

Newton Polytope-Based Strategy for Finding Small Roots of Multivariate Polynomials

Yansong Feng^{1,2}, Abderrahmane Nitaj³, and Yanbin Pan^{1,2}

¹ Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

² School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
{fengyansong, panyanbin}@amss.ac.cn

³ Normandie Univ, UNICAEN, CNRS, LMNO, 14000 Caen, France abderrahmane.nitaj@unicaen.fr

Abstract. Coppersmith’s method plays an important role in cryptanalysis. By introducing a new tool called Sumsets theory from Additive Combinatorics, we propose a novel strategy for Coppersmith’s method based on Newton polytope. With our novel strategy, we provide the first provable and efficient algorithm for calculating the asymptotic bounds of Coppersmith’s method, which is typically a tedious and non-trivial task. Moreover, our new perspective offers a better understanding of Coppersmith’s method. As a byproduct, we apply our new technique and prove the new heuristic introduced by Meers and Nowakowski at Asiacrypt’23 and improve the cryptanalytic result for the Commutative Isogeny Hidden Number Problem.

Keywords: Coppersmith’s method, Sumsets, Newton polytopes, Additive combinatorics

1 Introduction

In 1996, Coppersmith [Cop96, Cop97] introduced methods to find the small solutions of univariate polynomial modular equation and bivariate polynomial. Since then, Coppersmith’s methods have been extended in several ways, such as [HG01, May03], and have found significant applications in cryptanalysis [BHHG01, BV96, HR23, May02, May03, MNS22, DMH20, TLP17].

The main idea behind Coppersmith’s methods starts with constructing a set G of polynomials sharing common roots with the original polynomial. In general, the coefficients of such polynomials are used to build a lattice \mathcal{L} to be reduced. Roughly speaking, the set of polynomials determines everything about Coppersmith’s methods, such as the bound of the desired root and the running time. Therefore, to improve Coppersmith’s method, the key is to construct a better family of polynomials G .

To find the small roots of a single polynomial equation $g(x_1, \dots, x_k) = 0$ over the integers, or the small solutions of the modular equation $g(x_1, \dots, x_k) \equiv 0 \pmod{M}$, Jochemsz and May [JM06] presented in 2006 a heuristic strategy, known as the Jochemsz-May Strategy, to choose a collection G_m of polynomials $g_{[i_1, \dots, i_k]}(x_1, \dots, x_k)$ satisfying $g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) \equiv 0 \pmod{M^m}$ for a specific integer m . The main idea in the Jochemsz-May strategy is to decrease the order of M in $g_{[i_1, \dots, i_k]}$, which generalizes the work of Blömer and May [BM05] that finds optimal bound for small integer roots of bivariate polynomials. At Asiacrypt’23, Meers and Nowakowski [MN23] proposed Automated Coppersmith, which generalized the Jochemsz-May Strategy from a single polynomial equation to a system of polynomial equations.

The goal of Jochemsz-May Strategy is to achieve better bound of Coppersmith’s method for general polynomial equations and the bound should be determined before constructing the lattice. However, as pointed out in [MN23], it is typically a tedious and non-trivial task to determine the asymptotic upper bounds for Coppersmith’s method and manual analysis has to be performed anew when a new set of polynomials is considered. It seems convoluted to prove the asymptotic bound.

More precisely, Coppersmith’s methods encounter estimating the exponents of the following inequality at the end ⁴, where X_i is the upper bound, to be determined, of the absolute value of root x_i for $i = 1, \dots, k$:

$$\det(\mathcal{L}) < M^{m \dim(\mathcal{L}) - \epsilon}.$$

How to quickly compute these $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ is an unavoidable issue.

⁴ This is just a simplified version, details can be found in Section 2

For some simple polynomials, we can compute them by summation. Taking the modular polynomial equation $f(x_1, x_2) = a_1x_1 + a_2x_2 + C \equiv 0 \pmod{M}$ as an example, the Jochemsz-May Strategy⁵ yields a lattice \mathcal{L} and the corresponding $\dim(\mathcal{L})$ is

$$|\{\lambda \mid \lambda \text{ is a monomial of } f^m\}| = \sum_{i_1=0}^m \sum_{i_2=0}^{m-i_1} 1 = \frac{1}{2}m^2 + \frac{3}{2}m + 1 = \frac{1}{2}m^2 + o(m^2).$$

However, it can be easily seen that it become more and more difficult when dealing with polynomials with more and more variables.

To avoid the significantly tedious and time-consuming manual analysis, Meers and Nowakowski [MN23] heuristically assumed that the functions $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ are polynomials in m . This allows them to select specific values m and then utilize Lagrange interpolation to compute $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ with the help of a computer, although it is still time-consuming usually.

Unfortunately, there is some flaw in their claim. They claim that for a polynomial f with k variables, $\dim(\mathcal{L})$ is a polynomial in m with degree k . Considering the counterexample when $f = x^5 + x + 1$ for $m = 1, 2, \dots$, the corresponding $\dim(\mathcal{L})$ is the number of monomials in f^m , which is 3, 6, 10, 15, 20, 25 and so on. It can be seen that for $m \geq 3$, $\dim(\mathcal{L})$ always satisfies the polynomial $5 * m - 5$ while this does not hold for $m < 3$. In fact, we need the constraint that m is big enough to ensure that $\dim(\mathcal{L})$ is some polynomial in m .

However, even counting the condition that m is sufficiently large, there are still some unsatisfactory aspects of the interpolation method.

First, since we do not know how large m can be to ensure that $\dim(\mathcal{L})$ is a polynomial in m , a natural idea is that we can terminate the interpolation method for some m 's when the corresponding polynomial is stable. However, we find that sometimes the interpolation method may get stuck in local convergence, that is the polynomial will be stable for a long time before the correct polynomial appears, which will certainly result in an incorrect polynomial when a natural termination strategy is employed (see more details in Section 5.2). Hence, the output of the interpolation method seems untrusted.

Second, by the state-of-the-art results in additive combinatorics, for some polynomials f with just 4 variables like the examples in Section 5.2 or Section 5.3, m needs to be greater than 2^{300} in theory in order for $\dim(\mathcal{L})$ to be a polynomial in m [GSW23]. Since m needs to be sufficiently large to ensure that $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ are polynomials on m , it is necessary to compute the values of $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ for large enough m . Hence, the interpolation method must involve the computation of f^m for a large m to make the results provable, which requires a significant amount of time in the worst case, as the number of monomials in the powers of f grows very quickly. Therefore, Automated Coppersmith is still very time-consuming for general polynomials, which is also verified by our experiments in Section 5.

As a consequence, the following natural question arises:

Can we compute $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ more efficiently?

1.1 Our contributions

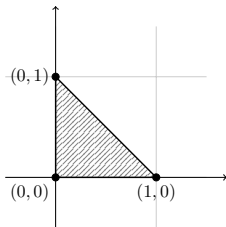
By introducing new tools, we provide a novel way to avoid the tedious computation in Coppersmith's method, which allows us to handle more complicated polynomial equations and achieve better results.

New Tools from Additive Combinatorics We introduce new tools for Coppersmith's method from Additive Combinatorics, including Sumset theory and a series of results for counting integer points in convex hulls. The new perspective based on the new tools will help us clarify further Coppersmith's method and may inspire some novel ideas.

For example, we can reformulate the lattice dimension and determinant computation in Coppersmith's method with the Sumset theory. Note that for the asymptotic upper bound of the roots, only the leading coefficient of these polynomials (if they are) are needed and for the single polynomial f , $\dim(\mathcal{L})$ is the number of monomials in f^m . In 1992, Khovanskii [Kho92] proved that this is indeed polynomial in m when m is big enough and the leading coefficient of $|\text{supp}\{f^m\}|$ is exactly the volume

⁵ Details about the Jochemsz-May Strategy can be found in Section 3.1.

of the convex hull related to f , where $\text{supp}\{f^m\}$ is the set of the monomials of f^m . See Fig. 1 for an example. Hence we can compute the leading coefficient of $\dim(\mathcal{L})(m)$ by computing the volume, which is usually very fast in practice.



$$\begin{aligned} f(x_1, x_2) &= a_1 x_1 + a_2 x_2 + C, \\ \Delta &\text{ is a triangle of } \{(0, 0), (0, 1), (1, 0)\}, \\ S_\Delta &= \frac{1}{2}. \end{aligned}$$

Fig. 1: Newton polytope corresponding to $\text{supp}\{f\} = \{x_1, x_2, 1\}$

Subsequently, researchers investigated how large m needs to be such that the number of monomials in f^m is a polynomial in m . Such explicit results were only previously known in the special cases when the number of variables is k with $k = 1$ [GS20, GW21, Nat72, WCC11], when the convex hull of f is a simplex or when f has $k + 2$ monomials [CG20] until 2023 Granville et al. [GSW23] gave the first effective upper bounds for this threshold for arbitrary f . When f has n monomials, it tends to be at least n^n . It might imply that in certain worst-case, if we use Lagrange interpolation, we would need to compute f^m for very large $m = O(n^n)$. Obviously, Khovanskii's results can help improve the computation of $\dim(\mathcal{L})$ for the Jochemsz-May strategy.

Moreover, we obtain similar results for the calculation of $\det(\mathcal{L})$, and we extend these results to a system of polynomial equations.

Novel strategy based on Newton polytope We also propose a novel Newton polytope-based strategy to choose polynomials for Coppersmith's method. In theory, we rigorously prove that our strategy can achieve the same bound as the Jochemsz-May Strategy, which is currently the optimal strategy used for solving general cases of f .

Compared to the original Jochemsz-May Strategy [JM06, MN23], our new strategy has the following advantages.

- We directly eliminate the time-consuming computation for f^m when computing asymptotic upper bound in Automated Coppersmith for large m , which brings a 1000x~1200x improvement in running time for some polynomials in our experiment, as presented in Section 5. Moreover, our strategy also provides a way to construct the lattice basis without computation for f^m .
- We overcome the issue of getting stuck in local convergence when using interpolation method to compute the bound, which means that the output of our method is reliable.
- Our new method is provable, which gets rid of the heuristic assumptions in [MN23]. In fact, our method also offers a new perspective on understanding Automated Coppersmith, thus we also prove Meers and Nowakowski's Heuristic 2 [MN23].
- Our new method provides an explicit formula for the computation in Coppersmith's method, which is no doubt significant for the further study. The same explicit formula also holds for the Jochemsz-May Strategy. Specifically, suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ has a small root $\mathbf{u} = (u_1, \dots, u_k)$ and known bounds X_j such that $u_j \leq X_j$ for $j = 1, \dots, k$. Let M be the modulus and m a fixed integer. Then, in Coppersmith's method, the basic inequality $\det(L) < M^{m \dim(L)}$ can be written as

$$X_1^{\int_{N(f)} x_1 dV} \dots X_k^{\int_{N(f)} x_k dV} M^{\frac{k}{k+1} \int_{N(f)} 1 dV} < M^{\int_{N(f)} 1 dV},$$

where $N(f)$ is the Newton polytope of f , that is the convex hull of $\{(i_1, \dots, i_k) \mid x_1^{i_1} \dots x_k^{i_k} \text{ is a monomial of } f^m\}$.

As with Meers and Nowakowski in [MN23], we also do not consider the Extended Strategy mentioned in [JM06]. Just Basic Strategy is enough in practice, such as CI-HNP in [MN23]. Moreover, considering whether the Extended Strategy has a result related to Newton polytope is quite challenging, and we leave it as an open problem.

New cryptanalytic result for CI-HNP Based on our strategy, we improve the results for the Commutative Isogeny Hidden Number Problem (CI-HNP) proposed by Meers and Nowakowski in Asi-crypt'23. Additionally, we rigorously prove that their Automated Coppersmith's method is equivalent to the Jochemsz-May Strategy when handling single polynomial equations, which is non-trivial.

Roadmap. The paper is organized as follows: In Section 2 we give some basic preliminaries about Coppersmith's method and newly introduced tools—Sumset Theory. In Section 3, we provide the explicit formula for the Jochemsz-May strategy and our novel strategy when handling a single polynomial equation. Then, in Section 4, we extend this to a system of polynomial equations. As an application, we improve the results of CI-HNP for CSURF, details can be found in Section 5. Moreover, in order to fully demonstrate the efficiency of computing asymptotic upper bounds, we have conducted sufficient experiments, which also can be found in Section 5. Finally, we conclude our work in Section 6.

2 Preliminaries

Let \mathbb{Z} denote the ring of integers and \mathbb{Q} denote the field of rational numbers. We use lowercase bold letters (e.g., \mathbf{v}) for vectors and uppercase bold letters (e.g., \mathbf{A}) for matrices. The notation $\binom{n}{m}$ represents the number of ways to select m items out of n items, which is defined as $\frac{n!}{m!(n-m)!}$. If $m > n$, we set $\binom{n}{m} = 0$. $o(\cdot)$ (Little-o) denotes the upper bounds that cannot be tight.

2.1 Polynomials

Let x_1, \dots, x_k be k variables. Suppose f is a polynomial in $\mathbb{Z}[x_1, \dots, x_k]$, then the polynomial f can be expressed as

$$f(x_1, \dots, x_k) = \sum_{i_1, \dots, i_k \in \mathbb{N}} \alpha_{i_1, \dots, i_k} \cdot x_1^{i_1} \cdot \dots \cdot x_k^{i_k}.$$

Here, $x_1^{i_1} \cdot \dots \cdot x_k^{i_k}$ is termed as a monomial of f if its coefficient $\alpha_{i_1, \dots, i_k} \neq 0$. The set of all monomials of f is denoted as $\text{supp}\{f\}$. The total degree $\text{deg}(f)$ of f is defined as

$$\text{deg}(f) := \max_{\alpha_{i_1, \dots, i_k} \neq 0} (i_1 + \dots + i_k).$$

The following definitions serve to simplify the notations related to multivariate polynomials.

Definition 1 (Monomial Order). Let \mathcal{M} be a set of monomials. A monomial order on \mathcal{M} is a total order \prec that satisfies the following two properties:

1. For every $\lambda \in \mathcal{M}$, it holds that $1 \prec \lambda$.
2. If $\lambda_1 \prec \lambda_2$, then $\lambda \cdot \lambda_1 \prec \lambda \cdot \lambda_2$ for every monomial $\lambda \in \mathcal{M}$.

For example, suppose $x_1 \prec x_2 \prec x_3$, then $x_2^2 \prec x_3$ and $x_1 \prec x_2 \prec x_2^2$ when using the lexicographic monomial order \prec_{lex} . Because lexicographic monomial order (\prec_{lex}) first compares exponents of x_1 in the monomials, and in case of equality compares exponents of x_2 , and so forth.

If \prec is a monomial order, the leading monomial of a polynomial f is the unique monomial λ of f that satisfies $\lambda' \prec \lambda$ for every monomial λ' of f . We denote the leading monomial, and the leading coefficient of the leading monomial of f by $\text{LM}(f)$ and $\text{LC}(f)$ respectively. The leading term of f is denoted $\text{LT}(f)$ and satisfies

$$\text{LT}(f) = \text{LC}(f) \times \text{LM}(f).$$

If $\text{LC}(f) = 1$, then we say that f is a monic polynomial.

Definition 2 (Ideal). Let $\mathcal{F} = \{f_1, \dots, f_n\}$ be a set of polynomials in $\mathbb{Z}[x_1, \dots, x_k]$. The ideal I generated by \mathcal{F} is the set of all linear polynomial combinations of f_1, \dots, f_n , that is

$$I = \{a_1 f_1 + \dots + a_n f_n : a_i \in \mathbb{Z}[x_1, \dots, x_k]\}.$$

If I is an ideal, then the set of all leading terms of the elements of I is denoted $\text{LT}(I)$ and satisfies $\text{LT}(I) = \{\text{LT}(f) | f \in I\}$.

Definition 3. Fix a monomial order, and let I be an ideal. A finite subset $G = \{g_1, \dots, g_r\} \subset I$ is a Gröbner basis for I if

$$\text{LT}(I) = \{\text{LT}(g_1), \dots, \text{LT}(g_r)\}.$$

Before we introduce the Hilbert Function of an ideal I , we need the following definition:

Definition 4. Suppose I is an ideal in $\mathbb{Z}[x_1, x_2, \dots, x_k]$ and then we define $\mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s}$ to be the set of polynomials in $\mathbb{Z}[x_1, x_2, \dots, x_k]$ of total degree $\leq s$, and $I_{\leq s}$ is the set of polynomials in I of total degree $\leq s$. That is,

$$\begin{aligned} \mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s} &= \{f \in \mathbb{Z}[x_1, x_2, \dots, x_k] : \deg(f) \leq s\}, \\ I_{\leq s} &= I \cap \mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s} = \{f \in I : \deg(f) \leq s\}. \end{aligned}$$

Both $\mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s}$ and $I_{\leq s}$ are vector spaces over \mathbb{Z} , with $I_{\leq s}$ exactly being a subspace of $\mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s}$. Now, we are prepared to introduce the Hilbert function.

Definition 5 (Hilbert function). Let I be an ideal in $\mathbb{Z}[x_1, x_2, \dots, x_k]$, and let $I_{\leq s}$ be the space of elements of I of degree at most s . The (affine) Hilbert function $HF(s)$ of I is defined to be the dimension of $\mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s}/I_{\leq s}$ as a vector space over \mathbb{Z} . That is,

$$HF_I(s) = \dim(\mathbb{Z}[x_1, x_2, \dots, x_k]_{\leq s}/I_{\leq s}).$$

There is a useful lemma for Hilbert function, which is called Hilbert's theorem (see [Mum76], Theorem 6.21):

Lemma 6. Let $I \subset \mathbb{Z}[x_1, x_2, \dots, x_k]$ be a proper ideal. Then there exists a polynomial $h(z) \in \mathbb{Q}[z]$ such that $\deg(h) = \dim(I)$, for sufficiently large m ,

$$HF_I(m) = h(m).$$

The polynomial $h(z)$ is often referred to as the Hilbert polynomial of I .

Remark 7. The concepts of Hilbert functions and Hilbert polynomials of graded algebras are crucial in commutative algebra. For more detailed results, please refer to [Sta78].

2.2 Lattices, SVP, and LLL

Let $m \geq 2$ be an integer. A lattice is a discrete additive subgroup of \mathbb{R}^m . A more explicit definition is presented as follows.

Definition 8 (Lattice). Let $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{R}^m$ be n linearly independent vectors with $n \leq m$. The lattice \mathcal{L} spanned by $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is the set of all integer linear combinations of $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$, i.e.,

$$\mathcal{L} = \left\{ \mathbf{v} \in \mathbb{R}^m \mid \mathbf{v} = \sum_{i=1}^n a_i \mathbf{v}_i, a_i \in \mathbb{Z} \right\}.$$

The integer n denotes the rank of the lattice \mathcal{L} , while m represents its dimension. The lattice \mathcal{L} is said to be full rank if $n = m$. We use the matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$, where each vector \mathbf{v}_i contributes a row to \mathbf{B} . The determinant of \mathcal{L} is defined as $\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}\mathbf{B}^t)}$, where \mathbf{B}^t is the transpose of \mathbf{B} . If \mathcal{L} is full rank, this reduces to $\det(\mathcal{L}) = |\det(\mathbf{B})|$.

Definition 9 (Fundamental domain). For a lattice basis $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{R}^m$, the space generated by all real number combinations in $[0, 1)^n$ is called the fundamental domain of the lattice \mathcal{L} . It is denoted as

$$\mathcal{P}(\mathcal{L}) = \left\{ \sum_{i=1}^n a_i \mathbf{v}_i \mid 0 \leq a_i < 1 \right\}.$$

The volume of the fundamental domain \mathcal{P} is equal to the determinant of the lattice, that is $\text{vol}(\mathcal{P}) = \det(\mathcal{L})$.

In lattice theory, numerous hard problems are used to secure several cryptosystems. The Shortest Vector Problem (SVP) is one of them.

Definition 10 (Shortest Vector Problem (SVP)). *Given a lattice \mathcal{L} , the Shortest Vector Problem (SVP) asks to find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}$ of minimum Euclidean norm, i.e., find $\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}$ such that $\|\mathbf{v}\| \leq \|\mathbf{w}\|$ for all non-zero $\mathbf{w} \in \mathcal{L}$.*

Although SVP is NP-hard under randomized reductions [Ajt98], there exist algorithms that can find a relatively short vector, instead of the exactly shortest vector, in polynomial time, such as the famous LLL algorithm proposed by Lenstra, Lenstra, and Lovász [LLL82] in 1982. The following result is useful for our analysis [May03].

Lemma 11 (LLL). *Let \mathcal{L} be a lattice spanned by a basis $(\mathbf{u}_1, \dots, \mathbf{u}_\omega)$. In polynomial time, the LLL algorithm finds a new basis $(\mathbf{v}_1, \dots, \mathbf{v}_\omega)$ of \mathcal{L} satisfying*

$$\|\mathbf{v}_1\| \leq \dots \leq \|\mathbf{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\mathcal{L})^{\frac{1}{\omega+1-i}},$$

for $i = 1, \dots, \omega$.

2.3 Sumsets theory

For any given finite subset A of an abelian group G , suppose $0 \in A$, we consider the sumset $mA := \{a_1 + a_2 + \dots + a_m : a_i \in A\}$. Khovanskii's 1992 theorem [Kho92] states that if $A \subset \mathbb{Z}^k$ is finite, then there exists $p(x) \in \mathbb{Q}[x]$ of degree k and $N_{\text{Kh}}(A)$ such that if $m \geq N_{\text{Kh}}(A)$, then $|mA| = p(m)$. Moreover, if the difference set $A - A$ generates all of \mathbb{Z}^k additively, then $\deg(p) = k$ and the leading coefficient of p is the volume of the convex hull of A , which we define as $N = H(A)$.

To make things more straightforward, we introduce the Newton polytope.

Definition 12 (Set of points). *Let $G = \mathbb{Z}^k$. For a polynomial f , consider $A(f)$ as the set of points corresponding to the monomials of f as follows:*

$$A(f) = \{(i_1, \dots, i_k) | x_1^{i_1} \cdot \dots \cdot x_k^{i_k} \text{ is a monomial of } f\}.$$

Definition 13 (Newton polytope). *Let f be a polynomial in $\mathbb{Z}[x_1, \dots, x_k]$. The Newton polytope $N(f)$ of f is defined as the convex hull of $A(f)$*

Obviously, the Newton polytope has the following property:

Property 14. For all polynomials f_1, f_2 in $\mathbb{Z}[x_1, \dots, x_k]$, it holds that

$$N(f_1 \cdot f_2) = N(f_1) + N(f_2).$$

Definition 15 (Saturated Newton polytope). *We say that a polynomial f has Saturated Newton Polytope if every integer point of the convex hull of its exponent vectors corresponds to a monomial of f .*

For example, when $\text{supp}\{f\} = \{x_1^2, x_1, x_2, 1\}$, $A(f)$ is $\{(0, 0), (1, 0), (2, 0), (0, 1)\}$, corresponding to $\{1, x, x^2, y\}$ and the Newton Polytope of f is a triangle with $\{(0, 0), (2, 0), (0, 1)\}$. Then $|mA(f)|$ corresponds to $\text{supp}\{f^m\}$. For simplicity, we write $A(f)$ as A and the convex hull of A as N . So Khovanskii's 1992 theorem [Kho92] can be stated as follows:

Lemma 16 (Khovanskii, [Kho92]). *Suppose $A \subset \mathbb{Z}^k$ is finite. Then there exists a value N_{kh} with the following property: if $m > N_{kh}$, then there exists a polynomial $p(x) \in \mathbb{Q}[x]$ such that $|mA| = p(m)$.*

Khovanskii proved this by constructing a finitely generated graded module M over the polynomial ring $\mathbb{C}[t_1, \dots, t_s]$, where the cardinality of set A is denoted by s . This module possesses the characteristic that its homogeneous component M_m forms a vector space over \mathbb{C} with precisely m dimensions for all $m \geq 1$. Therefore, the dimension of M_m over \mathbb{C} is exactly the Hilbert Function. According to Hilbert's theorem, the dimension of M_m over \mathbb{C} is a polynomial in m for sufficiently large m , thereby yielding the desired result.

Suppose $A \subset \mathbb{Z}^k$ is full rank, which means there exist $\mathbf{v}_1, \dots, \mathbf{v}_k$ that are linearly independent. We denote the linear space spanned by A over \mathbb{Z} as $\text{span}_{\mathbb{Z}}(A)$, that is

$$\text{span}_{\mathbb{Z}}(A) = \{a_1 \mathbf{v}_1 + \dots + a_k \mathbf{v}_k | a_1, \dots, a_k \in \mathbb{Z}\}.$$

We denote the convex hull of A as N and its volume as $V(N)$. Moreover, if $\text{span}_{\mathbb{Z}}(A) = \mathbb{Z}^k$ additively, then $\deg(p) = k$ and the leading coefficient of p is $V(N)$.

When $\text{span}_{\mathbb{Z}}(A) \neq \mathbb{Z}^k$, we denote $[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A)]$ as the index of $\text{span}_{\mathbb{Z}}(A)$ over \mathbb{Z}^k . Actually, if we view $\text{span}_{\mathbb{Z}}(A)$ as a lattice over \mathbb{Z} , then $[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A)]$ is equal to the fundamental domain of $\text{span}_{\mathbb{Z}}(A)$.

See Fig 2 as an example, where $f = a_1x_1 + a_2x_2 + C$. We have $\text{supp}\{f\} = \{x_1, x_2, 1\}$. Now it holds that

$$A(f) = \{(1, 0), (0, 1), (0, 0)\}$$

and

$$\text{span}_{\mathbb{Z}}(A) = \{(1, 0)z_1 + (0, 1)z_2 \mid z_1, z_2 \in \mathbb{Z}\}$$

is a lattice over \mathbb{Z} . Then the fundamental domain of $\text{span}_{\mathbb{Z}}(A)$ is a unit square and $[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A)] = 1$

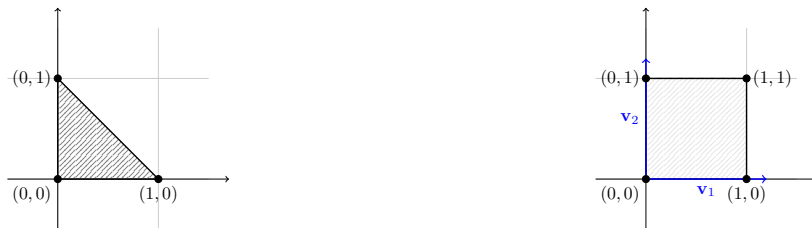


Fig. 2: $A(f)$ and $\mathcal{P}(f)$: we can see that $A(f)$ is a triangle and the fundamental domain of $\text{span}_{\mathbb{Z}}(A)$ is a unit square.

Then we can rewrite Corollary 2 in [Kho92] as follows:

Lemma 17. *Suppose $A \subset \mathbb{Z}^k$ is full rank. Then there exists a value N_{kh} with the following property: if $m > N_{kh}$, then there exists a polynomial $p(x) \in \mathbb{Q}[x]$ of degree k such that $|mA| = p(m)$ and the leading coefficient of p is $V(N)/[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A)]$, where N is the convex hull of A , and $V(N)$ is its volume.*

Regarding the size of N_{kh} , in 2023, Granville et al. [GSW23] provided the first effective upper bounds for this threshold for arbitrary A . For any such A in terms of the width of A , $w(A) = \text{width}(A) := \max_{a_1, a_2 \in A} \|a_1 - a_2\|_{\infty}$. Then the upper bound proposed by Granville et al. is as follows:

Lemma 18 (Granville et al., [GSW23]). *If $A \subset \mathbb{Z}^k$ is finite, then $|mA| = p(m)$ for all $m \geq (2|A| \cdot \text{width}(A))^{(k+4)|A|}$.*

We note that the former upper bound is too large. When $|A| = n$, it tends to be at least n^n !

Finally, we give another useful theorem called Ehrhart Theorem proposed by Ehrhart in 1962 [Ehr62].

Lemma 19 (Ehrhart, [Ehr62]). *Suppose $A \subset \mathbb{Z}^k$ is full-rank and N is the convex hull of A , then the number of integer points contained in the polytope mN is a polynomial of m with degree k . Moreover, the leading coefficient is $V(N)$.*

It was later generalized by Michel Brion and Michèle Vergne [BV97] into the following result.

Lemma 20 (Brion and Vergne, [BV97]). *Let N be a convex integer polytope with vertices in \mathbb{Z}^k and ϕ be any homogeneous polynomial function. Then the following counting function:*

$$\sum_{\delta \in mN \cap \mathbb{Z}^k} \phi(\delta)$$

is a polynomial of m with degree $k + \deg(\phi)$. Moreover, the leading coefficient is the integral of ϕ over the polytope N .

Now Lemma 19 can be seen as a special case of Lemma 20 when $\phi = 1$.

Considering A as a Saturated Newton polytope, the number of integer points in mN is not less than that in mA as $A \subset N$. However, by comparing Lemma 17 and Lemma 19, we can see that if we view the number of integer points in mN and mA as polynomials in m , the leading terms of these two polynomials are the same, both equal to $V(N)$.

2.4 Coppersmith's method

Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M . Here, a small root means $|u_i| < X_i$ for known bound X_i , for $i = 1, \dots, k$. To find such a root, Coppersmith's method is usually employed. Below we will give a brief introduction to Coppersmith's method for solving modular equations. More details can be found in [May03].

Coppersmith's method first constructs a lattice \mathcal{L} with the coefficient vector of a system of polynomials that has the same small root \mathbf{u} of f when modulo M^m where m is some positive integer. For example, the polynomials can be selected as:

$$g_{[i_1, \dots, i_k]} = x_1^{i_1} \cdot \dots \cdot x_k^{i_k} f^\ell M^{m-\ell}, \text{ for } \ell = 0, \dots, m.$$

Note that each $g_{[i_1, \dots, i_k]}$ has the same small root of f modulo M^m .

Coppersmith's method tries to find the short vectors, or equivalently, the short polynomials g_1, \dots, g_k , in the lattice \mathcal{L} by applying the LLL algorithm. Using the following result, due to Howgrave-Graham, and Lemma 11, we just need $\det(\mathcal{L}) < M^{m \dim(\mathcal{L})}$ to ensure that g_1, \dots, g_k have the same small root \mathbf{u} with f , not only modulo M^m but also over \mathbb{Z} .

Lemma 21 (Howgrave-Graham [HG97]). *Let $g(x_1, \dots, x_k) \in \mathbb{Z}[x_1, \dots, x_k]$ be a polynomial with at most ω monomials. Let M be a positive integer. If there exist k integers (u_1, \dots, u_k) satisfying the following two conditions:*

1. $g(u_1, \dots, u_k) \equiv 0 \pmod{M}$,
2. *there exist k positive integers X_1, \dots, X_k such that $|u_i| < X_i$ for $i = 1, \dots, k$, and $\|g(x_1 X_1, \dots, x_k X_k)\| < \frac{M}{\sqrt{\omega}}$,*

then $g(u_1, \dots, u_k) = 0$ holds over \mathbb{Z} .

Lastly, Coppersmith's method computes the desired root $\mathbf{u} = (u_1, \dots, u_k)$ by solving the system of polynomial equations $g_i(x_1, \dots, x_k) = 0$ for $i = 1, \dots, k$.

In the multivariate scenario, that is, for $k > 1$, we usually assume the ideal generated by g_1, \dots, g_k being zero-dimensional, allowing us to compute the small root \mathbf{u} by Gröbner basis. The following Assumption 1 is widely used in many works [May03, MR09, MNS21, MNS22, MN23, FNP24].

Assumption 1 *The ideal generated by the polynomials obtained from Coppersmith's method is zero-dimensional.*

3 Newton polytope-based strategy

In this section, we propose our Newton polytope-based strategy and provide the corresponding explicit formulas for $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ due to fruitful results in Additive Combinatorics [Ehr62, Kho92, Nat23, GSW23]. Based on the results proposed by Ehrhart [Ehr62] and Khovanskii [Kho92], we also prove that our strategy achieves the same bound as the Jochemsz-May strategy. Thus, the Jochemsz-May strategy and ours share the same explicit formulas. Note this strategy cannot simply be seen as a rewrite of the Jochemsz-May strategy on the Newton polytope. Actually, our strategy introduces some simplifications to the lattice construction.

In Section 3.1, we first recall the widely used Jochemsz-May Strategy [JM06] in Coppersmith's method, which is the optimal strategy for the general case. Then we propose our novel Newton Polytope-Based Strategy in Section 3.2. It has two appealing properties: simpler lattice construction and faster asymptotic bound computation. We will show them in Section 3.3 and Section 3.4. For simplicity, we assume f satisfies $[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A(f))] = 1$, otherwise, the result needs to be divided by $[\mathbb{Z}^k : \text{span}_{\mathbb{Z}}(A(f))]$.

3.1 Jochemsz-May Strategy and Meers-Nowakowski Strategy ⁶

Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M . Here a small root means we assume that we know an upper bound for the root, namely

⁶ The Jochemsz-May strategy can be viewed as the Meers-Nowakowski Strategy (Automated Coppersmith's method) [MN23] in the case $\mathbf{n} = \mathbf{1}$.

$|u_j| < X_j$ for some given X_j . The key step in Coppersmith's method is to construct a lattice \mathcal{L} with the coefficient vector of a system of polynomials as follows:

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = x_1^{i_1} \cdot \dots \cdot x_k^{i_k} f^\ell M^{m-\ell}, \text{ for } \ell = 0, \dots, m. \quad (1)$$

Note that each $g_{[i_1, \dots, i_k]}$ has the same small root of f modulo M^m , where m is some positive integer.

At Asiacrypt'06, Jochemsz and May [JM06] described a strategy to choose polynomials in Equation (1) to make $\det(\mathcal{L})$ as small as possible. This remains the best available strategy for the general problem. They defined the following set

$$J_\ell = \{\lambda | \lambda \in \text{supp}\{f^m\} \text{ and } \frac{\lambda}{\text{LM}(f)^\ell} \in \text{supp}\{f^{m-\ell}\}\}$$

and then define the shift polynomials as follows

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = \frac{x_1^{i_1} \cdot \dots \cdot x_k^{i_k}}{\text{LM}(f)^\ell} f^\ell M^{m-\ell}, \quad (2)$$

for $\ell = 0, \dots, m$, and $x_1^{i_1} \dots x_k^{i_k} \in J_\ell \setminus J_{\ell+1}$.

The lattice \mathcal{L} is constructed by taking the coefficient vectors of the polynomials $g(x_1 X_1, \dots, x_n X_n)$ as a basis.

Recently, Meers and Nowakowski proposed the Automated Coppersmith's method, which can be seen as a generalization of the Jochemsz-May Strategy from a single polynomial equation to a system of polynomial equations. However, the Meers-Nowakowski Strategy and the Jochemsz-May Strategy still have a slight difference in the single polynomial equation case. In the Meers-Nowakowski Strategy, they choose

$$J'_\ell = \{\lambda | \lambda \in \text{supp}\{f^m\} \text{ and } \frac{\lambda}{\text{LM}(f)^\ell} f^\ell \in \text{supp}\{f^m\}\}.$$

These two strategies can be seen as greedy strategies, aiming to reduce the order of M in Equation (2) as much as possible.

Although J_ℓ and J'_ℓ appear somewhat different, nevertheless, we will prove in Appendix A that this difference is essentially negligible. In this section, we only consider the case of a single polynomial equation. When comparing with previous work, we only just compare with the Jochemsz-May Strategy.

3.2 Newton polytope-based strategy

Now we propose our novel Newton polytope based strategy. For simplicity, we assume that in the following, f is monic and the lattice spanned by the points corresponding to all monomials of f is \mathbb{Z}^k . Otherwise, we need to divide by the volume of the lattice.

Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M , the leading monomial of f corresponds to the integer point $\alpha \in \mathbb{Z}^k$. Here, a small root means $|u_j| < X_j$ for known bound X_j , for $j = 1, \dots, k$. We also focus on selecting the basis of the lattice used in Coppersmith's method. Our algorithm is described as follows:

Fix an integer m : Consider $A(f) = \{(i_1, \dots, i_k) | x_1^{i_1} \cdot \dots \cdot x_k^{i_k} \text{ is a monomial of } f\}$ and its convex hull $N(f)$, then compute $mN(f)$.

Define the sets

$$S_\ell = \ell\alpha + (m - \ell)N(f),$$

so that $S_m \subset \dots \subset S_0$. For each integer point (i_1, \dots, i_k) in $mN(f)$, compute ℓ such that $(i_1, \dots, i_k) \in S_\ell$ but $(i_1, \dots, i_k) \notin S_{\ell+1}$. Record

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = \frac{x_1^{i_1} \cdot \dots \cdot x_k^{i_k}}{\text{LM}(f)^\ell} f^\ell M^{m-\ell}.$$

Use the coefficient vectors of all $g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)$ to form the basis of the lattice.

We provide Algorithm 1 to better understand our novel strategy. Compared with Jochemsz-May Strategy, we do not need to compute the monomials in f^m precisely. This leads to an efficient construction of the lattice.

It is worth mentioning that, due to the introduction of new tools from additive combinatorics, our proposed strategy can quickly compute asymptotic bounds.

Algorithm 1: Construction of the lattice in Coppersmith's method

Input: $f \in \mathbb{Z}[x_1, \dots, x_k]$, integer m , modulus M , and bounds X_j for $j = 1, \dots, k$
Output: Set of polynomials, whose coefficient vector forms the basis of the lattice \mathcal{L}

- 1 Define $\alpha \in \mathbb{Z}^k$ is the integer point correspond to $\text{LM}(f)$;
- 2 Compute Newton polytope $N(f)$ of f ;
- 3 $G \leftarrow \emptyset$;
- 4 **for** $(i_1, \dots, i_k) \in mN(f)$ **do**
- 5 $\ell \leftarrow 0$;
- 6 **while** $(i_1, \dots, i_k) \in \ell\alpha + (m - \ell)N(f)$ **do**
- 7 $\ell \leftarrow \ell + 1$;
- 8 **end**
- 9 $g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) \leftarrow \frac{x_1^{i_1} \dots x_k^{i_k}}{\text{LM}(f)^\ell} f^\ell M^{m-\ell}$;
- 10 $G \leftarrow G \cup \{g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)\}$;
- 11 **end**
- 12 **return** G ;

3.3 Computing $\dim(\mathcal{L})$ efficiently

Next, we will associate the leading coefficient of $\dim(\mathcal{L})$ with the volume of the convex hull by analyzing the Hilbert Function of some graded algebra. Therefore, we only need to compute the volume of the convex hull to obtain the desired value, which is a very fast operation.

For example, when we consider a modular polynomial equation $f \equiv 0 \pmod{M}$ with $\text{supp}\{f\} = \{x_1, x_2, 1\}$, the computation about $\dim(\mathcal{L})$ is

$$\sum_{i_1=0}^m \sum_{i_2=0}^{m-i_1} 1 = \frac{1}{2}m^2 + \frac{3}{2}m + 1.$$

For calculating the asymptotic bounds of Coppersmith's method, we only need the leading coefficient $1/2$ for the asymptotic bound in Coppersmith's method. This can be computed easily by calculating the volume of the Newton polytope of f , that is, a triangle with vertices at $(0, 0)$, $(0, 1)$, and $(1, 0)$.

According to our strategy in Algorithm 1, for each integer point (i_1, \dots, i_k) in $mN(f)$, we selected a corresponding polynomial $g_{[i_1, \dots, i_k]}$, and then constructed the lattice \mathcal{L} using the coefficient vectors of all $g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)$. Therefore, the dimension of our lattice is equal to the number of integer points in $mN(f)$. Next, we present a formal theorem regarding the calculation of $\dim(\mathcal{L})$.

Theorem 22. *Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. The dimension of the lattice obtained by Algorithm 1 is a polynomial in m , and the leading coefficient is the volume of the Newton polytope corresponding to f . That is,*

$$\dim(\mathcal{L}) = V(N(f))m^k + o(m^k).$$

Proof. The dimension of the lattice is the number of integer points in $mN(f)$. Therefore, by Lemma 19, we know that there exists a degree k polynomial p_E such that $\dim \mathcal{L} = p_E(m)$, and $\text{LM}(p_E) = V(N(f))$. \square

In fact, we find that, not only our strategy can compute $\dim(\mathcal{L})$ in such an elegant way, but the Jochemsz-May Strategy also has a similar result.

Theorem 23. *Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. There exists N_{kh} such that for $m > N_{kh}$, the dimension of the lattice obtained by the Jochemsz-May Strategy is a polynomial in m , and the leading coefficient is the volume of the Newton polytope corresponding to f . That is,*

$$\dim(\mathcal{L}) = V(N(f))m^k + o(m^k).$$

Proof. The dimension of the lattice is the number of integer points in $mN(f)$. Therefore, by Lemma 17, there exists N_{kh} and a degree k polynomial p_K , such that for $m > N_{kh}$, it holds that $\dim(\mathcal{L}) = p_K(m)$, and the leading term of p_K is $\text{LM}(p_K) = V(N(f))$. \square

Comparison with the Jochemsz-May Strategy. By comparing the theorems for calculating $\dim(\mathcal{L})$ in both our strategy (Theorem 22) and Jochemsz-May Strategy (Theorem 23), we find that if we view $\dim(\mathcal{L})$ as a polynomial in m , the leading terms of these two polynomials are the same. However, Jochemsz-May Strategy involves a more precise calculation of mA , which results in a greater computational burden.

3.4 Computing $\det(\mathcal{L})$ efficiently

Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M . Here, a small root means $|u_j| < X_j$ for known bound X_j , for $j = 1, \dots, k$. Recall we take the coefficient vectors of $g(x_1X_1, \dots, x_nX_n)$ to construct the lattice \mathcal{L} in Algorithm 1. Then we can write $\det(\mathcal{L})$ as

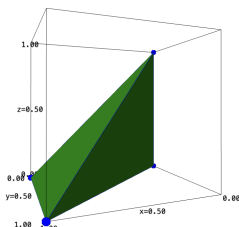
$$\det(\mathcal{L}) = X_1^{p_1(m)} \cdot \dots \cdot X_k^{p_k(m)} M^{p_{\mathcal{F}}(m)}.$$

We will divide the discussion into three parts regarding $\det(\mathcal{L})$. In the first two parts, we show that we can compute the leading coefficients of p_j for $j = 1, \dots, k$ and $p_{\mathcal{F}}$ in a manner similar to how we compute $\dim(\mathcal{L})$. Finally, we will show that the leading coefficient of $\det(\mathcal{L})$ in our strategy is the same as that in the Jochemsz-May Strategy, which can also be viewed as a proof of Heuristic 2 (for $n = 1$) in [MN23]. Therefore, the asymptotic bounds computed by both strategies are the same, which demonstrates that our strategy can achieve the same effectiveness as the Jochemsz-May Strategy.

Computing the leading coefficient of p_j . Considering the computation of p_j , the main idea is to transform p_j into an integral or a higher-dimensional Newton polytope. For example, we choose f with $\text{supp}\{f\} = \{x_1, x_2, 1\}$. If we want to directly compute p_1 , we need to compute

$$p_1(m) = \sum_{i_1=0}^m \sum_{i_2=0}^{m-i_1} i_1 = \frac{1}{6}m^3 + o(m^3).$$

However, we can use the following method to compute $\text{LC}(p_1)$. Now we consider about the convex hull of $\{(0, 0, 0), (0, 1, 0), (1, 0, 0), (1, 0, 1)\}$ as in Figure 3. Note that the volume of this tetrahedron is equal to the integral of x_1 over $N(f)$.



A tetrahedron of $\{(0, 0, 0), (0, 1, 0), (1, 0, 0), (1, 0, 1)\}$,

$$\text{Volume of tetrahedron} = \frac{1}{6} \cdot 1 \cdot 1 \cdot 1 = \frac{1}{6},$$

$$\text{LC}(p_1) = \frac{1}{6}.$$

Fig. 3: $\text{supp}\{f\} = \{x_1, x_2, 1\}$

Our formal theorem regarding the calculation of p_j is as follows.

Theorem 24. *Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. Then p_j in $\det(\mathcal{L})$ obtained by Algorithm 1 is a polynomial in m , and the leading coefficient is the integral of x_j over the Newton polytope corresponding to f . That is,*

$$p_j(m) = \int_{N(f)} x_j dV m^{k+1} + o(m^{k+1}).$$

Proof. Regarding the lattice constructed in our strategy, we can write p_j as follows:

$$p_j(m) = \sum_{\delta \in mN(f) \cap \mathbb{Z}^k} x_j(\delta), \tag{3}$$

Here x_j is a homogeneous polynomial. When $\delta = (i_1, \dots, i_k)$, it holds that $x_j(\delta) = i_j$.

Therefore, by Lemma 20, we know that $p_j(m)$ is a degree $k + 1$ polynomial and $\text{LM}(p_j) = \int_{N(f)} x_j dV$. \square

Computing the leading coefficient of $p_{\mathcal{F}}$. Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M , the leading monomial of f correspond to the integer point $\alpha \in \mathbb{Z}^k$. Here, a small root means $|u_j| < X_j$ for known bound X_j , for $j = 1, \dots, k$. In our novel Newton polytope-based strategy, we define the sets

$$S_\ell = \ell\alpha + (m - \ell)N(f),$$

satisfied $S_m \subset \dots \subset S_0$. For each integer point (i_1, \dots, i_k) in $mN(f)$, compute ℓ such that $(i_1, \dots, i_k) \in S_\ell$ but $(i_1, \dots, i_k) \notin S_{\ell+1}$. Recall that

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = \frac{x_1^{i_1} \cdots x_k^{i_k}}{\text{LM}(f)^\ell} f^\ell M^{m-\ell},$$

and then use the coefficient vectors of all $g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)$ to form the basis of the lattice.

Define $T_m = \{m\alpha\}$ and T_ℓ for $\ell = 0, \dots, m - 1$ as follows:

$$T_\ell = S_\ell \setminus S_{\ell+1}.$$

Therefore, we can write $p_{\mathcal{F}}(m)$ as follows:

$$p_{\mathcal{F}}(m) = m \dim(\mathcal{L}) - \sum_{\ell=0}^m \ell |T_\ell \cap \mathbb{Z}^k|.$$

Before presenting the theorem for calculating $p_{\mathcal{F}}$, we first give a lemma for counting the sum of the number of integer points in S_ℓ .

Lemma 25. For $f \in \mathbb{Z}[x_1, \dots, x_k]$, $\sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| = V(N(f))m^{k+1} + o(m^{k+1})$.

Proof. From Lemma 19, we know that there exists a polynomial $p_E(m)$ such that $|mN(f) \cap \mathbb{Z}^k| = p_E(m)$, and the leading coefficient is $V(N(f))$. Therefore, we have

$$\begin{aligned} \sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| &= \sum_{\ell=0}^m |(m - \ell)N(f) \cap \mathbb{Z}^k| \\ &= \sum_{\ell=0}^m |\ell N(f) \cap \mathbb{Z}^k| \\ &= \sum_{\ell=0}^m V(N(f)) \ell^k + o(m^{k+1}) \\ &= \frac{1}{k+1} V(N(f)) m^{k+1} + o(m^{k+1}). \end{aligned}$$

□

Therefore, we can compute $p_{\mathcal{F}}$ as in the following theorem:

Theorem 26. Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. Then the exponent $p_{\mathcal{F}}$ of M in $\det(\mathcal{L})$ obtained by Algorithm 1 is a polynomial in m , and its leading coefficient is $kV(N(f))/(k+1)$. That is,

$$p_{\mathcal{F}}(m) = \frac{k}{k+1} V(N(f)) m^{k+1} + o(m^{k+1}).$$

Proof. As $p_{\mathcal{F}}(m) = m \dim(\mathcal{L}) - \sum_{\ell=0}^m \ell |T_\ell \cap \mathbb{Z}^k|$ and $T_\ell = S_\ell \setminus S_{\ell+1}$ for $\ell = 0, \dots, m - 1$, we can rewrite $p_{\mathcal{F}}$ as follows:

$$p_{\mathcal{F}}(m) = m \dim(\mathcal{L}) - m |S_m \cap \mathbb{Z}^k| - \sum_{\ell=0}^{m-1} \ell (|S_\ell \cap \mathbb{Z}^k| - |S_{\ell+1} \cap \mathbb{Z}^k|). \quad (4)$$

Then Equation (4) holds that

$$p_{\mathcal{F}}(m) = m \dim(\mathcal{L}) - \sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k|.$$

Using Theorem 22 and Lemma 25, we have

$$m \dim(\mathcal{L}) = V(N(f)) m^{k+1} + o(m^{k+1}),$$

and

$$\sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| = \frac{1}{k+1} V(N(f)) m^{k+1} + o(m^{k+1}).$$

Therefore, the exponent $p_{\mathcal{F}}$ in $\det(\mathcal{L})$ obtained by Algorithm 1 is a polynomial in m , and the leading coefficient is $V(N(f)) - V(N(f))/(k+1) = kV(N(f))/(k+1)$, that is

$$p_{\mathcal{F}}(m) = \frac{k}{k+1} V(N(f)) m^{k+1} + o(m^{k+1}).$$

□

Comparison with the Jochemsz-May Strategy. One may ask whether our strategy can achieve the same bounds as the Jochemsz-May Strategy. Next, we will prove that our strategy can achieve the same bounds, i.e., the leading coefficients of both $\dim(\mathcal{L})$ and $\det(\mathcal{L})$ are the same.

Define $\Delta(f) = (mN(f) \cap \mathbb{Z}^k) \setminus mA(f)$. In Section 3.3, we have already proven that the difference between our strategy and the Jochemsz-May Strategy lies in $\Delta(f)$. Comparing Theorem 22 with Theorem 23, this part has been shown to be a very small portion, i.e., $o(m^k)$. Therefore, the leading coefficient of $\dim(\mathcal{L})$ in our strategy is the same as that in the Jochemsz-May Strategy, which is the volume of $N(f)$.

Considering $\det(\mathcal{L})$, specifically the leading coefficients of p_j and $p_{\mathcal{F}}$, we discuss this in two parts.

For p_j , we need to show that the difference between our p_j and the p_j in the Jochemsz-May Strategy, that is

$$\sum_{\delta \in mN(f) \cap \mathbb{Z}^k} x_j(\delta) - \sum_{\delta \in mA(f)} x_j(\delta) = \sum_{\delta \in \Delta(f)} x_j(\delta),$$

is within $o(m^{k+1})$. Note that $x_j(\delta) \leq \deg(f)m$, thus $\sum_{\delta \in \Delta(f)} x_j(\delta)$ belongs to $o(m^{k+1})$. Hence, the leading coefficient of our p_j is the same as in the Jochemsz-May Strategy.

For $p_{\mathcal{F}}$, for $(i_1, \dots, i_k) \in \Delta(f)$, the order of M in $g_{[i_1, \dots, i_k]}$ is less than m . Therefore, the difference in $p_{\mathcal{F}}$ between our strategy and the Jochemsz-May Strategy is less than $m \cdot |\Delta(f)|$, which is within $o(m^{k+1})$. As a result, the leading coefficient of $p_{\mathcal{F}}$ is also the same in both strategies.

In summary, our strategy can achieve the same bounds as the Jochemsz-May Strategy.

4 Generalization to a system of equitons

The Jochemsz-May Strategy [JM06] is designed for the case of a single polynomial equation. However, in cryptanalysis, it is sometimes necessary to handle a system of polynomial equations, such as in the Implicit Factorization Problem [MR09] and the Modular Inversion Hidden Number Problem [XSH⁺19]. Recently, Meers et al. [MN23] introduced the Commutative Isogeny Hidden Number Problem at Asiacrypt'23, where solving this problem is reduced to solving a system of equations.

Next, we will generalize our Newton polytope-based strategy to handle a system of polynomial equations. Using summation techniques for proving some combinatorial identities, we will also provide similar results for calculating asymptotic bounds as in the single polynomial equation case in Section 3. Finally, we will compare our strategy with the Meers-Nowakowski Strategy (the Automated Coppersmith method) proposed by Meers and Nowakowski [MN23]. They proposed a heuristic and used Lagrange interpolation to compute asymptotic bounds, whereas our approach computes these bounds faster and is provable. As an application, we improve the results for the CSURF problem proposed by Meers and Nowakowski in Asiacrypt'23 [MN23]. The details about the improvement of CI-HNP will be provided in Section 5.

4.1 Newton polytope-based strategy

There is more than one way to extend to multiple polynomial equations. Meers et al. [MN23] proposed one approach in Asiacrypt'23, that is the Meers-Nowakowski Strategy (the Automated Coppersmith method), but here we choose an alternative method, which allows us to improve the result of CI-HNP over CSURF proposed by Meers and Nowakowski in Asiacrypt'23 [MN23].

Suppose $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$ have a common small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M , $\alpha_j \in \mathbb{Z}^k$ are the integer points corresponding to the leading monomial of f_j for $j = 1, \dots, n$. Here, a small root means $|u_j| < X_j$ for known bound X_j , for $j = 1, \dots, k$. Our strategy is described as follows:

Fix an integer m . Consider

$$A = \bigcup_{j=1}^n A(f_j)$$

and denote its convex hull N , then compute mN .

Define the sets

$$S_\ell = (m - \ell)N(f) + \bigcup_{\substack{\ell_1 + \dots + \ell_n = \ell \\ 0 \leq \ell_j \leq m}} \sum_{j=1}^n \ell_j \alpha_j, \quad (5)$$

so that $S_m \subset \dots \subset S_0$. For each integer point (i_1, \dots, i_k) in $mN(f)$, compute ℓ such that $(i_1, \dots, i_k) \in S_\ell$ but $(i_1, \dots, i_k) \notin S_{\ell+1}$. Then there exist ℓ_1, \dots, ℓ_n with $\sum_{j=1}^n \ell_j = \ell$, such that

$$(i_1, \dots, i_n) \in (m - \ell)N(f) + \sum_{j=1}^n \ell_j \alpha_j.$$

Recall that

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = \frac{x_1^{i_1} \cdots x_k^{i_k}}{\prod_{j=1}^n \text{LM}(f_j)^{\ell_j}} \prod_{j=1}^n f_j^{\ell_j} M^{m-\ell}.$$

Use the coefficient vectors of all $g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)$ to form the basis of the lattice.

4.2 Compute $\dim(\mathcal{L})$ efficiently

In our strategy, we selected a set of polynomial $g_{[i_1, \dots, i_k]}$, and then constructed the lattice \mathcal{L} using the coefficient vectors of all $g_{[i_1, \dots, i_k]}(x_1 X_1, \dots, x_k X_k)$.

For $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$, let A be the union of all $A(f_j)$ and N be its convex hull. Then, the dimension of the lattice is the number of integer points in mN . This means that the case of multiple polynomial equations can be reduced to a single one. The formal theorem for calculating $\dim(\mathcal{L})$ is as follows.

Theorem 27. *Suppose $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. Let A be the union of all $A(f_j)$ and N be its convex hull. The dimension of the lattice obtained by the Newton polytope-based strategy is a polynomial in m , and the leading coefficient is the volume of N . That is,*

$$\dim(\mathcal{L}) = V(N)m^k + o(m^k).$$

Proof. The dimension of the lattice is the number of integer points in mN . Therefore, by Lemma 19, we know that there exists a degree k polynomial p_E such that $\dim \mathcal{L} = p_E(m)$, and $\text{LM}(p_E) = V(N(f))$. \square

In fact, A corresponds to the set of all monomials of f_j for $j = 1, \dots, n$. Informally speaking, A can be written as $A(f_1 + \dots + f_n)$.

4.3 Compute $\det(\mathcal{L})$ efficiently

Suppose $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$ have a common small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M , which satisfies $|u_j| < X_j$ for known bounds X_j , for $j = 1, \dots, k$.

Let the lattice \mathcal{L} be constructed by our novel strategy. Then we can write $\det(\mathcal{L})$ as

$$\det(\mathcal{L}) = X_1^{p_1(m)} \cdots X_k^{p_k(m)} M^{p_{\mathcal{F}}(m)}.$$

We will discuss the computation of the leading coefficients of p_j and $p_{\mathcal{F}}$ in the following two parts.

Computing the leading coefficient of p_j . Similar to $\dim(\mathcal{L})$, we can reduce the case of multiple polynomial equations to that of a single one. The formal theorem is as follows.

Theorem 28. *Suppose $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. Let A be the union of all $A(f_j)$ and N be its convex hull. Then p_j of the lattice obtained by the Newton polytope-based strategy is a polynomial in m , and the leading coefficient is the integral of x_j over the Newton polytope corresponding to N . That is,*

$$p_j(m) = \int_N x_j dV m^{k+1} + o(m^{k+1}).$$

The proof is similar to that of Theorem 24, so we omit it here.

Computing the leading coefficient of $p_{\mathcal{F}}$. The $p_{\mathcal{F}}$ corresponding to a system of polynomial equations is fundamentally different from the single polynomial case because we need to consider all the leading monomials, $\text{LM}(f_j)$ for $j = 1, \dots, n$. We use a *lifting* technique to address this issue here, that is, lifting a calculation from \mathbb{Z}^k to \mathbb{Z}^{k+1} .

First, we define $B = \{\alpha_1, \dots, \alpha_n\}$ and rewrite S_ℓ in Equation (5) as

$$\begin{aligned} S_\ell &= (m - \ell)N(f) + \bigcup_{\substack{\ell_1 + \dots + \ell_n = \ell \\ 0 \leq \ell_j \leq m}} \sum_{j=1}^n \ell_j \alpha_j \\ &= \ell B + (m - \ell)N(f). \end{aligned}$$

Next, we provide the result of Lemma 25 in the case of a system of polynomial equations.

Denote $\tilde{A} = (A, 1) \cup (B, 0)$ and its convex hull as \tilde{N} . Here, $(A, 1)$ means $\{(i_1, \dots, i_k, 1) \mid (i_1, \dots, i_k) \in A\}$, and similarly for $(B, 0)$.

Lemma 29. *For $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$, $\sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| = V(\tilde{N}) m^{k+1} + o(m^{k+1})$.*

Proof. From Lemma 19, we know that there exists a polynomial $p_E(m)$ such that $|mN(f) \cap \mathbb{Z}^k| = p_E(m)$, and the leading coefficient is $V(N(f))$. Therefore, we have

$$\begin{aligned} \sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| &= \sum_{\ell=0}^m |\ell B + (m - \ell)N(f) \cap \mathbb{Z}^k| \\ &= |m\tilde{N} \cap \mathbb{Z}^k| \\ &= V(\tilde{N}) m^{k+1} + o(m^{k+1}). \end{aligned}$$

□

The key idea is lifting a calculation from \mathbb{Z}^k to \mathbb{Z}^{k+1} . Therefore, we can compute $p_{\mathcal{F}}$ as the following theorem:

Theorem 30. *Suppose $f_1, \dots, f_n \in \mathbb{Z}[x_1, \dots, x_k]$ and m is an integer. Then $p_{\mathcal{F}}$ of the lattice obtained by the Newton polytope-based strategy is a polynomial in m , and the leading coefficient is $V(N) - V(\tilde{N})$. That is,*

$$p_{\mathcal{F}}(m) = (V(N) - V(\tilde{N})) m^{k+1} + o(m^{k+1}).$$

Proof. As $p_{\mathcal{F}}(m) = m \dim(\mathcal{L}) - \sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k|$, then using Theorem 27 and the above Lemma 29, we have

$$m \dim(\mathcal{L}) = V(N(f)) m^{k+1} + o(m^{k+1}),$$

and

$$\sum_{\ell=0}^m |S_\ell \cap \mathbb{Z}^k| = V(\tilde{N}) m^{k+1} + o(m^{k+1}).$$

Therefore, the leading coefficient of $p_{\mathcal{F}}$ is $V(N) - V(\tilde{N})$

□

4.4 Comparison with the Meers-Nowakowski Strategy

We mainly discuss two issues and place the improved results on CI-HNP for CSURF in Section 5:

- For a polynomial f with k variables, the claim of Meers and Nowakowski is that $m \dim(\mathcal{L})$, p_j and $p_{\mathcal{F}}$ are $k + 1$ -degree polynomials is incorrect. We provide the revised version and proof for Heuristic 2 in their paper [MN23].
- Comparing the efficiency of calculating asymptotic bounds, noting that their results are not provable.

Specifically, they claim that for a polynomial f with k variables, $\dim(\mathcal{L})$ is a degree k polynomial in m . This is incorrect because this property only holds when m is sufficiently large.

Considering the counterexample when $f = x^5 + x + 1$ for $m = 1, 2, \dots$, the corresponding $\dim(\mathcal{L})$ is the number of monomials in f^m , which is 3, 6, 10, 15, 20, 25 and so on. It can be seen that for $m \geq 3$, $\dim(\mathcal{L})$ always satisfies the polynomial $5 * m - 5$ while this does not hold for $m < 3$. In fact, we need the constraint that m is big enough to ensure that $\dim(\mathcal{L})$ is some polynomial in m . The same applies to p_j as well.

They propose a new heuristic assumption that $p_{\mathcal{F}}$ is also a $k + 1$ -degree polynomial in m . We find that this heuristic assumption is still incorrect, as it also requires m to be sufficiently large. Additionally, we present a counterexample for $p_{\mathcal{F}}$ involved in Heuristic 2, still considering f as $x^5 + x + 1$. For $m = 1, 2, \dots$, the number of $p_{\mathcal{F}}(m)$ is 2, 8, 20, 40, 65, 95 \dots . If we interpolate with $m = 1, 2, 3$, we get

$$p_{\mathcal{F}}(m) = 3m^2 - 3m + 2.$$

If we continue with $m = 2, 3, 4$, we get

$$p_{\mathcal{F}}(m) = 4m^2 - 8m + 8$$

and with $m = 3, 4, 5$, we get

$$p_{\mathcal{F}}(m) = \frac{5}{2}m^2 + \frac{5}{2}m - 10.$$

It is only at this point that we get the correct result $\frac{5}{2}$ but it still looks like not provable. However, using our Newton polytope-based explicit formula, we can quickly compute that the leading term of $p_{\mathcal{F}}(m)$ is

$$\frac{k}{k+1} \cdot V(N(f)) = \frac{1}{2} \cdot 5 = \frac{5}{2}.$$

Therefore, for the heuristic of Meers and Nowakowski, we need to add the condition that m is sufficiently large as a correction. The corrected proof can be obtained through our Theorem 30.

When using Lagrange interpolation to compute asymptotic bounds, by the state-of-the-art results in additive combinatorics, for some polynomials f with just 4 variables like examples in Section 5.2 or Section 5.3, m needs to be greater than 2^{300} in theory for $\dim(\mathcal{L})$ to be a polynomial in m [GSW23]. Hence, the interpolation method must involve the computation of f^m for a large m to make the results provable. Since we do not know how large m can be to ensure that $\dim(\mathcal{L})$ is a polynomial in m , a natural idea is that we can terminate the interpolation method for some m 's when the corresponding polynomial is stable. However, we find that sometimes the interpolation method may get stuck in local convergence (see more details in Section 5.2). Hence, the output of the interpolation method seems unbelievable. Moreover, for some f , it still requires a significant amount of time. We conducted several experiments in Section 5.

5 Applications

In this section, we present improved results on CI-HNP for CSURF as an application of our theory. We also compare with the Lagrange interpolation method. In Section 5.2, we point out that it suffers from cases of local convergence, leading to unprovable results. Then, we compare the time required for calculating asymptotic bounds to validate the efficiency of our algorithm in Section 5.3 and Section 5.4. Our experiments were performed using SageMath 10.3 on a MacBook Pro with an M1 chip, boasting a maximum CPU clock rate of 3.2 GHz.

5.1 Improving the Hidden Number Problem for CSURF

We first introduce the result of CI-HNP for CSURF proposed by Meers and Nowakowski [MN23]. Essentially, the CI-HNP for CSURF is to solve a system of the following polynomial equations:

Lemma 31 (Meers et al., [MN23]). *Given a modulus $M \in \mathbb{N}$ and polynomials*

$$\begin{aligned} f_1(x_1, x_2, x_3) &:= x_1^2 + a_1 x_1 x_2^2 + a_2 x_1 x_2 + a_3 x_1 + a_4 x_2^2 + a_5 x_2 + a_6, \\ f_2(x_1, x_2, x_3) &:= x_3^2 + b_1 x_1^2 x_3 + b_2 x_1 x_3 + b_3 x_3 + b_4 x_1^2 + b_5 x_1 + b_6, \end{aligned}$$

for some constants $a_i, b_i \in \mathbb{Z}$, a bound $X \in \mathbb{N}$, and an arbitrarily small constant $\epsilon > 0$, if M is sufficiently large, and

$$X < M^{10/41-\epsilon},$$

then we can find all small roots $u = (u_1, u_2, u_3)$ of f_1 and f_2 modulo M , such that $|u_i| < X$ in time polynomial in $\log(M)$, under Assumption 1.

For CI-HNP in CSURF, the dimension and determinant of the lattice are quite complex to compute. However, using the explicit formulas we discovered, it becomes much more convenient. Below are the improved results we obtained by our strategy in Section 4:

Theorem 32. *Given a modulus $M \in \mathbb{N}$ and polynomials*

$$\begin{aligned} f_1(x_1, x_2, x_3) &:= x_1^2 + a_1 x_1 x_2^2 + a_2 x_1 x_2 + a_3 x_1 + a_4 x_2^2 + a_5 x_2 + a_6, \\ f_2(x_1, x_2, x_3) &:= x_3^2 + b_1 x_1^2 x_3 + b_2 x_1 x_3 + b_3 x_3 + b_4 x_1^2 + b_5 x_1 + b_6, \end{aligned}$$

for some constants $a_i, b_i \in \mathbb{Z}$, a bound $X \in \mathbb{N}$, and an arbitrarily small constant $\epsilon > 0$, if M is sufficiently large, and

$$X < M^{8/31-\epsilon},$$

then we can find all small roots $u = (u_1, u_2, u_3)$ of f_1 and f_2 modulo M , such that $|u_i| < X$. in time polynomial in $\log(M)$, under Assumption 1.

Proof. Using our strategy mentioned in Section 4 and explicit formulas, we can quickly compute

$$\begin{aligned} \dim(\mathcal{L}) &= \frac{8}{3}m^3 + o(m^3), \\ p_1 &= 2m^4 + o(m^4), \\ p_2 &= \frac{5}{3}m^4 + o(m^4), \\ p_3 &= \frac{3}{2}m^4 + o(m^4), \\ p_{\mathcal{F}} &= \frac{4}{3}m^4 + o(m^4), \end{aligned}$$

resulting in $X < M^{8/31}$. □

This improves Theorem 4 in [MN23] of $XYZ < M^{30/40}$, thereby improving the CSURF result from requiring 31/40 MSBs in Theorem 7 in [MN23] to only requiring 23/31 MSBs.

5.2 Example for Local Convergence

Let us illustrate the phenomenon of local convergence encountered when using the interpolation method proposed in [MN23]. Consider the following polynomial f with

$$\text{supp}\{f\} = \{x_1^3, x_1 x_2, x_1 x_3, x_2, x_3^2 x_4^2, x_4^5, 1\}.$$

Here we use $p_{\mathcal{M}}$ to represent $m \dim(\mathcal{L})$. We compute f^m and then track the corresponding $p_{\mathcal{M}}(m)$ and $p_j(m)$. Here, f has four variables, i.e., $k = 4$. According to the results of [Kho92], we know that

$\dim(\mathcal{L})$ should be polynomials in m with degree 4 when $m > N_{kh}$. However, according to the result in [GSW23], the upper bound,

$$N_{kh} = (2 \times 7 \times 5)^{8 \times 7} \approx 2^{343},$$

is extremely large, making it impractical.

A natural idea is to consider the values of $p_{\mathcal{M}}$ and p_j at $m-5, m-4, m-3, m-2, m-1, m$ when $m \geq 5$, and then interpolate to obtain a fifth-degree polynomial, recording the leading coefficient. The relevant numerical values are presented in Figure 4. In Figure 4a, we observe that as m increases, the leading coefficient of $p_{\mathcal{M}}$ stabilizes at $25/12$. However, this is incorrect. Continuing to increase m , we eventually find that the leading coefficient of $p_{\mathcal{M}}$ stabilizes at 2. All the information can be found in Figure 4b, where the part to the left of the gray dashed line corresponds to Figure 4a.

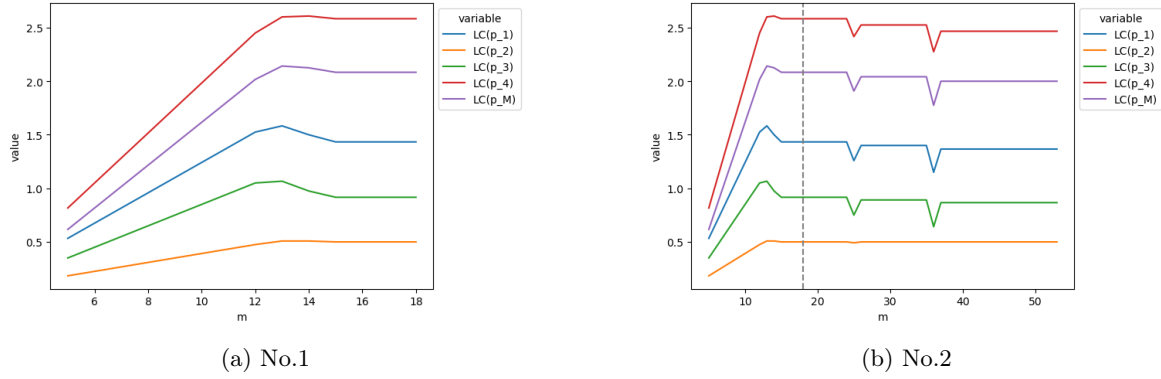


Fig. 4: An example for Local Convergence

In practical applications of the interpolation method, one might encounter issues with results getting stuck in local convergence. Additionally, even the state-of-the-art results in Additive Combinatorics as proposed in [GSW23] may lead to overly large computations. Our novel strategy effectively tackles these challenges, which further highlights the value of our work.

5.3 A Toy Example

Here we use a toy example to demonstrate the significant reduction in computational time achieved by our new method to compute asymptotic upper bounds.

We use the following system of polynomial equations:

$$\begin{cases} f_1 \text{ with } \text{supp}\{f_1\} = \{x_3^3 x_4^2, x_2^2, x_1 x_2, 1\} \\ f_2 \text{ with } \text{supp}\{f_2\} = \{x_4^5, x_3^2 x_1, x_1^3, x_2, 1\}. \end{cases}$$

Now we see the following Figure 5 to see how large m needs to be to satisfy the interpolation and get the value we want.

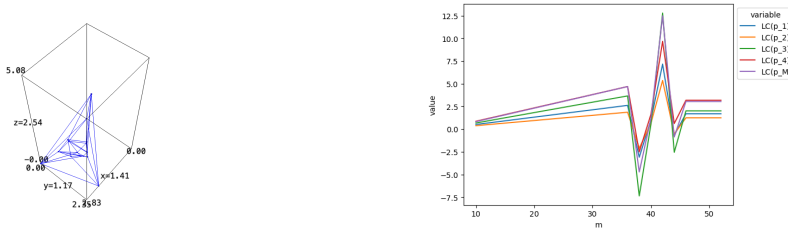


Fig. 5: The left subfigure represents the Newton polytope corresponding to $\{f_1, f_2\}$. For the right subfigure, for the horizontal coordinate m , the vertical coordinate indicates the leading coefficient obtained by interpolating $p_{\mathcal{M}}$ and p_j at $m-5, m-4, m-3, m-2, m-1, m$ when $m \geq 5$.

If we want to obtain provable results using Lagrange interpolation, we must require that m is greater than N_{kh} proposed by [GSW23], which is more than 2^{300} . It is infeasible to compute f^m for such a large m . If we heuristically choose m such that the leading term of $\dim(\mathcal{L})$ remains invariant with respect to m , we must also choose $m > 40$, which costs more than 20 minutes! However, if we use the explicit formula, the time cost is less than 0.5s to compute the volume of its Newton polytope!

5.4 More Experiments

We conducted more experiments to demonstrate the advantages of our method compared to the interpolation method.

As we discussed above, using interpolation requires m to be sufficiently large such that $|\text{supp}(f^m)|$ is a polynomial in m . Therefore, the inevitable computation of f^m consumes a significant amount of time. Intuitively, for $f \in \mathbb{Z}[x_1, \dots, x_k]$, the sparser the points in $A(f)$ are within $N(f)$, the larger the required m . As the number of variables k increases, the corresponding m also increases. We conducted experiments under different values of k as the following Table 1.

	k	Interpolation [MN23]	Ours
Exp. 1	3	10.9 s	0.2 s
Exp. 2	3	46.2 s	0.04 s
Exp. 3	3	2238.2 s	0.1 s
Exp. 4	3	828.3 s	0.1 s
Exp. 5	4	33112.4 s	0.2 s
Exp. 6	4	32451.7 s	0.1 s
Exp. 7	4	-	0.1 s
Exp. 8	4	-	0.2 s

Table 1: Running time for computing asymptotic upper bounds (" - " means longer than 24 h). The details about the polynomials used in our experiments can be found in Table 2 in Appendix B.

For the polynomial f , using Lagrange interpolation, we employ the Automated Coppersmith method to construct the lattice, compute $\dim(\mathcal{L})$ and $\det(\mathcal{L})$, and then use interpolation to obtain the leading coefficient. For our Newton polytope approach, we directly calculate the volume of the corresponding polytope and record the computation time. Our method shows significant time advantages, as these polynomials require large values of m before becoming polynomial-like. For instance, in Exp. 5 - Exp. 8, the minimum required values of m were 12, 12, 14, and 18, respectively.

6 Conclusion

In this paper, we introduced a new and powerful mathematical tool from Additive Combinatorics to improve Coppersmith's method for solving polynomial equations. We revisited the strategy of Jochemsz and May, as well as the strategy of Meers and Nowakowski, and proved that the bounds obtained by the two strategies can be efficiently achieved by the volume of the corresponding Newton polytope. We also proposed a novel Newton polytope-based strategy, eliminating the need for computing large exponentiations of polynomials in the previous strategies. As applications, we proved a heuristic introduced by Meers and Nowakowski at Asiacrypt'23 and improved the cryptanalytic result for the Commutative Isogeny Hidden Number Problem.

References

- Ajt98. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19, 1998.

- BHHG01. D. Boneh, S. Halevi, and N. Howgrave-Graham. The modular inversion hidden number problem. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pages 36–51. Springer, 2001.
- BM05. J. Blömer and A. May. A tool kit for finding small roots of bivariate polynomials over the integers. In *Advances in Cryptology—EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005. Proceedings 24*, pages 251–267. Springer, 2005.
- BV96. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *Annual International Cryptology Conference*, pages 129–142. Springer, 1996.
- BV97. M. Brion and M. Vergne. Lattice points in simple polytopes. *Journal of the American Mathematical Society*, pages 371–392, 1997.
- CG20. M. J. Curran and L. Goldmakher. Khovanskii’s theorem and effective results on sumset structure. *arXiv preprint arXiv:2009.02140*, 2020.
- Cop96. D. Coppersmith. Finding a small root of a univariate modular equation. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 155–165. Springer, 1996.
- Cop97. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of cryptology*, 10(4):233–260, 1997.
- DMH20. G. De Micheli and N. Heninger. Recovering cryptographic keys from partial information, by example. *Cryptology ePrint Archive*, 2020.
- Ehr62. E. Ehrhart. Sur les polyèdres rationnels homothétiques à n dimensions. *CR Acad. Sci. Paris*, 254:616, 1962.
- FNP24. Y. Feng, A. Nitaj, and Y. Pan. Partial prime factor exposure attacks on some RSA variants. *Theoretical Computer Science*, 999:114549, 2024.
- GS20. A. Granville and G. Shakan. The frobenius postage stamp problem, and beyond. *Acta Mathematica Hungarica*, 161(2):700–718, 2020.
- GSW23. A. Granville, G. Shakan, and A. Walker. Effective results on the size and structure of sumsets. *Combinatorica*, 43(6):1139–1178, 2023.
- GW21. A. Granville and A. Walker. A tight structure theorem for sumsets. *Proceedings of the American Mathematical Society*, 149(10):4073–4082, 2021.
- HG97. N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *IMA International Conference on Cryptography and Coding*, pages 131–142. Springer, 1997.
- HG01. N. Howgrave-Graham. Approximate integer common divisors. In *International cryptography and lattices conference*, pages 51–66. Springer, 2001.
- HR23. N. Heninger and K. Ryan. The hidden number problem with small unknown multipliers: Cryptanalyzing mega in six queries and other applications. In *IACR International Conference on Public-Key Cryptography*, pages 147–176. Springer, 2023.
- JM06. E. Jochemsz and A. May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In *Advances in Cryptology—ASIACRYPT 2006: 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3–7, 2006. Proceedings 12*, pages 267–282. Springer, 2006.
- Kho92. A. G. Khovanskii. Newton polyhedron, Hilbert polynomial, and sums of finite sets. *Functional Analysis and Its Applications*, 26(4):276–281, 1992.
- LLL82. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261:515–534, 1982.
- May02. A. May. Cryptanalysis of unbalanced RSA with small CRT-exponent. In *Annual International Cryptology Conference*, pages 242–256. Springer, 2002.
- May03. A. May. *New RSA vulnerabilities using lattice reduction methods*. PhD thesis, Citeseer, 2003.
- MN23. J. Meers and J. Nowakowski. Solving the Hidden Number Problem for CSIDH and CSURF via Automated Coppersmith. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 39–71. Springer, 2023.
- MNS21. A. May, J. Nowakowski, and S. Sarkar. Partial key exposure attack on short secret exponent CRT-RSA. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 99–129. Springer, 2021.
- MNS22. A. May, J. Nowakowski, and S. Sarkar. Approximate divisor multiples—factoring with only a third of the secret CRT-exponents. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 147–167. Springer, 2022.
- MR09. A. May and M. Ritzenhofen. Implicit factoring: On polynomial time factoring given only an implicit hint. In *International Workshop on Public Key Cryptography*, pages 1–14. Springer, 2009.
- Mum76. D. Mumford. *Algebraic geometry I: complex projective varieties*. Springer, 1976.

- Nat72. M. B. Nathanson. Sums of finite sets of integers. *The American Mathematical Monthly*, 79(9):1010–1012, 1972.
- Nat23. M. B. Nathanson. Extremal problems and the combinatorics of sumsets. *arXiv preprint arXiv:2310.18277*, 2023.
- Sta78. R. P. Stanley. Hilbert functions of graded algebras. *Advances in Mathematics*, 28(1):57–83, 1978.
- TLP17. A. Takayasu, Y. Lu, and L. Peng. Small CRT-exponent RSA revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 130–159. Springer, 2017.
- WCC11. J.-D. Wu, F.-J. Chen, and Y.-G. Chen. On the structure of the sumsets. *Discrete mathematics*, 311(6):408–412, 2011.
- XSH⁺19. J. Xu, S. Sarkar, L. Hu, H. Wang, and Y. Pan. New results on modular inversion hidden number problem and inversive congruential generator. In *Annual International Cryptology Conference*, pages 297–321. Springer, 2019.

A The equivalence of the Jochemsz-May Strategy and the Meers-Nowakowski Strategy for a single polynomial equation

Suppose $f \in \mathbb{Z}[x_1, \dots, x_k]$ is a polynomial with a small root $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{Z}^k$ modulo some integer M . Here a small root means we assume that we know an upper bound for the root, namely $|u_j| < X_j$ for some given X_j .

For the Jochemsz-May Strategy [JM06], they defined the following set

$$J_\ell = \{\lambda | \lambda \in \text{supp}\{f^m\} \text{ and } \frac{\lambda}{\text{LM}(f)^\ell} \in \text{supp}\{f^{m-\ell}\}\} \quad (6)$$

and then define the shift polynomials as follows

$$g_{[i_1, \dots, i_k]}(x_1, \dots, x_k) = \frac{x_1^{i_1} \cdots x_k^{i_k}}{\text{LM}(f)^\ell} f^\ell M^{m-\ell},$$

for $\ell = 0, \dots, m$, and $x_1^{i_1} \cdots x_k^{i_k} \in J_\ell \setminus J_{\ell+1}$.

Recently, Meers and Nowakowski generalized the Jochemsz-May Strategy from a single polynomial equation to a system of polynomial equations. However, the Meers-Nowakowski Strategy and the Jochemsz-May Strategy still have a slight difference in the single polynomial equation case. In the Meers-Nowakowski Strategy, they choose

$$J'_\ell = \{\lambda | \lambda \in \text{supp}\{f^m\} \text{ and } \frac{\lambda}{\text{LM}(f)^\ell} f^\ell \in \text{supp}\{f^m\}\}. \quad (7)$$

Although J_ℓ and J'_ℓ appear somewhat different, nevertheless, we will prove that this difference is essentially negligible.

Suppose the lattices obtained through the Jochemsz-May strategy and the Meers-Nowakowski strategy are \mathcal{L} and \mathcal{L}' , respectively. They have the same dimension, which is $mA(f)$. For determinate, we have the following theorem:

Theorem 33. *For $f \in \mathbb{Z}[x_1, \dots, x_k]$ and an integer m , suppose the lattices obtained through the Jochemsz-May strategy and the Meers-Nowakowski strategy are \mathcal{L} and \mathcal{L}' , respectively. Write*

$$\begin{aligned} \det(\mathcal{L}) &= X_1^{p_1(m)} \cdots X_k^{p_k(m)} M^{p_{\mathcal{F}}(m)}, \\ \det(\mathcal{L}') &= X_1^{p'_1(m)} \cdots X_k^{p'_k(m)} M^{p'_{\mathcal{F}}(m)}. \end{aligned}$$

We have

$$p_j(m) = p'_j(m), \text{ for } j = 1, \dots, k,$$

and

$$\lim_{m \rightarrow \infty} \frac{p_{\mathcal{F}}(m)}{p'_{\mathcal{F}}(m)} = 1.$$

Proof. The difference lies in the fact that $J_\ell \subset J'_\ell$, which results in differences in p_F and p'_F . However, assuming that the lattice obtained through the Newton polytope strategy is \mathcal{L}'' with corresponding p''_F , we have $p_F \geq p'_F \geq p''_F$. From Section 3, we know that

$$\lim_{m \rightarrow \infty} \frac{p_F}{p'_F} = 1.$$

Thus, it follows that

$$\lim_{m \rightarrow \infty} \frac{p_F}{p''_F} = 1.$$

In fact, they both have the leading coefficient $\frac{k}{k+1}V(N(f))$.

B Details for f in Section 5

We also provide detailed information about the polynomials used in our experiments in Table 2.

	$\text{supp}\{f\}$
Exp. 1	$\text{supp}\{(x_1 * x_2 + x_1 + x_2 + 1) * (x_2 * x_3 + x_2 + x_3 + 1) * (x_1 * x_3 + x_1 + x_3 + 1)\}$
Exp. 2	$\text{supp}\{x_1^3 + x_1 * x_2 + x_1 * x_3^2 + x_2^2 * x_3^3 + x_2^2 + x_2 + 2\}$
Exp. 3	$\text{supp}\{(x_3^3 * x_2^2 + x_2^2 + x_1 * x_2 + 1) * (x_3^2 * x_1 + x_1^3 + x_2 + 1)\}$
Exp. 4	$\text{supp}\{(x_2^2 + x_1^3 * x_2 + 1) * (x_1^2 + x_2 * x_3 + x_3^4 + 1)\}$
Exp. 5	$\text{supp}\{(x_2^2 + x_1^3 * x_2 + x_2^2 * x_4^3 + 1) * (x_1^2 + x_2 * x_3 + x_3^4 + 1)\}$
Exp. 6	$\text{supp}\{(x_2^2 + x_1^3 * x_2 + x_2^2 * x_4^5 + 1) * (x_1^2 + x_2 * x_3 + x_3^4 + 1)\}$
Exp. 7	$\text{supp}\{(x_2^2 + x_1^3 * x_2 + x_2^2 * x_4^3 + 1) * (x_1^2 + x_2 * x_3 + x_3^2 * x_4 + 1)\}$
Exp. 8	$\text{supp}\{(x_2^2 + x_1^3 * x_2 + x_2^2 * x_4^5 + 1) * (x_1^2 + x_2 * x_3 + x_3^2 * x_4 + 1)\}$

Table 2: Details of f in Table 1.