 2026.

[2] Coen Bron and Joep Kerbosch.
Algorithm 457: finding all cliques of an undirected graph.
Communications of the ACM, 16(9):575–577, 1973.

[3] Benjamin Chen, Yu Li, and Eugene Zima.
On a two-layer modular arithmetic.
57(3):133–136.

[4] Javad Doliskani, Pascal Giorgi, Romain Lebreton, and Eric Schost.
Simultaneous conversions with the residue number system using linear algebra.
44(3):1–21, 2018.

[5] Robert Dougherty-Bliss, Mits Kobayashi, Natalya Ter-Saakov, and Eugene Zima.
Dyadically resolving trinomials for fast modular arithmetic.
arXiv preprint arXiv:2508.11043, 2025.

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

[7] Aric Hagberg, Pieter J Swart, and Daniel A Schult.
Exploring network structure, dynamics, and function using networkx.
Technical report, Los Alamos National Laboratory (LANL), 2007.

[8] Darrel Hankerson, Scott Vanstone, and Alfred Menezes.
Guide to Elliptic Curve Cryptography.
Springer-Verlag.
OCLC: 1058873580.

[9] R.M. Hewlitt and E.S. Swartzlantler.

Canonical signed digit representation for FIR digital filters.
In 2000 IEEE Workshop on SiGNAL PROCESSING SYSTEMS. SiPS 2000. Design and Implementation (Cat. No.00TH8528), pages 416–426. IEEE.


[10] A. A. Hiasat.
A suggestion for a fast residue multiplier for a family of moduli of the form $(2^n - (2^p \pm 1))$.
47(1):93–102.


[11] Donald E. Knuth.
The Art of Computer Programming, Volume 2: Seminumerical Algorithms.
Addison-Wesley Professional, 2014.


[12] Ryan A Rossi, David F Gleich, and Assefaw H Gebremedhin.
Parallel maximum clique algorithms with applications to network analysis.
SIAM Journal on Scientific Computing, 37(5):C589–C616, 2015.

[13] Richard Swan.
Factorization of polynomials over finite fields.
Pacific Journal of Mathematics, 12(3):1099–1106, 1962.

[14] E. V. Zima and A. M. Stewart.
Cunningham numbers in modular arithmetic.
Programming and Computer Software, 33(2):80–86.