A Failure Probability Analysis of a Modular Algorithm to Compute the Monic GCD of Multivariate Polynomials over Algebraic Number Fields $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$

Mahsa Ansari and Michael Monagan

Department of Mathematics, Simon Fraser University
Burnaby, British Columbia, V5A 1S6, Canada
mansari@sfu.ca and mmonagan@sfu.ca

Abstract. Let $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ be an algebraic number field. In 2023, Ansari and Monagan designed a modular algorithm to compute the monic gcd g of two polynomials f_1 and f_2 in $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x_1, \ldots, x_k]$. The algorithm computes g modulo primes and uses interpolation to recover $x_2, x_3, \ldots, x_k$ in g. However, the algorithm may fail in certain cases, for instance, when encountering a zero divisor. In this paper, we present a refined classification of failure cases for this algorithm and provide a detailed analysis of their probabilities.

Keywords: Modular Algorithms. Failure Probability. Algebraic Number Fields.

1 Introduction

In 1967, Collins [6] introduced a modular algorithm for computing univariate gcds in $\mathbb{Z}[x]$ using homomorphic reductions and Chinese remaindering. In 1971 Brown [4] extended this approach to multivariate polynomials. Langemyr and McCallum [11] subsequently adapted these algorithms to work over algebraic number fields $\mathbb{Q}(\alpha)$. Later, Encarnacion [9] used rational number reconstruction to recover the rational coefficients in the target gcd and make the algorithm for $\mathbb{Q}(\alpha)$ output sensitive. In 2002, Monagan and Van Hoeij [14] generalized Encarnacion's method to treat polynomials in $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x]$ but they did not analyze the failure probability of their algorithm. Building on this foundation, in 2023, Ansari and Monagan [2] proposed a modular algorithm for computing the monic gcd of polynomials in $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x_1, \ldots, x_k]$ but they too did not do a failure probability analysis. Their algorithm, called MGCD (see Algorithm 5 in Appendix A), simplifies the computation over $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ by converting the input polynomials to their corresponding polynomials over $\mathbb{Q}(\gamma)$ where γ is a primitive element of $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$. To avoid coefficient growth, MGCD performs computations modulo primes. It also reduces the multivariate gcd problem to many univariate gcds through evaluation, and then interpolates $x_2, x_3, \ldots, x_n$

in the result using dense interpolation. The univariate gcds are computed using the monic Euclidean algorithm [14], which can fail if it encounters a zero divisor. To recover the rational coefficients of the monic gcd, MGCD employs Chinese remaindering and rational number reconstruction. In this paper, we categorize primes and evaluation points that can lead to failure and derive bounds on their likelihood.

2 Preliminaries

Let $L_0 = \mathbb{Q}$. For each $i = 1, 2, \ldots, n$, define $L_i = L_{i-1}[z_i]/\langle M_i(z_i)\rangle$ where $M_i(z_i)$ is the monic minimal polynomial of α_i over L_{i-1}. The field $L = L_n$ is a $\mathbb{Q}$-vector space of dimension $d = \prod_{i=1}^{n} d_i$ where $d_i = \deg(M_i, z_i)$ with basis $B_L = \{\prod_{i=1}^{n}(z_i)^{e_i} \mid 0 \le e_i < d_i\}$. Since $L \cong \mathbb{Q}(\alpha_1, \ldots, \alpha_n)$, computations in $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ can be done by replacing each α_i with the corresponding variable z_i, and then performing the computation within L. In algorithm MGCD (see Algorithm 5 in Appendix A), we assume that the minimal polynomials $M_1(z_1), \ldots, M_n(z_n)$ are provided, which allows us to construct L. We denote the **coordinate vector** of $a \in L$ w.r.t. the basis B_L by $[a]_{B_L}$.

In this paper, R refers to a commutative ring with identity $1 \neq 0$. Fix a monomial ordering in $R[x_1, \ldots, x_k]$. For $f \in R[x_1, \ldots, x_k]$ denote its leading coefficient and leading monomial by $\mathrm{lc}(f)$ and $\mathrm{lm}(f)$, respectively. If $f = 0$, define $monic(f) = 0$. If $f \neq 0$ and $\mathrm{lc}(f)$ is a unit in R, then $monic(f) = \mathrm{lc}(f)^{-1} f$. Otherwise, $monic(f) = failed$. Let $f_1, f_2 \in R[x_1, \ldots, x_k]$, and suppose a monic $g = \gcd(f_1, f_2)$ exists. Then g is unique [14], and there exist polynomials h_1 and h_2 such that $f_1 = h_1 \cdot g$ and $f_2 = h_2 \cdot g$; these are called **cofactors** of f_1 and f_2, respectively.

Example 1. Let $L = \mathbb{Q}[z_1, z_2]/\langle z_1^2 - 2, z_2^2 - 3\rangle$ with basis $B_L = \{1, z_2, z_1, z_1 z_2\}$. Let $f_1 = (z_2 x + z_1 y)(z_1 x + y)$ and $f_2 = (z_2 x + z_1 y)(x - z_2 y) \in L[x, y]$. By inspection, $\gcd(f_1, f_2) = z_2 x + z_1 y$. Fixing lexicographical order with $x > y$, the monic gcd is $g = x + \frac{1}{3} z_1 z_2 y$.

Definition 1. *Given $f_1, f_2 \in R[x]$ with $0 \le \deg(f_2) \le \deg(f_1)$, assume that Algorithm 1: Monic Euclidean Algorithm (MEA) does not fail for f_1 and f_2 and terminates after $l+1$ iterations. We define the Monic Polynomial Remainder Sequence, m.p.r.s., generated by polynomials f_1 and f_2 as the sequence $r_1, r_2, \ldots, r_l$ obtained from the execution of the Monic Euclidean Algorithm such that $r_1 = f_1$, $r_2 = f_2$, $r_3 = r_1 - M_2 q_3$, and $r_{i+2} = M_i - M_{i+1} q_{i+1}$ with $M_i = monic(r_i)$ and $\deg(r_{i+1}) < \deg(r_i)$ for $2 \le i \le l-1$ and $r_{l+1} = 0$.*

Let $L_{\mathbb{Z}} = \mathbb{Z}[z_1, \ldots, z_n]$. For any $f \in L[x]$, the **denominator** of f, denoted by $den(f)$, is the smallest positive integer such that $den(f) f \in L_{\mathbb{Z}}[x]$. In addition, the **associate** of f is defined as $\tilde{f} = den(h) h$ where $h = monic(f)$. The **semi-associate** of f, denoted by $\check{f}$, is defined as rf, where r is the smallest positive rational number for which $den(rf) = 1$. For instance, let L be as in Example 1 and $f = \frac{3}{2} z_1 x + z_2 \in L[x]$. Then $den(f) = 2$, $\check{f} = 3 z_1 x + 2 z_2$,

Algorithm 1: Monic Euclidean Algorithm (MEA)

Input: $f_1, f_2 \in R[x]$ such that $0 \leq \deg(f_2) \leq \deg(f_1)$ and R is a commutative ring with identity $1 \neq 0$.
Output: Either the monic $\gcd(f_1, f_2)$ or FAIL.

```
r_1, r_2 = f_1, f_2
M_1, i = r_1, 2
while r_i ≠ 0 do
    M_i = monic(r_i)
    if M_i = failed then return(FAIL) // The algorithm encountered a zero-divisor.
    Set r_{i+1} to be the remainder of M_{i-1} divided by M_i
    Set i = i + 1
l = i - 1
return(M_l)
```

monic$(f) = x + \frac{1}{3}z_1z_2$ and $\tilde{f} = 3x + z_1z_2$. To improve computational efficiency, in a preprocessing step, MGCD clears fractions by replacing the input polynomials f_1 and f_2 with their semi-associates. MGCD speeds up the computation by mapping $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ to $\mathbb{Q}(\gamma)$ where γ is a primitive element. This is done using the LAminpoly algorithm, Algorithm 2, over $\mathbb{F} = \mathbb{Z}_p$ where p is a prime. The computation is done mod p to prevent expression swell. However, not all the primes result in the successful reconstruction of the monic gcd. In the following example, we explain how the LAminpoly algorithm works and how polynomials over $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ are converted to $\mathbb{Q}(\gamma)$.

Algorithm 2: LAminpoly

Input: $[m_1(z_1), \ldots, m_n(z_n)]$, field $\mathbb{F} = \mathbb{Z}_p$, $\gamma = z_1 + \sum_{i=2}^{n} C_{i-1}z_i$ with $C_i \in \mathbb{Z} \setminus \{0\}$
Output: FAIL or $M(z) \in \mathbb{F}[z]$ s.t. $M(\gamma) = 0$, matrix A, and A^{-1}

```
Let d_i = deg(m_i(z_i)), B_{L_p} = {∏ z_i^{e_i} | 0 ≤ e_i < d_i}, d = ∏ d_i
Initialize A as d × d zero matrix over F
g_0 = 1
for i = 1 to d do
    Set column i of A to [g_{i-1}]_{B_{L_p}}
    g_i = γ · g_{i-1}
if det(A) = 0 then
    return (FAIL)
Compute A^{-1} and set q = A^{-1} · (−[g_d]_{B_{L_p}})
M(z) = z^d + q_d z^{d-1} + ··· + q_1
return (M(z), A, A^{-1})
```

Example 2. Given L as defined in Example 1, choose $p = 5$ so the ground field in LAminpoly algorithm is $\mathbb{F} = \mathbb{Z}_5$. After reducing the minimal polynomials modulo p, we have $L_5 = \mathbb{Z}_5[z_1, z_2]/\langle z_1^2 + 3, z_2^2 + 2\rangle$ with basis $B_{L_p} = \{1, z_2, z_1, z_1 z_2\}$. Let $\gamma = z_1 + z_2$. Algorithm 2 checks whether γ is a primitive element of L or not. If γ is a primitive element, then Algorithm 2 computes the characteristic polynomial of γ, $M(z)$, so we can construct $\bar{L}_5 = \mathbb{Z}_5[z]/\langle M(z)\rangle$ such that $\bar{L}_5 \cong L_5$. LAminpoly algorithm first constructs the 4×4 matrix $A = [[1, 0, 0, 0], [0, 1, 0, 4], [0, 1, 0, 1], [0, 0, 2, 0]]$ whose i'th column is $[\gamma^i]_{B_{L_p}}$ for $0 \le i \le 3$. Since $\det(A) = 9 \mod 5 \ne 0$, we consider $\gamma = z_1 + z_2$ as a primitive element of $\mathbb{Z}_5(\sqrt{2}, \sqrt{3})$. If we had chosen $p = 3$, then $\det(A) \mod p = 0$ and A would not be invertible. We call 3 det-bad prime and define it in section 3. Computing $q = A^{-1} \cdot (-[\gamma^4]_{B_L})$, we construct the characteristic polynomial $M(z) = z^d + q_d z^{d-1} + \ldots + q_2 z + q_1$. Thus, we have $M(z) = z^4 + 1$ and $\bar{L}_5 = \mathbb{Z}_5[z]/\langle z^4 + 1\rangle$.

Notations 1. *We use the following notation in this paper.*

- *Let p be a prime such that $p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i)$. Let $m_i(z_i) = M_i \mod p$ for $1 \le i \le n$. Define $L_p = \mathbb{Z}_p[z_1, \ldots, z_n]/\langle m_1, \ldots, m_n\rangle$.*
- *$\bar{L}_p = \mathbb{Z}_p[z]/\langle M(z)\rangle$ where $M(z)$ is obtained from Algorithm 2 over $\mathbb{Z}_p$.*
- *$\bar{L}_{\mathbb{Z}} = \mathbb{Z}[z]/\langle M(z)\rangle$ where $M(z)$ is obtained from Algorithm 2 over $\mathbb{Q}$.*

Let $B_{L_p} = \{\prod_{i=1}^{n} (z_i)^{e_i}$ *s.t* $0 \le e_i < d_i\}$ and $B_{\bar{L}_p} = \{1, z, z^2, \ldots, z^{d-1}\}$ be bases for L_p and $\bar{L}_p$, respectively. Let $C : L_p \longrightarrow \mathbb{Z}_p^d$ and $D : \bar{L}_p \longrightarrow \mathbb{Z}_p^d$ be bijections such that $C(a) = [a]_{B_{L_p}}$ and $D(b) = [b]_{B_{\bar{L}_p}}$. Define $\phi_\gamma : L_p \longrightarrow \bar{L}_p$ such that $\phi_\gamma(a) = D^{-1}(A^{-1} \cdot C(a))$, where A is the matrix obtained from the LAminpoly algorithm over $F = \mathbb{Z}_p$. Moreover, $\phi_\gamma^{-1} : \bar{L}_p \longrightarrow L_p$ such that $\phi_\gamma^{-1}(b) = C^{-1}(A \cdot D(b))$.

Example 3. Let $f_1 \in L$ be the polynomials in Example 1 and let $L_5 \cong \bar{L}_5$ where $\bar{L}_5 = \mathbb{Z}_5[z]/\langle z^4 + 1\rangle$ obtained from Example 2. Let $B_{L_p} = \{1, z_2, z_1, z_1 z_2\}$ and A be the matrix computed in Example 2. We have, $[f_1]_{B_{L_p}} = [2xy, xy, y^2, x^2]^T$ and $b = A^{-1} \cdot [f_1]_{B_{L_p}} = [2xy, 2xy + 3y^2, 2x^2, 3xy + 3y^2]$ as the coordinate vector of $\phi_\gamma(f)$ relative to $B_{\bar{L}_p} = \{1, z, z^2, z^3\}$. Therefore,

$$\phi_\gamma(f) = 2x^2 z^2 + (3z^3 + 2z + 2)yx + (3z^3 + 3z)y^2 \in \bar{L}_p[x, y].$$

Note that $\bar{L}_p$ is a finite ring with p^d elements which likely has zero divisors. After computing $\phi_\gamma(f_1), \phi_\gamma(f_2) \in \bar{L}_p[x_1, \ldots, x_k]$, the MGCD algorithm invokes PGCD (see Algorithm 6 in Appendix A) to compute the monic gcd over $\bar{L}_p$. PGCD is recursive. For $k = 1$ it applies the monic Euclidean Algorithm (MEA) [14]. For $k > 1$, PGCD uses a sequence of evaluation points to reduce the multivariate problem to the univariate case. It then uses MEA to compute the gcd. If MEA fails (e.g., due to encountering a zero-divisor), a new prime and evaluation point are chosen. PGCD reconstructs the gcd over $\bar{L}_p$ via dense interpolation. Once PGCD returns the monic gcd over $\bar{L}_p$, MGCD applies ϕ_γ^{-1} to undo ϕ_γ and map the gcd from $\bar{L}_p$ to its corresponding polynomials in L_p. To reconstruct the rational coefficients in g, MGCD applies Chinese remaindering and rational number reconstruction [12,13]. We give an example of MGCD to illustrate the treatment of zero-divisors in L_p and to motivate the use of a primitive element.

Example 4. Continuing Example 1, let MGCD pick $p = 5$ and define $L_5 = \mathbb{Z}_5[z_1, z_2]/\langle\, z_1^2 + 3, z_2^2 + 2\,\rangle$. From Example 2, we have $\bar{L}_p = \mathbb{Z}_5[z]/\langle z^4 + 1\rangle$. After converting f_1 and f_2 to $\phi_\gamma(f_1)$ and $\phi_\gamma(f_2) \in \bar{L}_p[x, y]$ as in Example 3, Algorithm PGCD chooses a random evaluation point, $y = 2$, and tries to compute $g_1 = \gcd((\phi_\gamma(f_1)(y = 2), \phi_\gamma(f_2)(y = 2))$ where

$$\phi_\gamma(f_1)(y = 2) = 2z^2x^2 + (z^3 + 4z + 4)x + 2z^3 + 2z$$
$$\phi_\gamma(f_2)(y = 2) = (3z^3 + 2z)x^2 + (z^3 + z + 4)x + 2z^2.$$

However, MEA fails after the second iteration since $\mathrm{lc}(r_3) = 4z^3 + 2z$ is not inverible over $\bar{L}_p$. We call $(p, \beta) = (5, 2)$ a zero-divisor pair.

Next, MGCD retries with $p = 7$ so $\bar{L}_7 = \mathbb{Z}_7[z]/\langle z^4 + 4z^2 + 1\rangle$. At $y = 1$, PGCD computes $g_1 = \gcd(\phi_\gamma(f_1)(y = 1), \phi_\gamma(f_2)(y = 1)) = 6z^2 + x + 5$. Notice that $\mathrm{lm}(g_1) = x$. A second evaluation at $y = 0$ yields $g_2 = \gcd(\phi_\gamma(f_1)(y = 0), \phi_\gamma(f_2)(y = 0)) = x^2$ which is discarded as $\mathrm{lm}(g_2) = x^2 > \mathrm{lm}(g_1)$, making $y = 0$ an unlucky evaluation point. Further evaluations at $y = 3$ and $y = 4$ allows interpolation of the monic gcd at y so $g_p = \gcd(\phi_\gamma(f_1), \phi_\gamma(f_2)) = x + (6z^2+5)y \in \bar{L}_7[x, y]$. Applying ϕ_γ^{-1} maps g_p back to $L_p[x, y]$. That is, $\phi_\gamma^{-1}(g_p) = x+(6z^2+5)y \in L_7[x, y]$. MGCD then attempts to reconstruct rational coefficients of g using Chinese remaindering and rational number reconstruction, but one modular image is insufficient. After doing the above process for additional primes $p = 11$, $p = 13$ and $p = 17$, reconstruction succeeds, yielding the correct monic gcd $\gcd(f_1, f_2) = x + \frac{1}{3}yz_2z_1 \in L[x, y]$.

Definition 2. *The resultant of f_1 and f_2 w.r.t. the variable x_i is defined as $res(f_1, f_2, x_i) = \det(\mathrm{sylv}(f_1, f_2, x_i)) \in R[x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_k]$ where $\mathrm{sylv}(f_1, f_2, x_i)$ is the Sylvester matrix of f_1 and f_2 w.r.t. the variable x_i.*

Theorem 1. *(See Corollary (Sylvester's Criterion), chapter 7 of [10].) Let $f_1, f_2 \in R[x]$ and suppose $g = \gcd(f_1, f_2)$ exists. Then $\deg(g, x) > 0$ if and only if $\mathrm{res}(f_1, f_2) = 0$.*

Algorithm 3 URES from [3] computes $\mathrm{res}(f_1, f_2)$ where $f_1, f_2 \in R[x]$. We note that Algorithm MEA fails if and only if Algorithm URES fails.

3 Failure Probability

In [2], we categorized the evaluation points that lead to failures in our PGCD algorithm into three types: lc-bad, zero-divisor, and unlucky evaluation points. Similarly, for the MGCD algorithm, we identified four types of primes that may cause failures: det-bad, lc-bad, zero-divisor, and unlucky primes. In this section, we refine these classifications and present a probabilistic analysis of their impact on the success of the MGCD algorithm.

Algorithm 3: URES

Input: $f_1, f_2 \in R[x]$ such that $0 \le \deg(f_2) \le \deg(f_1)$ where R is a commutative ring with identity $1 \ne 0$.
Output: Either $\mathrm{res}(f_1, f_2)$ or FAIL.

$r_1 = f_1$, $r_2 = f_2$, $i = 2$
$M_1 = r_1$, $R = 1$, $v = 0$
$n_1 = \deg(f_1)$,$n_2 = \deg(f_2)$
while $r_i \ne 0$ **do**
  $M_i = monic(r_i)$
  if $M_i = failed$ **return** *(FAIL)*`// The algorithm encounters a zero-divisor.`
  Set r_{i+1} to be the remainder of M_{i-1} divided by M_i
  Set $n_{i+1} = \deg(r_{i+1})$
  if $n_{i+1} < 0$ and $n_i \ne 0$ **then return**(0)`// If` $\gcd(f_1, f_2)$ `is a constant, then` $\mathrm{res}(f_1, f_2) = 0$
  Set $R = R \cdot \mathrm{lc}(r_i)^{n_{i-1}}$
  Set $v = v + n_i n_{i-1}$
  Set $i = i + 1$
$R = (-1)^v R$
return(R)

Notations 2. *We use the following notations in this section.*

- $\#f$ *denotes the number of terms of* $f \in R[x_1, \ldots, x_k]$.
- $d_i = \deg(M_i, z_i)$ *and* $d = \prod_{i=1}^{n} d_i$.
- $\mathbb{P}_b = \{$*all b-bit primes*$\}$, *that is, primes in* $(2^{b-1}, 2^b)$. *In our current code, we use* 31*-bit primes and* $\mid \mathbb{P}_{31} \mid = 50,697,537$.

Definition 3. *Let* $f \in L_{\mathbb{Z}}[x_1, \ldots, x_k, y]$. *We denote the* ***height*** *of* f *by* $\|f\|_\infty$ *and define it as the absolute value of the largest integer coefficient of* f *in magnitude.*

Let $f_1, f_2 \in R[x]$ be two non-zero polynomials such that $\mathrm{lc}(f_2)$ and $\mathrm{lc}(f_1)$ are units, and $0 \le \deg(f_2) \le \deg(f_1)$. Then

- Let rem (f_1, f_2) and $\mathrm{quo}(f_1, f_2)$ denote the remainder and quotient of f_1 divided by f_2, where rem $(f_1, f_2) = 0$ or $\deg(\mathrm{rem}\ (f_1, f_2)) < \deg(f_2)$, i.e., rem $(f_1, f_2) = f_1 - f_2\ \mathrm{quo}(f_1, f_2)$.
- Let $\mathrm{mrem}(f_1, f_2)$ and $\mathrm{mquo}(f_1, f_2)$ be the remainder and quotient of $\mathrm{monic}(f_1)$ divided by $\mathrm{monic}(f_2)$ i.e., $\mathrm{mrem}(f_1, f_2) = \mathrm{monic}(f_1) - \mathrm{monic}(f_2)\ \mathrm{mquo}(f_1, f_2)$.
- Let prem (f_1, f_2) and $\mathrm{pquo}(f_1, f_2)$ be the pseudo-remainder and pseudo-quotient of f_1 divided by f_2.

3.1 Lc-bad pairs

Definition 4. *Let* $f_1, f_2 \in L_{\mathbb{Z}}[x_1, \ldots, x_k]$ *be non-zero polynomials with* $\deg(f_2) \le \deg(f_1)$. *Let* p *be a prime and* $\beta \in [0, p)^{k-1}$ *be an evaluation point. We call the ordered pair* (p, β) ***lc-bad*** *if* $p \mid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i, z_i)$, *or* $p \mid \mathrm{lc}(f_2, x_1)(\beta)$.

Example 5. Let $f_1 = (y+z)x^3 + xz$ and $f_2 = (y+1)x + zx$ be polynomials in $L_{\mathbb{Z}}[x, y]$. The ordered pair $(p, \beta) = (7, 6)$ is lc-bad since $7 \mid \mathrm{lc}(f_2, x)(6) = 7$.

Theorem 2. *Let $f_1, f_2 \in L_{\mathbb{Z}}[x_1, \ldots, x_k]$ with $0 \le \deg(f_2) \le \deg(f_1)$, and $\|\mathrm{lc}(f_2, x_1)\|_\infty \le 2^h$, $T = \#\mathrm{lc}(f_2, x_1)$. Let $\mathrm{lc}(\check{M}_i, z_i) \le 2^m$ for $1 \le i \le n$, and let $D_l = \max_{i=1}^{k}(\deg(\mathrm{lc}(f_2, x_1), x_i))$. If p is chosen at random from $\mathbb{P}_b$ and β is chosen at random from $[0, p)^{k-1}$, then*

$$\mathrm{Prob}[(p, \beta) \text{ \it is lc-bad}] \le \frac{\lfloor h + D_l(k-1)b + \log_2 T \rfloor + nm}{b \mid \mathbb{P}_b \mid}.$$

Proof. By definition,

$$\mathrm{Prob}[(p, \beta) \text{ is lc-bad}] = \mathrm{Prob}[p \mid \mathrm{lc}(f_2, x_1)(\beta) \ \vee \ p \mid \mathrm{lc}(\check{M}_i, z_i) \text{ for some } 1 \le i \le n]$$

$$\le \mathrm{Prob}[p \mid \mathrm{lc}(f_2, x_1)(\beta)] + \sum_{i=1}^{n} \mathrm{Prob}[p \mid \mathrm{lc}(\check{M}_i, z_i)].$$

Write $\mathrm{lc}(f_2, x_1) = \sum_{i=1}^{t} a_{\alpha_i}(X) Z^{\alpha_i} \in \mathbb{Z}[x_2, \ldots, x_k][z_1, \ldots, z_n]$ whit $\alpha_i = (\alpha_{i_1}, \ldots, \alpha_{i_n}) \in \mathbb{Z}^n_{\ge 0}$ and $Z^{\alpha_i} = z_1^{\alpha_{i_1}} \cdots z_n^{\alpha_{i_n}}$ and $a_{\alpha_i}(X) \in \mathbb{Z}[x_2, \ldots, x_k]$. Then,

$$\mathrm{Prob}[p \mid \mathrm{lc}(f_2, x_1)(\beta)] = \mathrm{Prob}[p \mid a_{\alpha_1}(\beta) \wedge \ldots \wedge p \mid a_{\alpha_t}(\beta)] \le \mathrm{Prob}[p \mid a_{\alpha_1}(\beta)].$$

Let $D_i = \deg(a_{\alpha_1}, x_{i+1})$ for $1 \le i \le k-1$ so $\mid a_{\alpha_1}(\beta) \mid \le \|a_{\alpha_1}\|_\infty \cdot \#a_{\alpha_1} \cdot \prod_{i=1}^{k-1} \beta_i^{D_i}$. Since $\beta_i < p < 2^b$, $D_i \le D_l$, and $\|a_{\alpha_1}\|_\infty \le 2^h$, we have $\mid a_{\alpha_1}(\beta) \mid \le 2^h \cdot T \cdot p^{(k-1)D_l}$. Hence,

$$\mathrm{Prob}[p \mid a_{\alpha_1}(\beta)] \le \frac{\left\lfloor \frac{\log_2 (2^h \cdot T \cdot p^{(k-1)D_l})}{\log_2 2^b} \right\rfloor}{\mid \mathbb{P}_b \mid} \le \frac{\lfloor h + \log_2 T + D_l(k-1)b \rfloor}{b \mid \mathbb{P}_b \mid}.$$

Next, suppose that $\check{M}_i(z_i) = l_i z_i^{d_i} + \sum_{j=1}^{d_i - 1} a_{i,j} z_i^j$ where $a_{i,j}$ is a polynomial in $\mathbb{Z}[z_1, \ldots, z_{i-1}]/\langle \check{M}_1(z_1), \ldots, \check{M}_{i-1}(z_{i-1}) \rangle$. Since $l_i \le 2^m$, we have

$$\mathrm{Prob}[p \mid \mathrm{lc}(\check{M}_i)] \le \mathrm{Prob}[p \mid a_{i,j}] \le \frac{\lfloor \frac{m}{b} \rfloor}{\mid \mathbb{P}_b \mid} \le \frac{m}{b \mid \mathbb{P}_b \mid} \quad \text{for} \quad 1 \le i \le n. \tag{1}$$

Thus, $\sum_{i=1}^{n} \mathrm{Prob}[p \mid \mathrm{lc}(\check{M}_i)] \le n \frac{m}{b \mid \mathbb{P}_b \mid}$. This completes the proof.

3.2 Det-bad Primes

Definition 5. *Let p be a prime such that $p \nmid \prod_{i=1}^{n} \mathrm{lc}(M_i, z_i)$ and $p \nmid \mathrm{lc}(f_2, x_1)$. Let $\gamma = z_1 + C_1 z_2 + \ldots + C_{n-1} z_n$ where $0 \ne C_i \in \mathbb{Z}$ for $1 \le i \le n-1$. A prime p is called a **det-bad** prime if $p \mid \det(A)$, where A is the coefficient matrix of powers of γ obtained from Algorithm 2.*

This section bounds the probability that a randomly chosen prime $p \in \mathbb{P}_b$ s.t $p \nmid \prod_{i=1}^{n} \mathrm{lc}(M_i, z_i)$, is det-bad. This requires an upper bound on $\mid \det(A) \mid$. Let f rem $\langle \check{M}_n, \ldots, \check{M}_1 \rangle$ denote the remainder of f divided by $\check{M}_n, \ldots, \check{M}_1$ using a natural long division. We illustrate the construction of matrix A over $\mathbb{F} = \mathbb{Q}$ using the following example.

Example 6. Let $M_1(z_1) = z_1^2 - \frac{7}{2}$ and $M_2 = z_2^2 - \frac{11}{3}$. Then $\check{M}_1 = 2z_1^2 - 7$ and $\check{M}_2 = 3z_2^2 - 11$. Use the basis $B_L = \{1, z_2, z_1, z_1z_2\}$ for $\mathbb{Q}[z_1, z_2]/\langle M_1(z_1), M_2(z_2)\rangle$ of dimension $d = 4$. Take $\gamma = z_1 + 3z_2$, we have

$$\boxed{\begin{array}{l}\gamma^0 = 1\\ \gamma^1 = (z_1+3z_2)^1 \bmod \langle \check{M}_1, \check{M}_2\rangle = z_1 + 3z_2\\ \gamma^2 = (z_1+3z_2)^2 \bmod \langle \check{M}_1, \check{M}_2\rangle = 6z_1z_2 + \frac{73}{2}\\ \gamma^3 = (z_1+3z_2)^3 \bmod \langle \check{M}_1, \check{M}_2\rangle = \frac{205}{2}z_1 + \frac{261}{2}z_2\end{array}} \quad \text{and} \quad A = \begin{bmatrix} 1 & 0 & \frac{73}{2} & 0\\ 0 & 3 & 0 & \frac{261}{2}\\ 0 & 1 & 0 & \frac{205}{2}\\ 0 & 0 & 6 & 0\end{bmatrix}.$$

To bound the determinant of a matrix in $\mathbb{Z}$, we can employ Hadamard's bound.

Theorem 3. *[Hadamard's bound] Let A be an $n \times n$ matrix with $A_{i,j} \in \mathbb{Z}$. Then $|\det(A)| \leq \prod_{j=1}^{n} \sqrt{\sum_{i=1}^{n} A_{i,j}^2}$.*

As illustrated in Example 6, matrix $A \in \mathbb{Q}^{d\times d}$ so we cannot use Hadamard's bound to bound $|\det(A)|$. However, replacing long division with pseudo-division in Algorithm algorithm 2, we can obtain the matrix $\tilde{A} \in \mathbb{Z}^{d\times d}$ for which Hadamard's bound can be applied. Define f_1 prem $\langle \check{M}_n, \ldots, \check{M}_1\rangle$ as the recursive pseudo-remainder under $\check{M}_n$ through $\check{M}_1$.We construct $\tilde{A}$ such that its jth column is the coordinate vector $[\gamma^{j-1} \text{ prem } \langle \check{M}_n, \ldots, \check{M}_1\rangle]_{B_L}$.

Example 7. Considering Example 6, using pseudo-division, we obtain:

$$\boxed{\begin{array}{l}\gamma_0 = 1\\ \gamma_1 = (z_1+3z_2) \text{ prem } \langle \check{M}_1, \check{M}_2\rangle = z_1 + 3z_2\\ \gamma_2 = (z_1+3z_2)^2 \text{ prem } \langle \check{M}_1, \check{M}_2\rangle = 36z_1z_2 + 219\\ \gamma_3 = (z_1+3z_2)^3 \text{ prem } \langle \check{M}_1, \check{M}_2\rangle = 3690z_1 + 4698z_2\end{array}} \text{ and } \tilde{A} = \begin{bmatrix} 1 & 0 & 219 & 0\\ 0 & 3 & 0 & 4698\\ 0 & 1 & 0 & 3690\\ 0 & 0 & 36 & 0\end{bmatrix}.$$

Although pseudo-division allows us to construct an integer matrix $\tilde{A} \in \mathbb{Z}^{d\times d}$ suitable for Hadamard's bound, natural division is significantly faster in practice. For this reason, Algorithm 2 uses natural division. As shown in Corollary 2, the determinants of A and $\tilde{A}$ are related by $\det(\tilde{A}) = (\prod_{i=0}^{n-1} \mathrm{lc}(\check{M}_{n-i})^{\Delta_i}) \det(A)$ for some $\Delta_i \in \mathbb{Z}$. Since $p \nmid \prod_{i=1}^{n} \mathrm{lc}(M_i, z_i)$, we have $|\det(A)| < |\det(\tilde{A})|$. Therefore, to bound $|\det(A)|$, it suffices to bound $|\det(\tilde{A})|$. To do this, we need to bound the entries of $\tilde{A}$. Among the vectors$[\gamma^j]_{B_L}$, the largest entries occur in $[\gamma^{d-1}]_{B_L}$. Moreover, $\|\gamma^{d-1} \text{ prem}\langle \check{M}_n, \ldots, \check{M}_1\rangle\|_\infty < \|\gamma^d \text{ prem}\langle \check{M}_n, \ldots, \check{M}_1\rangle\|_\infty$. Thus, by bounding the $\|\gamma^d \text{ prem}\langle \check{M}_n, \ldots, \check{M}_1\rangle\|_\infty$, we can use it as an upper bound for $\tilde{A}_{i,j}$. Note that $\deg(\gamma^j, z_i) = j$ for $1 \leq i \leq n$. We present Lemma 1 without proof.

Lemma 1. *Let $f, g \in \mathbb{Z}[z_1, \ldots, z_n]$ and let $\check{M}_i = l_i z_i^{d_i} + \sum_{j=1}^{d_i-1} a_{i,j} z_i^j$ be the minimal polynomial of α_i where $a_{i,j} \in \mathbb{Z}[z_1, \ldots, z_{i-1}]$. we have,*

(i) $\|fg\|_\infty \leq \|f\|_\infty \|g\|_\infty \min(\#f, \#g)$.
(ii) $\deg(a_{i,j}, z_k) \leq d_k - 1$ *for* $1 \leq k \leq i-1$ *and* $\#a_{i,j} \leq \prod_{j=1}^{i-1} d_j = \frac{d}{d_i d_{i+1} \cdots d_n} < d$.

Notations 3. *We adopt the following terminology throughout this section:*

- $\gamma = z_1 + C_1 z_2 + \ldots + C_{n-1} z_n$ *where* $0 \neq C_i \in \mathbb{Z}$ *for* $1 \leq i \leq n-1$.
- $d_i = \deg(\check{M}_i, z_i)$.
- $D_i = \frac{d}{\prod_{j=1}^{i} d_{n-j+1}} = \prod_{j=1}^{n-i} d_j$.

In Theorem 4, we bound $\|\text{prem }(\gamma^d, \check{M}_n, z_n)\|_\infty$, the pseudo-remainder of γ^d divided by the polynomial $\check{M}_n$ w.r.t z_n.

Theorem 4. *Let* $f = \gamma^d$ *and* $\tilde{r} = \text{prem }(f, \check{M}_n, z_n)$ *where* $\check{M}_n = l_n z_n^{d_n} + \sum_{j=0}^{d_n-1} a_j z_n^j$ *such that* $l_n \in \mathbb{Z}$ *and* $a_j \in \mathbb{Z}[z_1, \ldots, z_{n-1}]$ *for* $0 \leq j \leq d_n - 1$. *Let* $\delta = d - d_n + 1$ *be the maximum number of division steps. Then,*

(i) $\deg(\tilde{r}, z_n) \leq d_n - 1$ *and* $\deg(\tilde{r}, z_i) \leq d + \delta(d_i - 1)$, *for* $1 \leq i \leq n-1$.
(ii) $\|\tilde{r}\|_\infty \leq \|f\|_\infty (l + d/d_n \|\check{M}_n\|_\infty)^\delta$.

Proof. Since $f = \gamma^d$, We have $\deg(f, z_i) = d$ for $1 \leq i \leq n$. Let $f = \sum_{i=0}^{d} f_i z_n^i$ such that $f_i \in \mathbb{Z}[z_1, \ldots, z_{n-1}]$ for $1 \leq i \leq d$.

(i) $\text{pquo}(f, \check{M}_n, z_n)$ has degree $d - d_n$ so the pseudo-division of f by $\check{M}_n$ has up to $\delta = d - d_n + 1$ steps. In the first step of the pseudo division, we have $\tilde{r}_1 = l_n f - f_d z_n^{d-d_n} \check{M}_n$. Hence, $\deg(\tilde{r}_1, z_n) \leq d-1$. Moreover, for $1 \leq i \leq n-1$, we have $\deg(f_d, z_i) \leq \deg(f, z_i) = d$ and $\deg(\check{M}_n, z_i) \leq d_i - 1$. Consequently,

$$\begin{aligned}\deg(\tilde{r}_1, z_i) &= \max\{\deg(f, z_i), \deg(f_d, z_i) + \deg(\check{M}_n, z_i)\} \\ &\leq \deg(f, z_i) + \deg(\check{M}_n, z_i) \leq d + 1(d_i - 1).\end{aligned}$$

If $\deg(\tilde{r}_1, z_n) \geq d_n$, we continue the division. Let $b_1 = \text{lc}(\tilde{r}_1, z_n)$ and $\deg(\tilde{r}_1, z_n) = d - 1$. In the second division step, we have $\tilde{r}_2 = l_n \tilde{r}_1 - b_1 z_n^{d-d_n-1} \check{M}_n$. Thus, $\deg(\tilde{r}_2, z_n) \leq \deg(\tilde{r}_1, z_n) - 1 \leq d - 2$ and

$$\deg(\tilde{r}_2, z_i) \leq \deg(\tilde{r}_1, z_i) + \deg(\check{M}_n, z_i) \leq d + 2(d_i - 1).$$

Since the division algorithm has at most δ steps, in the last step, we have $\deg(\tilde{r}, z_n) \leq d - \delta = d_n - 1$ and $\deg(\tilde{r}, z_i) \leq d + \delta(d_i - 1)$.

(ii) In the first step of dividing f by $\check{M}_n$ using pseudo division, we have $\tilde{r}_1 = l_n f - f_d z^{d-d_n} \check{M}_n$. Thus, $\|\tilde{r}_1\|_\infty \leq \|l_n f\|_\infty + \|f_d \check{M}_n\|_\infty$. To compute an upper bound for $\|f_d \check{M}_n\|_\infty$, it is sufficient to compute an upper bound for $\|f_d a_j\|_\infty$ where $a_j \in \mathbb{Z}[z_1, \ldots, z_{n-1}]$. Using part (ii) of Lemma 1, we have $T_{a_j} < d/d_n$ which implies that $\|f_d a_j\|_\infty \leq \|f_d\|_\infty \|\check{M}_n\|_\infty \min(T_{a_j}, T_{f_d}) \leq d/d_n \|f_d\|_\infty \|\check{M}_n\|_\infty$ for $1 \leq j \leq d_n - 1$. Moreover, since $\|f_d\|_\infty \leq \|f\|_\infty$, we obtain

$$\|\tilde{r}_1\|_\infty \leq l_n \|f\|_\infty + \|f_d \check{M}_n\|_\infty \leq \|f\|_\infty (l_n + d/d_n \|\check{M}_n\|_\infty).$$

Furthermore, $\deg(\tilde{r}_1, z_n) \leq d - 1$. If $\deg(\tilde{r}_1, z_n) \geq d_n$, we continue the division. In the second division step, we have $\tilde{r}_2 = l_n \tilde{r}_1 - b_1 z_n^{d-d_n-1} \check{M}_n$ where $b_1 = \text{lc}(\tilde{r}_1, z_n)$. Since $\|b_1\|_\infty \leq \|\tilde{r}_1\|_\infty$, using the same strategy as the first division step, we have

$$\begin{aligned}\|\tilde{r}_2\|_\infty &\leq l_n \|\tilde{r}_1\|_\infty + \|b_1 \check{M}_n\|_\infty \leq l_n \|\tilde{r}_1\|_\infty + d/d_n \|\tilde{r}_1\|_\infty \|\check{M}_n\|_\infty \\ &\leq \|\tilde{r}_1\|_\infty (l_n + d/d_n \|\check{M}_n\|_\infty) \leq \|f\|_\infty (l_n + d/d_n \|\check{M}_n\|_\infty)^2.\end{aligned}$$

Continuing this argument, the result is obtained.

In Theorem 4, we bound $\|\text{prem}\ (\gamma^d, \check{M}_n, z_n)\|_\infty$. In the following theorem, Theorem 5, we apply Theorem 4 to bound $\|\gamma^d\ \text{prem}\langle \check{M}_n, \dots, \check{M}_1\rangle\|_\infty$.

Theorem 5. *Let* $f = \gamma^d \in \mathbb{Z}[z_1, \dots, z_n]$ *and* $\check{M}_i = l_i z_i^{d_i} + \sum_{j=0}^{d_i-1} b_{i,j} z_i^j$ *such that* $l_i \in \mathbb{Z}$ *and* $b_{i,j} \in \mathbb{Z}[z_1, \dots, z_{i-1}]$ *for* $1 \le i \le n$ *and* $0 \le j \le d_i - 1$. *Let* $\tilde{r} = f\ \text{prem}\langle \check{M}_n, \dots, \check{M}_1\rangle$. *Then* $\|\tilde{r}\|_\infty \le \|f\|_\infty \prod_{i=1}^n (l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i}$ *where* $\delta_1 = d - d_n + 1$, *and* $\delta_i = d - d_{n-i+1} + 1 + (d_{n-i+1} - 1)\sum_{j=1}^{i-1} \delta_j$ *for* $2 \le i \le n$.

Proof. Since $f = \gamma^d$, we have $\deg(f, z_i) \le d$ for $1 \le i \le n$. Let $\tilde{r}_1 = \text{prem}\ (f, \check{M}_n, z_n)$ and $\delta_1 = d - d_n + 1$ be the maximum number of division steps. From Theorem 4, we have

$$\|\tilde{r}_1\|_\infty \le \|f\|_\infty (l_n + D_1 \|\check{M}_n\|_\infty)^{\delta_1}. \tag{2}$$

Let $\tilde{r}_2 = \text{prem}\ (\tilde{r}_1, \check{M}_{n-1}, z_{n-1})$. From Theorem 4 part (i), we have $\deg(\tilde{r}_1, z_{n-1}) \le d + \delta_1(d_{n-1} - 1)$ and $\deg(\tilde{r}_1, z_{n-1}) - d_{n-1} + 1 \le d - \delta_1(d_{n-1} - 1) - d_{n-1} + 1$. Thus, $\delta_2 = d - \delta_1(d_{n-1} - 1) - d_{n-1} + 1$ is the maximum number of division steps. Let $\check{M}_{n-1} = l_{n-1} z_{n-1}^{d_{n-1}} + \sum_{j=0}^{d_{n-1}-1} b_{n-1_j} z_{n-1}^j$ such that $l_{n-1} \in \mathbb{Z}$ and $b_{n-1,j} \in \mathbb{Z}[z_1, \dots, z_{n-2}]$ for $0 \le j \le d_{n-1} - 1$. Hence, applying Lemma 1, we have $\#b_{n-1,j} \le D_2 = \frac{d}{d_n d_{n-1}}$. Using the same strategy as the proof of part (ii) of Theorem 4, we have

$$\begin{aligned} \|\tilde{r}_2\|_\infty \le& \|\tilde{r}_1\|_\infty (l_{n-1} + D_2 \|\check{M}_{n-1}\|_\infty)^{\delta_2} \\ \le& \underbrace{\|f\|_\infty (l_n + D_1 \|\check{M}_n\|_\infty)^{\delta_1}}_{\text{According to } \textit{Equation } 2} (l_{n-1} + D_2 \|\check{M}_{n-1}\|_\infty)^{\delta_2}. \end{aligned}$$

The result is obtained by repeating this process for polynomials $\check{M}_{n-2}, \dots, \check{M}_1$.

Corollary 1. *Let* $\tilde{r} = \gamma^d\ \text{prem}\ \langle \check{M}_n, \dots, \check{M}_1\rangle$, $l_i = \text{lc}(\check{M}_i, z_i)$, *and* δ_i *be as defined in Theorem 5. If* $\|\gamma^d\|_\infty \le 2^C$, *then*

$$|\ \tilde{A}_{i,j}\ | \le \|\tilde{r}\|_\infty \le 2^C \prod_{i=1}^n (l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i}.$$

Proof. This is a direct result from Theorem 5.

Theorem 6. *Let* δ_i *be as defined in Theorem 5. Let* $p \in \mathbb{P}_b$ *be chosen randomly,* $\|\gamma^d\|_\infty \le 2^C$, *and* $l_i = \text{lc}(\check{M}_i, z_i)$ *then*

$$\text{Prob}[p|\det(\tilde{A})] \le \frac{\lfloor d/2 \log_2 d + d(C + \sum_{i=1}^n \delta_i \log_2(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty))\rfloor}{b\ |\ \mathbb{P}_b\ |}$$

Proof. To bound $\text{Prob}[p \mid \det(\tilde{A})]$, we first bound $|\ \det(\tilde{A})\ |$. From Theorem 3 and Corollary 1, $|\ \det(\tilde{A})\ | \le d^{d/2}(2^C \prod_{i=1}^n (l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i})^d$. Since $p \in \mathbb{P}_b$, we have $\log_2 p < b$. Thus,

$$\begin{aligned}\mathrm{Prob}[p|\det(\tilde{A})] &\leq \frac{\lfloor \frac{\log_2|\det(\tilde{A})|}{\log_2 2^b} \rfloor}{\mid \mathbb{P}_b \mid} \\ &\leq \frac{\lfloor d/2\log_2 d + dC + d\sum_{i=1}^{n} \delta_i \log_2(l_{n-i+1} + D_i\|\check{M}_{n-i+1}\|_\infty)\rfloor}{b \mid \mathbb{P}_b \mid}\end{aligned}$$

Lemma 2. *Let $f_1, f_2 \in R[x]$ be non-zero polynomials with $\mathrm{lc}(f_2)$ a unit, and $0 \leq \deg(f_2) \leq \deg(f_1)$. Let $f_1 = f_2 q(x) + r(x)$ be the natural division with $r(x) = 0$ or $\deg(r(x)) < \deg(f_2(x))$. Suppose that $\tilde{r} = \mathrm{prem}(f_1, f_2)$ and $\tilde{q} = \mathrm{pquo}(f_1, f_2)$. Then, $\tilde{r} = \mathrm{lc}(f_2)^\delta r$ and $\tilde{q} = \mathrm{lc}(f_2)^\delta q$.*

Proof. Multiplying both sides of the equation $r = f_1 - f_2 q$ by $\mathrm{lc}(f_2)^\delta$, we have

$$\mathrm{lc}(f_2)^\delta r = \mathrm{lc}(f_2)^\delta f_1 - f_2(\mathrm{lc}(f_2)^\delta q). \tag{3}$$

Subtracting Equation 3 from $\tilde{r} = \mathrm{lc}(f_2)^\delta f_1 - f_2\tilde{q}$, we have $\tilde{r} - \mathrm{lc}(f_2)^\delta r = f_2(\tilde{q} - \mathrm{lc}(f_2)^\delta q)$. Since $\deg(r), \deg(\tilde{r}) < \deg(f_2)$ and $f_2 \neq 0$, we must have $\tilde{q} - \mathrm{lc}(f_2)^\delta q = 0$ which implies that $\tilde{q} = \mathrm{lc}(f_2)^\delta q$ and $\tilde{r} = \mathrm{lc}(f_2)^\delta r$.

Theorem 7. *Let $\tilde{r} = \gamma^d$ prem $\langle \check{M}_n, \ldots, \check{M}_1 \rangle$ and $r = \gamma^d$ rem $\langle \check{M}_n, \ldots, \check{M}_1 \rangle$. Let $r_0 = \tilde{r}_0 = \gamma^d$, $r_i =$ rem $(r_{i-1}, \check{M}_{n-i+1}, z_{n-i+1})$, $\tilde{r}_i =$ prem $(\tilde{r}_{i-1}, \check{M}_{n-i+1}, z_{n-i+1})$ for $1 \leq i \leq n$. Then, $\tilde{r} = r \prod_{i=0}^{n} \mathrm{lc}(\check{M}_{n-i})^{\Delta_i}$ where $\Delta_i = d_{r_i} - d_{n-i} + 1$ such that $d_{r_i} = \deg(r_i, z_{n-i}) = \deg(\tilde{r}_i, z_{n-i})$ for $0 \leq i \leq n$.*

Proof. We compute $\tilde{r}$ and r step by step in parallel. First, $\tilde{r}_1 =$ prem $(\tilde{r}_0, \check{M}_n, z_n)$ and $r_1 =$ rem $(r_0, \check{M}_n, z_n)$. By Lemma 2, $\tilde{r}_1 = \mathrm{lc}(\check{M}_n)^{\Delta_0} r_1$. Next,

$$\tilde{r}_2 = \text{prem } (\tilde{r_1}, \check{M}_{n-1}) = \mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\tilde{r}_1 - \check{M}_{n-1}\tilde{q}_2 \tag{4}$$

$$r_2 = \text{rem } (r_1, \check{M}_{n-1}) = r_1 - \check{M}_{n-1} q_2. \tag{5}$$

Multiplying Equation 5 by $\mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\mathrm{lc}(\check{M}_n)^{\Delta_0}$ we have

$$\mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\mathrm{lc}(\check{M}_n)^{\Delta_0} r_2 = \mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\tilde{r}_1 - \mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\mathrm{lc}(\check{M}_n)^{\Delta_0}\check{M}_{n-1} q_2.$$

Subtracting Equation 4 from above, we have

$$\mathrm{lc}(\check{M}_n)^{\Delta_0}\mathrm{lc}(\check{M}_{n-1})^{\Delta_1} r_2 - \tilde{r}_2 = \check{M}_{n-1}(\mathrm{lc}(\check{M}_n)^{\Delta_0}\mathrm{lc}(\check{M}_{n-1})^{\Delta_1} q_2 - \tilde{q}_2).$$

Since $\deg(r_2)$ and $\deg(\tilde{r}_2) < \deg(\check{M}_{n-1})$ and $\check{M}_{n-1} \neq 0$, we have

$$\mathrm{lc}(\check{M}_n)^{\Delta_0}\mathrm{lc}(\check{M}_{n-1})^{\Delta_1} q_2 - \tilde{q}_2 = 0$$

which implies that $\tilde{q}_2 = \mathrm{lc}(\check{M}_n)^{\Delta_0}\mathrm{lc}(\check{M}_{n-1})^{\Delta_1} q_2$ and $\tilde{r}_2 = \mathrm{lc}(\check{M}_{n-1})^{\Delta_1}\mathrm{lc}(\check{M}_n)^{\Delta_0} r_2$. Continuing this argument, in the last division we have $\tilde{r}_n = \prod_{i=0}^{n} \mathrm{lc}(\check{M}_{n-i})^{\Delta_i} r_n$. By construction, $\tilde{r} = \tilde{r}_n$ and $r = r_n$.

Corollary 2. $\det(\tilde{A}) = \prod_{i=0}^{n} \mathrm{lc}(\check{M}_{n-i})^{\Delta_i} \det(A)$ *where Δ_i is defined in Theorem 7.*

Proof. Follows directly from Theorem 7.

Theorem 8. *Let C, δ_i, and l_i be as defined in Theorem 6. Let $p \in \mathbb{P}_b$ be chosen randomly such that $p \nmid \prod_{i=0}^{n} \mathrm{lc}(\check{M}_{n-i})$ for $1 \le i \le n$. Then*

$$\mathrm{Prob}[p \mid \det(A)] \le \frac{\lfloor (d/2 \log_2 d + d(C + \sum_{i=1}^{n} \delta_i \log_2(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)))\rfloor}{b \mid \mathbb{P}_b \mid}.$$

Proof. From Corollary 2, $\det(\tilde{A}) = \prod_{i=0}^{n} \mathrm{lc}(\check{M}_{n-i})^{\Delta_i} \det(A)$ where $\Delta_i \in \mathbb{Z}$. Since $p \nmid \prod_{i=0}^{n} \mathrm{lc}(\check{M}_i)$ for $1 \le i \le n$, we have $\mathrm{Prob}[p \mid \det(\tilde{A})] = \mathrm{Prob}[p \mid \det(A)]$. Thus,

$$\begin{aligned}\mathrm{Prob}[p \mid \det(A)] &= \mathrm{Prob}[p \mid \det(\tilde{A})] \\ &\le \frac{\lfloor (d/2 \log_2 d + d(C + \sum_{i=1}^{n} \delta_i \log_2(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)))\rfloor}{b \mid \mathbb{P}_b \mid}.\end{aligned}$$

Now we can get a bound for $\|M(z)\|_\infty$ where $M(z)$ is the characteristic polynomial obtained from Algorithm 2.

Theorem 9. *Let $M(z)$ be the characteristic polynomial obtained from Algorithm 2. Define $B_M = d^{d/2}(2^C \prod_{i=1}^{n}(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i})^d$, where C and l_i are from Theorem 6. Then $\|M(z)\|_\infty \le B_M$.*

Proof. To construct $M(z)$, we solve the linear system $Aq = -[\gamma^d]_{B_L}$ for $q \in \mathbb{Q}^d$ by Cramer's rule, $q_k = \frac{\det(A^{(k)})}{\det(A)}$, where $A^{(k)}$ is the matrix formed by replacing the k-th column of A by $[\gamma^d \text{ rem } \langle \check{M}_n, \ldots, \check{M}_1 \rangle]_{B_L}$ for $1 \le k \le d$. Thus, the largest entries of $A^{(k)}$ appear in the k-th column. Applying the same justification as Theorem 8, $\mid \det(A^{(k)}) \mid \le \mid \det(\tilde{A}^{(k)}) \mid$. Using Theorem 3, we have

$$\mid \det(A^{(k)}) \mid \le \prod_{i=1}^{d} \sqrt{\sum_{j=1}^{d} \tilde{A}_{j,i}^{(k)2}} \le d^{d/2}(\underbrace{2^C \prod_{i=1}^{n}(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i}}_{\text{Corollary 1}})^d.$$

Since $\check{M}_i(z_i) \in \mathbb{Z}[z_1, \ldots, z_i]$ for $1 \le i \le n$, we have $M(z) \in \mathbb{Z}[z]$ which implies that $\det(A) \mid \det(A^{(k)})$. Thus, $q_k \in \mathbb{Z}$ and $q_k \le \mid \det(A^{(k)}) \mid \le d^{d/2}(2^C \prod_{i=1}^{n}(l_{n-i+1} + D_i \|\check{M}_{n-i+1}\|_\infty)^{\delta_i})^d$.

3.3 Zero-Divisor Prime and Evaluation Point

PGCD runs over $\bar{L}_p$. Since $\bar{L}_p$ is not a field, it is possible that PGCD encounters a zero-divisor while trying to compute a gcd in lines 4, 6, 7, 8, 10, and 30. In this section, we bound the probability that PGCD encounters a zero-divisor.

Definition 6. *Let $f_1, f_2 \in \bar{L}_{\mathbb{Z}}[x_2, \ldots, x_k][x_1]$. Let p be a prime and $\beta \in [0, p)^{(k-1)}$ be an evaluation point such that (p, β) is not an lc-bad pair. We call the ordered pair (p, β) a **zero-divisor pair** if Algorithm URES, Algorithm 3, returns FAIL for the inputs $\phi_p(f_1)(\beta)$ and $\phi_p(f_2)(\beta) \in \bar{L}_p[x_1]$.*

Example 8. Let $f_1 = (x+1)w^3 + xz$ and $f_2 = (x+y+8z)w + zyx$ be two polynomials in $\mathbb{Z}[z]/\langle z^2\rangle[x,y][w]$. The ordered pair $(p,\beta) = (7,(0,0))$ is a zero-divisor unit since $\mathrm{lc}(f_2,w)(\beta) \mod 7 = z$ is not invertible over $\mathbb{Z}_7[z]/\langle z^2\rangle[x,y][w]$. Consequently, Algorithm URES returns FAIL when attempting to make f_2 monic.

Before bounding the probability of hitting a zero-divisor pair, we must first define the subresultant polynomial remainder sequence (s.p.r.s.) and establish its connection to the m.p.r.s.. This requires clarifying the relationships between prem (f_1,f_2), rem (f_1,f_2), and mrem(f_1,f_2). We present Lemma 3 and Lemma 4 without proof.

Lemma 3. *Let $f_1, f_2 \in R[x]$ be non-zero polynomials with $\mathrm{lc}(f_2)$ and $\mathrm{lc}(f_1)$ units, and $0 \le \deg(f_2) \le \deg(f_1)$. Let $\delta = \deg(f_1) - \deg(f_2) + 1$. Then,*

(i) $\mathrm{rem}\,(f_1,f_2) = \mathrm{lc}(f_1)\mathrm{mrem}(f_1,f_2)$ *and* $\mathrm{quo}(f_1,f_2) = \mathrm{lc}(f_2)^{-1}\mathrm{lc}(f_1)\mathrm{mquo}(f_1,f_2)$.
(ii) $\mathrm{prem}\,(f_1,f_2) = \mathrm{lc}(f_2)^{\delta}\mathrm{rem}\,(f_1,f_2)$ *and* $\mathrm{pquo}(f_1,f_2) = \mathrm{lc}(f_2)^{\delta}\mathrm{quo}(f_1,f_2)$.
(iii) $\mathrm{prem}\,(f_1,f_2) = \mathrm{lc}(f_2)^{\delta}\mathrm{lc}(f_1)\mathrm{mrem}(f_1,f_2)$ *and*
$\mathrm{pquo}(f_1,f_2) = \mathrm{lc}(f_2)^{\delta-1}\mathrm{lc}(f_1)\mathrm{mrem}(f_1,f_2)$.

Lemma 4. *Let $f_1, f_2 \in R[x]$ be non-zero polynomials with $\mathrm{lc}(f_2)$ a unit. Let $a, b \in R$ be units and $\delta = \deg(f_1) - \deg(f_2) + 1$. Then,*

(i) $\mathrm{rem}\,(af_1,bf_2) = a\cdot\mathrm{rem}\,(f_1,f_2)$ *and* $\mathrm{quo}(af_1,bf_2) = \frac{a}{b}\mathrm{quo}(f_1,f_2)$.
(ii) $\mathrm{prem}\,(af_1,bf_2) = ab^{\delta}\cdot\mathrm{prem}\,(f_1,f_2)$ *and* $\mathrm{pquo}(af_1,bf_2) = ab^{\delta-1}\mathrm{pquo}(f_1,f_2)$.
(iii) $\mathrm{mrem}(af_1,bf_2) = \mathrm{mrem}(f_1,f_2)$ *and* $\mathrm{mquo}(af_1,bf_2) = \mathrm{mquo}(f_1,f_2)$.

We are interested in a Polynomial Remainder Sequence (p.r.s.) that avoids the appearance of fractions in the remainders. The Subresultant Polynomial Remainder Sequence (s.p.r.s.) is such a sequence. We present two ways of defining the subresultants. The first way, Algorithm 4, uses pseudo-division for univariate polynomials, and the second one, Definition 7, uses determinants. Let $S_1, S_2, S_3, \ldots, S_k$ be the s.p.r.s. obtained from Algorithm 4. Note that the last subresultant is $S_k = \mathrm{res}(f_1,f_2,y)$. Theorem 10 presents a connection between the remainders obtained from Algorithm 3, m.p.r.s. and Algorithm 4, s.p.r.s..

Theorem 10. *Let $f_1, f_2 \in R[x]$ be non-zero polynomials such that $\deg(f_2) \le \deg(f_1)$. Suppose that Algorithm 3 and 4 do not fail for f_1 and f_2. Let $r_1, r_2, \ldots, r_l$ denote the m.p.r.s. from Definition 1 and let $S_1, \ldots, S_k$ be the s.p.r.s. from Algorithm 4. Let $d_i = \deg(S_i)$ for $1 \le i \le k$, we have,*

$$\begin{cases} S_1 &= r_1 \\ S_2 &= r_2 \\ S_3 &= (-\mathrm{lc}(S_2))^{d_1-d_2+1} r_3 \\ S_4 &= \frac{(-\mathrm{lc}(S_3))^{d_2-d_3+1}}{\mathrm{lc}(S_2)^{(d_1-d_2)(d_2-d_3)}} r_4 \quad \text{if } \deg(\mathrm{S_1}) - 1 \neq \deg(\mathrm{S_2}) \\ S_i &= \frac{(-\mathrm{lc}(S_{i-1}))^{d_{i-2}-d_{i-1}+1}}{\mathrm{lc}(S_{i-2})^{d_{i-2}-d_{i-1}}} r_i \quad \text{if } \deg(\mathrm{S_{i-3}}) - 1 = \deg(\mathrm{S_{i-2}}) \text{ for i} \ge 4 \\ S_i &= \frac{(-\mathrm{lc}(S_{i-1}))^{d_{i-2}-d_{i-1}+1}\mathrm{lc}(S_{i-3})^{(d_{i-3}-d_{i-2}-1)(d_{i-2}-d_{i-1})}}{\mathrm{lc}(S_{i-2})^{(d_{i-3}-d_{i-2})(d_{i-2}-d_{i-1})}} r_i \text{ if } \deg(\mathrm{S_{i-3}}) - 1 \neq \deg(\mathrm{S_{i-2}}) \text{ for i} > 4. \end{cases}$$

Algorithm 4: s.p.r.s. Algorithm

Input: $f_1, f_2 \in R[y]$, non-zero polynomials such that $\deg(f_1) \geq \deg(f_2) \geq 0$
Output: Either the s.p.r.s. generated by f_1, f_2, $S = S_1, S_2, \ldots, S_k$, or FAIL

$m, n = \deg(f_1), \deg(f_2)$
$S_1, S_2 = f_1, f_2$
if $\deg(f_1) = 0$ *and* $\deg(f_2) = 0$ **then**
  return$(S_1, S_2, 1)$
if $\deg(f_1) \neq 0$ *and* $\deg(f_2) = 0$ **then**
  return(S_1, S_2, f_2^m)
$c, r = 1, n$
$j, i = m - 1, 2$`// i counts the number of subresultants`
$S = S_1, S_2$
while $r \neq 0$ **do**
  $r = \deg(S_i)$
  if $r = 0$ **then**
    return(S)
  if *c is not invertible in R* **then**
    return*(FAIL)*
  $S_{i+1} = \text{prem}(S_{i-1}, S_i)/(-c)^{j-r+2}$
  if $j \neq r$ **then**
    $S_i = (\text{lc}(S_i)^{j-r} S_i)/c^{j-r}$
  $S = S, S_{i+1}$
  $j = r - 1$
  $c = \text{lc}(S_i)$
  $i = i + 1$
return(S);

Proof. From Definition 1 and Algorithm 4, we have $r_1 = S_1 = f_1$ and $r_2 = S_2 = f_2$. Comparing the iterations of Algorithm 1 with the iterations of Algorithm 4, we prove the theorem. In the first iteration of Algorithm 1, we have $r_3 = M_1 - M_2 q_3 = f_1 - \text{lc}(f_2)^{-1} f_2 q_3$. From Lemma 4, part (i), we have $r_3 = \text{rem}\ (f_1, f_2)$. In the first iteration of Algorithm 4, we have $j = d_1 - 1$, $c = 1$, and $r = \deg(f_2) = d_2$. We set $S_3 = \frac{\text{prem}\ (S_1, S_2)}{(-c)^{j-r+2}} = \frac{\text{prem}\ (f_1, f_2)}{(-1)^{d_1 - d_2 + 1}}$. From Lemma 2, we have prem $(f_1, f_2) = \text{lc}(f_2)^{d_1 - d_2 + 1} r_3$ which implies that $S_3 = (-\text{lc}(S_2))^{d_1 - d_2 + 1} r_3$. If $j \neq r$, we set $S_2 = \text{lc}(f_2)^{d_1 - d_2 - 1} f_2$. In the second iteration of Algorithm 1, we have $r_4 = M_2 - M_3 q_4$ where $M_2 = \text{monic}(f_2)$ and $M_3 = \text{monic}(r_3)$. Apploying part (i) of Lemma 3, we have rem $(f_2, r_3) = \text{lc}(f_2) r_4$. In the second iteration of Algorithm 4, we have $i = 3$, $j = d_2 - 1$, $r = \deg(S_3)$, and $c = \text{lc}(S_2)$. Two cases are possible for S_4. The first case happens if in the first iteration $r = j$. In this case, $S_4 = \frac{\text{prem}\ (S_2, S_3)}{(-c)^{j-r+2}}$. Employing Lemma 3, part (iii), we have prem $(S_2, S_3) = \text{lc}(S_2)\text{lc}(S_3)^{d_2 - d_3 + 1} \text{mrem}(S_2, S_3) = \text{lc}(S_2)\text{lc}(S_3)^{d_2 - d_3 + 1} r_4$. Thus, $S_4 = \frac{\text{prem}\ (S_2, S_3)}{(-c)^{j-r+2}} = \frac{(-\text{lc}(S_3))^{d_2 - d_3 + 1}}{(\text{lc}(S_2))^{d_2 - d_3}} r_4$. The second case happens when $r \neq j$ in the first iteration. In this case, we replace

S_2 by $\mathrm{lc}(S_2)^{d_1-d_2-1}S_2$. Hence, $c = \mathrm{lc}(S_2)^{d_1-d_2}$. Using Lemma 4, part (ii), we have prem $(\mathrm{lc}(S_2)^{d_1-d_2-1}S_2, S_3) = \mathrm{lc}(S_2)^{d_1-d_2-1}$prem (S_2, S_3). Moreover, using Lemma 3, part (iii), we have prem $(S_2, S_3) = \mathrm{lc}(S_2)\mathrm{lc}(S_3)^{d_2-d_3+1}r_4$. Hence,

$$\begin{aligned} S_4 &= \frac{\text{prem } (\mathrm{lc}(S_2)^{d_1-d_2-1}S_2, S_3)}{(-c)^{j-d_3+2}} = \frac{\mathrm{lc}(S_2)^{d_1-d_2-1}\text{prem } (S_2, S_3)}{(-(\mathrm{lc}(S_2)^{d_1-d_2}))^{d_2-d_3+1}} \\ &= \frac{\mathrm{lc}(S_2)^{d_1-d_2-1}\mathrm{lc}(S_2)\mathrm{lc}(S_3)^{d_2-d_3+1}r_4}{(-(\mathrm{lc}(S_2)^{d_1-d_2}))^{d_2-d_3+1}} = \frac{(-\mathrm{lc}(S_3))^{d_2-d_3+1}}{(\mathrm{lc}(S_2)^{d_1-d_2})^{d_2-d_3}}r_4. \end{aligned}$$

Employing the same argument for $i > 4$, the result will be obtained.

Example 9. Let $f_1 = 24x^6 + 12x^5z + 8x^4 + 2x^3z + 8x^2 + xz + 4$ and $f_2 = 8x^6 + 8x^5z + 4x^4 + 8x^3z + 2xz + 4$ be two polynomials in $L_{\mathbb{Z}}[x] = \mathbb{Z}[z]/\langle z^2 - 2\rangle[x]$. Table 1 demonstrates the s.p.r.s. obtained from Algorithm 4 and m.p.r.s obtained from Algorithm 1 for the input polynomials f_1 and f_2. As seen, the coefficients of the subresultants grow significantly. Particularly, S_8, has the largest coefficient.

Table 1. s.p.r.s.

s.p.r.s.	m.p.r.s.
$S_1 = 24x^6 + 12x^5z + 8x^4 + 2x^3z + 8x^2 + xz + 4$	$r_1 = 24x^6 + 12x^5z + 8x^4 + 2x^3z + 8x^2 + xz + 4$
$S_2 = 8x^6 + 8x^5z + 4x^4 + 8x^3z + 2xz + 4$	$r_2 = 8x^6 + 8x^5z + 4x^4 + 8x^3z + 2xz + 4$
$S_3 = 96x^5z + 32x^4 + 176x^3z - 64x^2 + 40xz + 64$	$r_3 = -12x^5z - 4x^4 - 22x^3z + 8x^2 - 5xz - 8$
$S_4 = -29696x^4 - 3584x^3z + 2560x^2 - 7936xz - 1024$	$r_4 = -29/18x^4 - 7/36zx^3 + 5/36x^2 - 31/72xz - 1/18$
$S_5 = 8765440x^3z - 5480448x^2 + 1642496xz + 3047424$	$r_5 = 1605/841x^3 - 2007/3364zx^2 + 1203/3364x + 279/841z$
$S_6 = 13828096x^2 - 63209472xz + 601096192$	$r_6 = -6119/2289800x^2 + 55941/4579600xz - 66497/572450$
$S_7 = -47448064xz - 4034396160$	$r_7 = -193670/44521x - 8233650/44521z$
$S_8 = 131776013926$	$r_8 = 132583327/32761$

Corollary 3. *Let $f_1, f_2 \in \bar{L}_p[x_1]$ with $f_2 \neq 0$ and $0 \leq \deg(f_2) \leq \deg(f_1)$. Suppose that Algorithm 3 and Algorithm 4 do not fail for f_1 and f_2. Let $r_1, r_2, \ldots, r_l$ be the m.p.r.s. and $S_1, \ldots, S_h$ be the s.p.r.s.. Then,*

(i) $l = h$
(ii) $\mathrm{lc}(r_i, x_1) = u \cdot \mathrm{lc}(S_i, x_1)$ *for a unit* $u \in \bar{L}_p$.
(iii) $\deg(r_i, x_1) = \deg(S_i, x_1)$

Proof. (i) Since Algorithm 3 does not fail for f_1 and f_2, $\mathrm{lc}(r_i)$ is invertible for $1 \leq i \leq l$. Thus, we can alternate natural division with pseudo-division to compute S_i for $1 \leq i \leq l$. According to Theorem 10, since $\mathrm{lc}(r_i)$ is invertible, $\mathrm{lc}(S_i)$ is also invertible for $1 \leq i \leq l$. Thus Algorithm 4 must have the same number of division steps as Algorithm 3 which implies that $l = h$.

(ii) From part (i), Algorithm 4 terminates after l iterations so $\mathrm{lc}(S_i)$ is invertible for $1 \leq i \leq l$. Thus, in Theorem 10 the fractions are units. Accordingly, $\mathrm{lc}(r_i) = u \cdot \mathrm{lc}(S_i)$ for some unit $u \in \bar{L}_p$.

(iii) Since Algorithm 4 terminates after l iterations, $\mathrm{lc}(S_i)$ is invertible for $1 \leq i \leq l$. Thus, from Theorem 10, we have $S_i = u_i r_i$ for a unit u_i. Multiplying by a unit does not change $\deg(r_i)$. Hence, $\deg(r_i) = \deg(S_i)$.

The second way of defining s.p.r.s. is to use determinants.

Definition 7. *Let $f_1 = \sum_{i=1}^{m} a_i x_1^i$ and $f_2 = \sum_{i=1}^{n} b_i x_1^i \in R[x_2, \ldots, x_k][x_1]$ with $0 < n \leq m$. Let $M_{i,j}$ be the $(m+n-2j) \times (m+n-2j)$ matrix determined from $\mathrm{sylv}(f_1, f_2, x_1)$ by deleting rows $n-j+1$ to n, rows $m+n-j+1$ to $m+n$, and columns $m+n-2j$ to $n+m$ except for column $m+n-i-j$. The coefficients with negative subscripts are zero(see Definition 7.3 in [10]). The j-th subresultant of f_1 and f_2 w.r.t. x_1 is the polynomial of degree j defined by*

$$S(j, f_1, f_2, x_1) = \det(M_{0j}) + \det(M_{1j})x_1 + \cdots + \det(M_{jj})x_1^j.$$

Since $\det(M_{ij}) = 0$ for $i > j$, we can present $S(j, f_1, f_2, x_1)$ as

$$S(j, f_1, f_2, x_1) = \det\left(\begin{bmatrix} a_m & a_{m-1} & \cdots & & a_1 & a_{2j-n} & x_1^{n-j-1} f_1 \\ & a_m & a_{m-1} & \cdots & & & \\ & & & \cdots & & & \\ & & & & a_m & a_{j+1} & f_1 \\ b_n & b_{n-1} & \cdots & & b_1 & b_{2j-m} & x_1^{m-j-1} f_2 \\ & b_n & b_{n-1} & \cdots & & & \\ & & & \cdots & & & \\ & & & & b_n & b_{j+1} & f_2 \end{bmatrix}\right).$$

Theorem 11. *Let $S_1, \ldots, S_l$ be the s.p.r.s. obtained from Algorithm 4 for the input polynomials $f_1, f_2 \in R[x_1]$ where $R = \mathbb{Z}[x_2, \ldots, x_k]$ and $\deg(f_2, x_1) \leq \deg(f_1, x_1)$. Let $0 \leq j \leq \deg(f_2, x_1)$ and $S(j, f_1, f_2, x_1) \neq 0$, then there exists $1 \leq i \leq l$ such that $S(j, f_1, f_2, x_1) = S_i$. In particular, $S(0, f_1, f_2, x_1) = S_l$.*

Proof. A direct consequence of the Subresultant Chain Algorithm, page 129, [5].

Theorem 12. *Let $f_1 = \sum_{j=0}^{t} a_j x_i^j$ and $f_2 = \sum_{j=0}^{s} b_j x_i^j$ be non-zero polynomials where $a_j, b_j \in R[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]$ for $0 \leq j \leq s \leq t$. Let $\phi : R[x_1, \ldots, x_k] \longrightarrow \tilde{R}[x_1, \ldots, x_k]$ be a ring homomorphism, $\deg(\phi(f_1), x_i) = d_{1,i}$, and $\deg(\phi(f_2), x_i) = d_{2,i}$.*

(i) If $d_{1,i} = t$ and $0 \leq d_{2,i} \leq s$, then

$$\phi(\mathrm{res}(f_1, f_2, x_i)) = \phi(a_t)^{s-d_{2,i}} \mathrm{res}(\phi(f_1), \phi(f_2), x_i).$$

(ii) If $d_{1,i} < t$ and $d_{2,i} = s$, then

$$\phi(\mathrm{res}(f_1, f_2, x_i)) = (-1)^{s(t-d_{1,i})} \phi(b_s)^{t-d_{1,i}} \mathrm{res}(\phi(f_1), \phi(f_2), x_i).$$

(iii) If $d_{1,i} < t$ and $d_{2,i} < s$, then $\phi(\mathrm{res}(f_1, f_2)) = 0$.

Proof. Parts (i) and (ii) follow similarly to Proposition 6 in Chapter 6 of [8]. For (iii), when $d_{1,i} < t$ and $d_{2,i} < s$, the Sylvester matrix $\phi(\mathrm{sylv}(f_1, f_2, x_i))$ contains at least one column of zeros. Thus, its determinant is zero.

Theorem 13. *Let $f_1, f_2 \in \bar{L}_{\mathbb{Z}}[x_2, \ldots, x_k][x_1]$. Let p be a prime and $\beta \in [0, p)^{k-1}$ such that (p, β) is not lc-bad. Let S_i be among s.p.r.s over $R = \mathbb{Z}[x_2, \ldots, x_k, z]$ and $l_i = \mathrm{lc}(S_i, x_1) \in \mathbb{Z}[x_2, \ldots, x_k][z]$. If the ordered pair (p, β) is a zero-divisor unit, then there exists $i \geq 2$ such that $p \mid \mathrm{res}(l_i(\beta), \check{M}, z)$.*

Proof. By Definition 6, (p, β) is a zero-divisor pair if Algorithm URES returns FAIL for the input polynomials $\phi_p(f_1)(\beta), \phi_p(f_2)(\beta) \in \bar{L}_p[x_1]$. Let r_i be the i-th remainder computed by URES with $i \geq 2$. The algorithm fails iff $\mathrm{lc}(r_i)$ is not invertible in $\bar{L}_p$ for some i. By Theorem 10 and Corollary 3, there exists a unit u_i such that $\phi_p(S_i(\beta)) = u_i r_i$ so $\phi_p(l_i(\beta)) = u_i \cdot \mathrm{lc}(r_i)$. Thus, $\mathrm{lc}(r_i)$ is not invertible in $\bar{L}_p$ iff $\phi_p(l_i(\beta))$ is not invertible. Therefore, URES fails for $\phi_p(f_1)(\beta)$ and $\phi_p(f_2)(\beta)$ iff $\gcd(\phi_p(l_i(\beta)), \phi_p(\check{M}), z) \neq 1$. By Theorem 1, this implies that $\mathrm{res}(\phi_p(l_i(\beta)), \phi_p(\check{M}), z) = 0$. Then, from Theorem 12, $p \mid \mathrm{res}(l_i(\beta), \check{M}, z)$.

Let l_i be as defined in Theorem 13. As a consequence of Theorem 13, we have $\mathrm{Prob}[(p, \beta)$ is a zero-divisor unit$] \leq \mathrm{Prob}[p \mid \mathrm{res}(l_i(\beta), M, z)]$. To bound $\mathrm{Prob}[p \mid \mathrm{res}(l_i(\beta), M, z)]$, we first need an upper bound on the integer $\mathrm{res}(l_i(\beta), M, z)$. By Hadamard's bound, Theorem 3, this requires an upper bound on $\|M(z)\|_\infty$ and $\|l_i(\beta)\|_\infty$. Theorem 9 provides $\|M(z)\|_\infty \leq B_M$, so it remains to bound $\|l_i(\beta)\|_\infty$. Assume $f_1, f_2 \in \mathbb{Z}[X, z][x_1]$ where $X = x_2, \ldots, x_k$. Let $R_{x_1} = \mathrm{res}(f_1, f_2, x_1) \in \mathbb{Z}[X, z]$ and $R_z = \mathrm{res}(R_{x_1}, \check{M}, z) \in \mathbb{Z}[X]$. We summarize some properties of R_{x_1} in Proposition 1 below, without proof.

Proposition 1. *Let $f_1 = \sum_{i=0}^{m} a_i(X, z) x_1^i$ and $f_2 = \sum_{i=0}^{n} b_i(X, z) x_1^i$ be two non-zero polynomials in $\mathbb{Z}[X, z][x_1]$ such that $\deg(f_1, z), \deg(f_2, z) \leq d-1$ where $d = \deg(M, z)$. Let*

- $d_x = \max_{j=2}^{k}(\deg(f_1, x_j), \deg(f_2, x_j))$
- $a_M = \max_{i=0}^{m}(\#a_i(X, z))$, $b_M = \max_{j=0}^{n}(\#b_j(X, z))$, *and* $T_M = \max(a_M, b_M)$
- $H = \max(\|f_1\|_\infty, \|f_2\|_\infty)$

Then we have

(i) $\deg(R_{x_1}, X) \leq (n+m) d_x$,
(ii) $\deg(R_{x_1}, z) \leq (n+m)(d-1)$,
(iii) $\# R_{x_1} \leq (n+m)! T_M^{(n+m)}$,
(iv) and $\|R_{x_1}\|_\infty \leq (n+m)! H^{(n+m)} T_M^{(n+m-1)}$.

For $0 \leq i \leq l$, suppose that $\|S_i\|_\infty \leq B_i$ and $\|R_{x_1}\|_\infty \leq B_l$. Due to coefficient growth in Algorithm 4, we have $B_i \leq B_l$. Moreover, $\|l_i\|_\infty \leq \|S_i\|_\infty$ which implies that $\|l_i\|_\infty \leq B_i \leq B_l$. Accordingly,

$$\mathrm{Prob}[p \mid \mathrm{res}(l_i(\beta), M, z)] \leq \mathrm{Prob}[p \mid \mathrm{res}(R_{x_1}(\beta), M, z)]. \tag{6}$$

From right-hand side of Equation 6, we have

$$\begin{aligned} \mathrm{Prob}[(p, \beta) \text{ is a zero-divisor pair}] &\leq \mathrm{Prob}[R_z(\beta) = 0 \text{ or } R_z(\beta) \neq 0 \text{ and } p \mid R_z(\beta)] \\ &\leq \mathrm{Prob}[R_z(\beta) = 0] + \mathrm{Prob}[p \mid R_z(\beta)]. \end{aligned} \tag{7}$$

To bound $\mathrm{Prob}[R_z(\beta) = 0]$, we apply Schwartz-Zippel lemma as follows.

Lemma 5. *(Schwartz-Zippel lemma) Let R be an integral domain and let $S \subseteq R$ be finite. Let $f \in R[x_1, x_2, \ldots, x_k]$ be a non-zero polynomial with total degree D. Then the number of roots of f in S^n is at most $D|S|^{k-1}$. Hence, if β is chosen at random from S^k, then* $\text{Prob}[f(\beta) = 0] = \frac{D}{|S|}$.

Lemma 6. *Let n, m, and d_x be as defined in Proposition 1. Then* $\text{Prob}[R_z(\beta) = 0] \leq \frac{d(n+m)d_x}{p}$.

Proof. From Proposition 1, we have $\deg(R_{x_1}, z) \leq D$ where $D = (n+m)(d-1)$ and $\deg(R_{x_1}, X) \leq (n+m)d_x$. The Sylvester matrix $\text{sylv}(R_{x_1}, M, z)$ contains d rows of the coefficients of R_{x_1}, which are polynomials in $\mathbb{Z}[X]$, and at most D rows of the coefficients of $M(z)$, which lie in $\mathbb{Z}$. Therefore, $\deg(R_z, X) \leq d(n+m)d_x$. By the Shwartz-Zippel lemma, $\text{Prob}[R_z(\beta) = 0] \leq \frac{d(n+m)d_x}{p}$.

Lemma 7. *Let p be a prime number and $\beta \in [0, p)^{k-1}$. Then $\|R_{x_1}(\beta)\|_\infty \leq B_R$, where $B_R = p^{(n+m)d_x} T_M^{(2n+2m-1)} H^{(n+m)} (n+m)^2!$.*

Proof. From Proposition 1, we have $\deg(R_{x_1}, X) \leq (n+m)d_x$, $\#R_{x_1} \leq (n+m)! T_M^{(n+m)}$, and $\|R_{x_1}\|_\infty \leq (n+m)! H^{(n+m)} T_M^{(n+m-1)}$. Therefore,

$$\begin{aligned}\|R_{x_1}(\beta)\|_\infty &\leq \#R_{x_1} p^{\deg(R_{x_1}, X)} \|R_{x_1}\|_\infty \\ &\leq p^{(n+m)d_x} T_M^{(2n+2m-1)} H^{(n+m)} ((n+m)!)^2.\end{aligned}$$

Lemma 8. *Let $\|M(z)\|_\infty \leq B_M$ (from Theorem 9), and $\|R_{x_1}(\beta)\|_\infty \leq B_R$ (from Lemma 7). Assume that $R_z(\beta) \neq 0$, Then,*

$$\text{Prob}[p \mid R_z(\beta)] \leq \frac{\lfloor (d + (n+m)(d-1))/2 \log(dB_R^2 + (n+m)(d-1)B_M^2) \rfloor}{b \mid \mathbb{P}_b \mid}.$$

Proof. The matrix $S = \text{sylv}(R_{x_1}(\beta), M, z)$ contains d rows of coefficients of $R_{x_1}(z)$ and $\deg(R_{x_1}(\beta), z) < d(n+m)$ rows of coefficients of $M(z)$. Since $\|R_{x_1}(\beta)\|_\infty \leq B_R$ and $\|M(z)\|_\infty \leq B_M$, by Hadamard's bound, we have

$$\begin{aligned}\mid \det(S) \mid = \mid R_z(\beta) \mid &\leq \prod_{i=1}^{d(n+m+1)} \sqrt{\sum_{j=1}^{d(n+m+1)} S_{i,j}^2} \leq \prod_{i=1}^{d(n+m+1)} \sqrt{\sum_{j=1}^{d} B_R^2 + \sum_{j=1}^{d(n+m)} B_M^2} \\ &\leq (dB_R^2 + d(n+m)B_M^2)^{d(n+m+1)/2}.\end{aligned}$$

Hence, $\text{Prob}[p \mid R_z(\beta) \mid \leq \frac{\lfloor (d(n+m+1))/2 \log_2(dB_R^2 + d(n+m)B_M^2) \rfloor}{b|\mathbb{P}_b|}$.

Let $B = \frac{d(n+m)d_x}{p} + \frac{\lfloor (d(n+m+1))/2 \log(dB_R^2 + d(n+m)B_M^2) \rfloor}{b|\mathbb{P}_b|}$ obtained by combining bounds from Lemma 6 and Lemma 8. From Equation 7, we have

$$\text{Prob}[(p, \beta) \text{ is a zero-divisor pair}] \leq B. \tag{8}$$

Definition 8. *Let $f = \sum_{i=1}^{t} a_i(x_k)Y_i(X_k)$ where $a_i \in \bar{L}_p[x_k]$ and Y_i is a monomial in $X_k = x_1, \ldots, x_{k-1}$. Define cont$(f, X_k) = \gcd(a_1, \ldots, a_t) \in \bar{L}_p[x_k]$.*

Let $D = \max_{i=1}^{k}(\deg(f_1, x_i), \deg(f_2, x_i))$, and let $\#\beta$ be the number of points required to interpolate $x_2, \ldots, x_k$ in the gcd. Then, $\#\beta \leq (D+1)^{k-1}$. In PGCD, the probability that any gcd computation in lines 4, 6, 7, 8, 10, or 30 fails is bounded by B as defined in Equation 8. Let $X_i = x_1, \ldots, x_{i-1}$ for $2 \leq i \leq k$. In line 6, to compute $\mathrm{cont}(f_1, X_i)$, $\leq (D+1)^{i-1} - 1$ univariate gcds are computed. But this is done for $\leq (D+1)^{k-i}$ evaluation points for $x_{i+1}, \ldots, x_k$, hence a total of $\leq (D+1)^{k-1}$ gcds. Thus, computing all $\mathrm{cont}(f_1, X_i)$ for $2 \leq i \leq k$ requires $\leq (k-1)(D+1)^{k-1}$ gcd computations. The same applies to lines 7, and 30. Line 4 performs $\leq (D+1)^{k-1}$ gcd computations. Lines 8 and 10 each involve one gcd computation, which we can include in the count for lines 6 and 7. Therefore, Prob[PGCD encounters a zero-divisor] $\leq$ $\underbrace{3(k-1)(D+1)^{k-1}B}_{\text{Lines 6, 7, and 30}} + \underbrace{(D+1)^{k-1}B}_{\text{Line 4}} = 3k(D+1)^{k-1}B$.

3.4 Unlucky primes

Definition 9. *Let $f_1, f_2 \in \bar{L}_{\mathbb{Z}}[x_1, \ldots, x_k]$ be non-zero polynomials, and let $g = \gcd(f_1, f_2)$ be their monic gcd. Let h_1 and h_2 denote the cofactors of f_1 and f_2, respectively. Let p be a prime number such that $p \nmid \mathrm{lc}(f_2)$, $p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i)$, p is not a det-bad prime, and $\gcd(\phi_p(f_1), \phi_p(f_2))$ exists. We call p **unlucky** if $\deg(\gcd(\phi_p(h_1), \phi_p(h_2)) > 0$.*

Example 10. Let $f_1 = (zx+y)(5x+2y+z)$ and $f_2 = (zx+y)(5x+9y+z)$ be polynomials in $\bar{L}_{\mathbb{Z}}[x, y]$ where $\bar{L}_{\mathbb{Z}} = \mathbb{Z}[z]/\langle z^2 - 2\rangle$. By inspection, we have $h_1 = 5x+2y+z$, $h_2 = 5x+9y+z$, and $\gcd(f_1, f_2) = zx+y$. Let $p = 7$. Then $g_p = \gcd(\phi_p(h_1), \phi_p(h_2)) = 5x+2y+z$ and $\deg(g_p) > 0$. Thus, $p = 7$ is an unlucky prime.

Theorem 14. *Let $f_1, f_2 \in \bar{L}_{\mathbb{Z}}[x_1, \ldots, x_k]$ be non-zero polynomials with cofactors h_1 and h_2, respectively. If p is an unlucky prime, then $p \mid \prod_{j=1}^{k} \mathrm{res}(h_1, h_2, x_j)$.*

Proof. Let $g_p = \gcd(\phi_p(h_1), \phi_p(h_2))$. By Definition 9, if p is unlucky, then $\deg(g_p) > 0$, i.e., $g_p \neq 1$. This implies there exists some variable x_i such that $\deg(g_p, x_i) > 0$. Treat $\phi_p(h_1)$ and $\phi_p(h_2)$ as univariate polynomials in x_i over the ring $\bar{L}_{\mathbb{Z}}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]$. Then, by Theorem 1, $g_p \neq 1$ iff $\mathrm{res}(\phi_p(h_1), \phi_p(h_2), x_i) = 0$. By Theorem 12, this implies

$$\mathrm{res}(\phi_p(h_1), \phi_p(h_2), x_i) = 0 \Rightarrow \phi_p(\mathrm{res}(h_1, h_2, x_i)) = 0 \Rightarrow p \mid \mathrm{res}(h_1, h_2, x_i). \quad (9)$$

Since Equation 9 holds for some x_i, we conclude that $p \mid \prod_{j=1}^{k} \mathrm{res}(h_1, h_2, x_j)$.

According to Theorem 14, the set of unlucky primes is finite.

Theorem 15. *Let f_1 and $f_2 \in \bar{L}_{\mathbb{Z}}[x_1,\ldots,x_k]$ be non-zero polynomials with cofactors h_1 and h_2, respectively, and let $g = \gcd(f_1, f_2)$. Define $T_M = \max(\#h_1, \#h_2)$, $t = \max_{i=1}^{k}(\deg(h_1, x_i))$, $s = \max_{i=1}^{k}(\deg(h_2, x_i))$, and $H = \max(\|h_1\|_\infty, \|h_2\|_\infty)$. Let $p \in \mathbb{P}_b$, then*

$$\text{Prob}[p \textit{ is unlucky}] \leq k \frac{\lfloor \log_2 (t+s)! + (t+s-1)\log_2 T_M + (t+s)\log_2 H \rfloor}{b \mid \mathbb{P}_b \mid}$$

Proof. From Theorem 14, $\{p \in \mathbb{P}_b \text{ s.t } p \text{ is unlucky}\} \subseteq \{p \in \mathbb{P}_b \text{ s.t } p \mid \prod_{j=1}^{k} \text{res}(h_1, h_2, x_j)\}$. Thus, for any $p \in \mathbb{P}_b$,

$$\text{Prob}[p \text{ is unlucky}] \leq \text{Prob}[p \mid \prod_{j=1}^{k} \text{res}(h_1, h_2, x_j)] \leq \sum_{i=1}^{k} \text{Prob}[p \mid \text{res}(h_1, h_2, x_i)]$$

From Proposition 1 (iv), $\|\text{res}(h_1, h_2, x_i)\|_\infty \leq (t+s)! T_M^{(t+s-1)} H^{t+s}$ for each $1 \leq i \leq k$. Since $p \in \mathbb{P}_b$, we have $\log_2 p < b$. Therefore, for each $1 \leq i \leq k$,

$$\begin{aligned}\text{Prob}[p \mid \text{res}(h_1, h_2, x_i)] &\leq \frac{\left\lfloor \frac{\log_2 (t+s)! T_M^{(t+s-1)} H^{t+s}}{\log_2 p} \right\rfloor}{\mid \mathbb{P}_b \mid} \\ &\leq \frac{\lfloor \log_2 (t+s)! + (t+s-1)\log_2 T_M + (t+s)\log_2 H \rfloor}{b \mid \mathbb{P}_b \mid}.\end{aligned}$$

Summing over all k variables gives the final bound:

$$\text{Prob}[p \text{ is unlucky}] \leq k \frac{\lfloor \log_2 (t+s)! + (t+s-1)\log_2 T_M + (t+s)\log_2 H \rfloor}{b \mid \mathbb{P}_b \mid}$$

3.5 Unlucky evaluation points

Definition 10. *Let $f_1, f_2 \in \bar{L}_p[x_1,\ldots,x_k]$ with $0 \leq \deg(f_2, x_k) \leq \deg(f_1, x_k)$, and suppose the monic $g = \gcd(f_1, f_2)$ exists. Let h_1 and h_2 denote the cofactors of f_1 and f_2. Let $\beta_k \in [0, p)$ be chosen randomly such that $\text{lc}(f_2)(\beta_k) \neq 0$, and $g_{\beta_k} = \gcd(f_1(x_k = \beta_k), f_2(x_k = \beta_k))$ exists. We call β_k an **unlucky evaluation point** if $\deg(\gcd(h_1(x_k = \beta_k), h_2(x_k = \beta_k))) > 0$.*

Example 11. Let $g = (y+2z)x$, $f_1 = g \cdot (x+z+4y+8)$ and $f_2 = g \cdot (x+2y+z+10)$ be polynomials in $\bar{L}_{11}[x, y]$ listed in the the lexicographic order with $x > y$ where $\bar{L}_{11} = \mathbb{Z}_{11}[z]/\langle z^2+8 \rangle$. Then $\gcd(f_1, f_2) = g$. Choosing $y = 1$, we have $\gcd(h_1(y=1), h_2(y=1)) = x + z + 1$ so $y = 1$ is an unlucky evaluation point.

Theorem 16. *Let f_1 and $f_2 \in \bar{L}_p[x_1,\ldots,x_k]$ be non-zero polynomials, and let monic $g = \gcd(f_1, f_2)$ exists. Let h_1 and h_2 be the cofactors of f_1 and f_2. If $\beta_k \in [0, p)$ is an unlucky evaluation point, then $x_k = \beta_k$ is a root of $\prod_{i=1}^{k-1} \text{res}(h_1, h_2, x_i)$.*

Proof. Let $g_{\beta_k} = \gcd(h_1(x_k = \beta_k), h_2(x_k = \beta_k))$. If $\beta_k \in [0,p)$ is unlucky, then $\deg(g_{\beta_k}) > 0$, i.e., $g_{\beta_k} \neq 1$. Hence, there exists some $1 \leq i \leq k-1$ such that $\deg(g_{\beta_k}, x_i) > 0$. Treat $h_1(x_k = \beta_k)$ and $h_2(x_k = \beta_k)$ as univariate polynomials in x_i, over the ring $\bar{L}_p[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_{k-1}]$. Since $g_{\beta_k} \neq 1$, Theorem 1 implies $\mathrm{res}(h_1(x_k = \beta_k), h_2(x_k = \beta_k), x_i) = 0$. By Theorem 12, $\mathrm{res}(h_1,h_2,x_i)(x_k = \beta_k) = 0$, i.e., $x_k = \beta_k$ is a root of $\prod_{i=1}^{k-1} \mathrm{res}(h_1,h_2,x_i)$.

Theorem 17. *Let f_1 and $f_2 \in \bar{L}_p[x_1,\ldots,x_k]$ with $0 \leq \deg(f_2,x_k) \leq \deg(f_1,x_k)$. Let monic $g = \gcd(f_1,f_2)$ exist and h_1 and h_2 be the cofactors of f_1 and f_2, respectively. Define $R_i = \mathrm{res}(h_1,h_2,x_i)$ and $D_r = \max_{i=1}^{k-1}(\deg(R_i,x_k))$. Let $\beta_k \in [0,p)$. Then* $\mathrm{Prob}[x_k = \beta_k$ *is an unlucky evaluation point*$] \leq \frac{(k-1)D_r}{p-\deg(f_2,x_k)}$.

Proof. Since, in general, $\bar{L}_p$ is not a field, the number of roots of R_i may exceed $\deg(R_i,x_k)$. To avoid this, assume $R_i \in \mathbb{Z}_p[z][x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_{k-1}][x_k]$. By Theorem 16, if $x_k = \beta_k$ is an unlucky evaluation point, then $\prod_{i=1}^{k-1} R_i(x_k = \beta_k) = 0$ so $\mathrm{Prob}[x_k = \beta_k$ is an unlucky evaluation point$] \leq \sum_{i=1}^{k-1} \mathrm{Prob}[R_i(x_k = \beta_k) = 0]$. For $1 \leq i \leq k$, we have $\mathrm{Prob}[R_i(x_k = \beta_k) = 0] \leq \frac{\deg(R_i,x_k)}{p-\deg(f_2,x_k)} \leq \frac{D_r}{p-\deg(f_2,x_k)}$. Summing over all $k-1$ variables yields the result.

To summarize, let $\#p$ be the number of primes used in MGCD, and $\#\beta$ be the number of evaluation points used in PGCD. Then,

$$\mathrm{Prob}[\text{MGCD Fails}] \leq \#p(\#\beta(\underbrace{\frac{\lfloor h + D_l(k-1)b + \log_2 T \rfloor + nm}{b \mid \mathbb{P}_b \mid}}_{\text{Prob}[(p,\beta)\text{ is lc-bad}](\textit{Theorem } 2)} \tag{10}$$

$$+ \underbrace{3k(\frac{d(n+m)d_x}{p} + \frac{\lfloor (d(n+m+1))/2\log(dB_R^2 + d(n+m)B_M^2)\rfloor}{b \mid \mathbb{P}_b \mid})}_{\text{Prob}[\text{PGCD encounters a zero-divisor}](\textit{Equation } 8)} \tag{11}$$

$$+ \underbrace{\frac{(k-1)D_r}{p-\deg(f_2,x_k)}}_{\text{Prob}[x_k = \beta_k\text{ is an unlucky evaluation point}](\textit{Theorem } 17)}) \tag{12}$$

$$+ \underbrace{\frac{\lfloor d/2\log_2 d + d(C + \sum_{i=1}^{n}\delta_i \log_2(l_{n-i+1} + D_i\|\check{M}_{n-i+1}\|_\infty))\rfloor}{b \mid \mathbb{P}_b \mid}}_{\text{Prob}[p\text{ is det-bad}](\textit{Theorem } 8)} \tag{13}$$

$$+ \underbrace{\frac{k\lfloor \log_2 (t+s)! + (t+s-1)\log_2 T_M + (t+s)\log_2 H\rfloor}{b \mid \mathbb{P}_b \mid})}_{\text{Prob}[p\text{ is an unlucky prime}](\textit{Theorem } 15)} \tag{14}$$

4 Conclusion

We have analyzed all failure cases of the MGCD and PGCD algorithms of Ansari and Monagan from [2] and have determined that the numerators in (10), (11), (12), (13) and (14) of the failure probabilities are all polynomial in the sizes of the input and output, namely, d the degree of $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$, the size of the integer coefficients in $\check{m}_1, \check{m}_2, \ldots, \check{m}_n$, $\check{f}_1, \check{f}_2, g$, the degrees of f_1, f_2 in $x_1, x_2, \ldots, x_k$, the number of evaluation points used, the number of primes used, and the number of terms of f_1, f_2, g.

References

1. Saban Alaca and Kenneth S. Williams. Introductory algebraic number theory. 2003.
2. Mahsa Ansari and Michael Monagan. Computing GCDs of multivariate polynomials over algebraic number fields presented with multiple extensions. In *Computer Algebra in Scientific Computing*, LNCS 14139, page 1–20. Springer, 2023.
3. Mahsa Ansari and Michael Monagan. A modular algorithm to compute the resultant of multivariate polynomials over algebraic number fields presented with multiple extensions. In *Computer Algebra in Scientific Computing*, volume 14938, pages 27–46. Springer, 2024.
4. W. S. Brown. On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *J. ACM*, 18:478–504, 1971.
5. B. Buchberger, George E. Collins, Rudiger Loos, and Rudolf Albrecht, editors. *Computer algebra: symbolic and algebraic computation (2nd ed.)*. Springer-Verlag, Berlin, Heidelberg, 1983.
6. George E. Collins. The calculation of multivariate polynomial resultants. *J. ACM*, 18(4):515–532, oct 1971.
7. D. Cox, J. Little, and D. OSHEA. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2013.
8. David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag, 2007.
9. Mark J. Encarnación. Computing gcds of polynomials over algebraic number fields. *J. Symb. Comput.*, 20:299–313, 1995.
10. K.O. Geddes, S.R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Springer, 1992.
11. Lars Langemyr and Scott McCallum. The Computation of Polynomial Greatest Common Divisors over an Algebraic Number Field. *J. Symb. Comput.*, 8(5):429–448, 1989.
12. Michael Monagan. Maximal Quotient Rational Reconstruction: An Almost Optimal Algorithm for Rational Reconstruction. In *Proceedings of ISSAC 2004*, pages 243–249. ACM, 2004.
13. Michael Monagan. Maximal Quotient Rational Reconstruction: An Almost Optimal Algorithm for Rational Reconstruction. In *Proceedings of ISSAC 2004*, pages 243–249. ACM, 2004.
14. Mark van Hoeij and Michael Monagan. A Modular GCD Algorithm over Number Fields Presented with Multiple Extensions. In *Proceedings of ISSAC 2002*, page 109–116. ACM, 2002.

5 Appendix A (Algorithms MGCD and PGCD)

Algorithm 5: MGCD

Input: $f_1, f_2 \in L[x_1, \ldots, x_k]$ where $L = \mathbb{Q}[z_1, \ldots, z_n]/\langle M_1(z_1), \ldots, M_n(z_n)\rangle$
Output: $\gcd(f_1, f_2)$

1 $M := 1$
2 $f_1 := \check{f}_1$ and $f_2 := \check{f}_2$ `// Clear fractions`
3 **while** *true* **do**
4   Choose a new random prime p that is not lc-bad.
5   Choose $C_1, \ldots, C_{n-1} \in [1, p)$ at random and set $\gamma = z_1 + \sum_{i=2}^{n} C_{i-1} z_i$
6   Call Algorithm 2 with inputs $[\phi_p(\check{M}_1), \ldots, \phi_p(\check{M}_n)]$, $\mathbb{Z}_p$ and $\phi_p(\gamma)$ to compute $M(z)$, A, and A^{-1}
7   **if** *Algorithm 2 fails* **then**
8   Go back to step 4
    `// Apply Algorithm 6 to get the monic gcd over` $\bar{L}_p$
9     $G_p = PGCD(\phi_\gamma(\phi_p(f_1)), \phi_\gamma(\phi_p(f_2))) \in \bar{L}_p[x_1, \ldots, x_k]$
10   **if** $G_p = FAIL$ **then**
        `// PGCD has encountered a zero-divisor.`
11   Go back to step 4.
12   **if** $\deg(G_p) = 0$ **then**
13   **return**(1)
    `// Convert` $G_p \in \bar{L}_p$ `to its corresponding polynomial over` L_p
14     $G_p := \phi_\gamma^{-1}(G_p)$
15     $lm := \mathrm{lm}(G_p)$ w.r.t lexicographic order with $x_1 > x_2 \ldots > x_k$
16   **if** $M = 1$ *or* $lm < least$ `// First iteration or all the previous primes were unlucky.`
17   **then**
18         $G, least, M := G_p, lm, p$
19   **else**
20   **if** $lm = least$ **then**
21   Using CRT, compute $G' \equiv G \mod M$ and $G' \equiv G_p \mod p$
22   set $G = G'$ and $M = M \cdot p$
23   **else if** $lm > least$ **then**
            `//` p `is an unlucky prime`
24   Go back to step 4
25     $H :=$ Rational Number Reconstruction of G mod M
26   **if** $H \neq FAIL$ **then**
27   Choose a new prime q and $b_2, \ldots, b_n \in \mathbb{Z}_q$ at random such that $\mathrm{lc}(H)(x_1, b_2, \ldots, b_k) \neq 0$
28         $A,\ B,\ C := f_1(x_1, b_2, \ldots, b_k), f_2(x_1, b_2, \ldots, b_k), H(x_1, b_2, \ldots, b_k)$
        `//` A, B, C `are polynomials in` $L_q[x_1]$
29   **if** $C \mid A$ *and* $C \mid B$ **then**
30   **return**(H)

Algorithm 6: PGCD

Input: $f_1, f_2 \in \bar{L}_p[x_1, \ldots, x_k]$
Output: $\gcd(f_1, f_2) \in \bar{L}_p[x_1, \ldots, x_k]$ or FAIL

1 $Xk := [x_1, \ldots, x_{k-1}]$
2 $prod := 1$
3 **if** $k = 1$ **then**
4 $\quad H := \gcd(f_1, f_2) \in \bar{L}_p[x_1]$
5 $\quad$ **return**(H)
6 $c_1 := \mathrm{cont}(f_1, Xk) \in \bar{L}_p[x_k]$. **if** $c_1 = FAIL$ **then return**(FAIL)
7 $c_2 := \mathrm{cont}(f_2, Xk) \in \bar{L}_p[x_k]$. **if** $c_2 = FAIL$ **then return**(FAIL)
8 $c := \gcd(c_1, c_2) \in \bar{L}_p[x_k]$. **if** $c = FAIL$ **return**(FAIL)
9 $f_1, f_2 := f_1/c_1, f_2/c_2$
10 $\Gamma := \gcd(lc(f_1, Xk), lc(f_2, Xk)) \in \bar{L}_p[x_k]$. **if** $\Gamma = FAIL$ **return**(FAIL)
11 **while** *true* **do**
12 $\quad$ Take a new random evaluation point, $j \in \mathbb{Z}_p$, which is not lc-bad.
13 $\quad F_{1_j} := f_1(x_1, \ldots, x_{k-1}, x_k = j)$ and $F_{2_j} := f_2(x_1, \ldots, x_{k-1}, x_k = j)$
14 $\quad G_j := PGCD(F_{1_j}, F_{2_j}, p) \in \bar{L}_p[x_1, \ldots, x_{k-1}]$
$\quad$ // $\mathrm{lc}(G_j) = 1$ in lex order with $x_1 > x_2 > \ldots > x_{k-1}$
15 $\quad$ **if** $G_j = FAIL$ **then**
16 $\quad\quad$ return(FAIL)
17 $\quad lm := \mathrm{lm}(G_j, Xk)$ // in lex order with $x_1 > x_2 > \ldots > x_{k-1}$
18 $\quad \Gamma_j := \Gamma(j) \in \mathbb{Z}_p$
19 $\quad g_j := \Gamma_j \cdot G_j$ // Solve the leading coefficient problem
20 $\quad$ **if** $prod = 1$ *or* $lm < least$ **then**
$\quad\quad$ // First iteration or all previous evaluation points were unlucky.
21 $\quad\quad least,\ H,\ prod := lm,\ g_j,\ x_k - j$
22 $\quad$ **else**
23 $\quad\quad$ **if** $lm > least$ **then**
$\quad\quad\quad$ // j is an unlucky evaluation point
24 $\quad\quad\quad$ Go back to step 12.
25 $\quad\quad$ **else if** $lm = least$ **then**
$\quad\quad\quad$ // Interpolate x_k in the gcd H incrementally
26 $\quad\quad\quad V_j := prod(x_k = j)^{-1} \cdot (g_j - H(x_k = j))$
27 $\quad\quad\quad H := H + V_j \cdot prod$
28 $\quad\quad\quad prod := prod \cdot (x_k - j)$
29 $\quad$ **if** $\deg(prod, x_k) > \deg(H, x_k) + 1$ **then**
$\quad\quad$ // Make H primitive in $\bar{L}_p[x_k][x_1, \ldots, x_{k-1}]$.
30 $\quad\quad c_3 = \mathrm{cont}(H, Xk)$. **if** $c_3 = FAIL$ **then return**(FAIL) **else** $H := H/c_3$.
$\quad\quad$ // Test if H is the gcd of f_1 and f_2.
31 $\quad\quad$ Choose $b_2, \ldots, b_k \in \mathbb{Z}_p$ at random such that $\mathrm{lc}(H)(x_1, b_2, \ldots, b_k) \neq 0$
32 $\quad\quad A,\ B,\ C := f_1(x_1, b_2, \ldots, b_k), f_2(x_1, b_2, \ldots, b_k), H(x_1, b_2, \ldots, b_k)$
33 $\quad\quad$ **if** $C \mid A$ *and* $C \mid B$ **then**
34 $\quad\quad\quad$ **return**$(c \cdot H)$