Computing GCDs of polynomials modulo triangular sets

John Kluesner and Michael Monagan

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

Abstract

We present a modular algorithm for computing GCDs of univariate polynomials with coefficients modulo a zero-dimensional triangular set. Our algorithm generalizes previous work for computing GCDs over algebraic number fields. The main difficulty is when a zero divisor is encountered modulo a prime number. We give two ways of handling this: Hensel lifting, and fault tolerant rational reconstruction. We compare the two methods with illustrative examples. Both approaches have been implemented in Maple using the RECDEN package.

1 Introduction

Suppose that we seek to find the greatest common divisor of two polynomials $a, b \in \mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x]$ where α_i are algebraic numbers. This problem was first solved using a modular algorithm by Langemyr and McCallum [11] and improved by Encarnacion [6]. Their solution first found a primitive element and then applied an algorithm for one extension. Monagan and van Hoeij [9] improved the multiple extension case by circumventing the primitive element.

The computational model used for an algebraic number field is $\mathbb{Q}[z_1, \ldots, z_n]/T$ where $T = \langle t_1(z_1), t_2(z_1, z_2), \ldots, t_n(z_1, \ldots, z_n)\rangle$ and each t_i is the minimal polynomial of α_i over $\mathbb{Q}(\alpha_1, \ldots, \alpha_{i-1})$. A standard result in a course on rings and fields states that t_i is irreducible in its associated field. A natural generalization is to consider the same problem when each t_i is possibly reducible. Let $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ and $a, b \in R[x]$. This paper considers when $\gcd(a, b)$ exists and how to compute it. This problem also has applications to solving systems of polynomial equations and also to computing with solutions sets of polynomial systems.

The generators of T form what is known as a triangular set. Moreno Maza has done extensive research on the gcd problem and created an algorithm using subresultants [12]. Hubert also has considered this problem in a tutorial article [10]. Her article is recommended as an introduction to the theory of triangular sets. See also Aubry, Lazard and Moreno-Maza [2].

Our approach is best viewed as a generalization of Monagan and van Hoeij's algorithm. It is a modular algorithm that finds gcds modulo multiple primes, combines them using Chinese remaindering and uses rational number reconstruction (see [14, 13, 7]) to recover any fractions in g. This may make the algorithm sound simple, but this is far from true. We'd like to share some examples that illustrate some of the difficulties.

Example 1. Suppose we are working in $R[x]$ where $R = \mathbb{Q}[z_1, z_2]/T$ and $T = \langle z_1^2 + 1, z_2^2 + 1\rangle$. Note that $z_1 - z_2$ is a zero divisor with cofactor $z_2 + z_1$ in R. Consider computing the gcd of

$$\begin{aligned} a &= x^5 + 18\,x^4 z_1 + (20\,z_1 - 1)\,x^3 + (-323\,z_1 - z_2 - 18)\,x^2 + (-321 - 324\,z_1 - z_2)\,x + 4, \\ b &= x^4 + (18\,z_1 - 1)\,x^3 + (z_1 - 18)\,x^2 + (-324\,z_1 - z_2)\,x + 4 \end{aligned}$$

using the Euclidean algorithm. The remainder of $a \div b$ is

$$r_1 = (z_1 + 18)x^3 - 325x$$

Since $z_1 + 18$ is a unit, a division can be performed; dividing b by r_1 gives the remainder

$$r_2 = (z_1 - z_2)x + 4.$$

At this point, the Euclidean algorithm would attempt to invert $z_1 - z_2$, but instead will determine that it's a zero divisor.

Working modulo a prime p, one would expect the Euclidean algorithm to terminate at the same step, finding the zero divisor $z_1 - z_2 \pmod{p}$. However, consider $p = 17$. Here, $r_1 = (z_1 + z_2)x^3 - 2x$ and it will instead terminate when it encounters the zero divisor $z_1 + z_2 \pmod{17}$. Because of this, attempting to combine zero divisors using the CRT will always fail if the modular algorithm happens to pick the prime 17 and encounter the unlucky zero divisor $z_1 + z_2$.

Example 2. In the last example, suppose the zero divisor $z_1 - z_2$ was found and lifted successfully. From here, the algorithm would like to split the computation into two triangular sets $T^{(1)} = \{z_1^2 - 1, z_2 - z_1\}$ and $T^{(2)} = \{z_1^2 - 1, z_2 + z_1\}$. This works if the new zero divisor found is monic. However, it's possible for a monic polynomial to factor as two polynomials with zero divisors as leading coefficients. If this occurs, it will limit our ability reduce polynomials modulo the triangular sets. For example, consider the triangular set $T = \{(z_1^2 + 2)(z_1^2 + 1), z_2^3 - z_2\}$. Observe that when working modulo $(z_1^2 + 2)(z_1^2 + 1)$,

$$z_2^3 - z_2 = \left((z_1^2 + 2)z_2^2 - 1\right)\left((z_1^2 + 1)z_2^3 + z_2\right).$$

Of course, a nicer factorization may exist, like $z_2^3 - z_2 = (z_2^2 - 1)z$. However, it's not clear how to obtain one factorization from the other. This greatly enhances the complexity of handling zero divisors. The above example also shows that the degree formula for the product of two polynomials doesn't hold in this setting.

Example 3. Another difficulty is that denominators in the factors of a polynomial $a(x) \in R[x]$ may not appear in the denominators of $a(x)$. Weinberger and Rothschild give the following example in [15]. Let $t_1(z_1) = z_1^6 + 3z_1^5 + 6z_1^4 + z_1^3 - 3z_1^2 + 12z_1 + 16$ which is irreducible over $\mathbb{Q}$. The polynomial

$$f = x + \tfrac{4}{3} - \tfrac{11}{12}z_1 + \tfrac{7}{12}z_1^2 - \tfrac{1}{6}z_1^3 - \tfrac{1}{12}z_1^4 - \tfrac{1}{12}z_1^5$$

is a factor of $a(x) = x^3 - 3$ in $R[x]$.

The denominator of any factor of $a(x)$ ($\mathrm{denom}(f) = 12$ in this example) must divide the defect d of the field R. It is known that the discriminant Δ of $t_1(z_1)$ is a multiple of d, usually, much larger than d. Thus we could try to recover Δf with Chinese remaindering then make this result monic. Although one could try to generalize the discriminant to the case $n > 1$, using rational number reconstruction circumvents this difficulty and also allows us to recover g without using a lot more primes than necessary.

In section 2, we prove that greatest common divisors exist if the triangular set is radical and zero-dimensional. We use this to state exactly what will be computed. We also include relevant results that will be useful later. In section 3, we consider the monic Euclidean algorithm over a ring, as seen in Monagan's and van Hoeij's paper [9]. This will be used when computing modulo a prime number p. We modify the algorithm to output either a gcd or a zero divisor, if encountered.

In section 4, we present our new modular algorithm. The main complication comes when attempting to invert a zero divisor modulo a prime p. In Monagan and van Hoeij's paper R was a field, so they simply disregarded p and chose a new prime; this can't be done modulo a triangular set. Examples will be given illustrating this. We consider two approaches to handle zero divisors: one based on Hensel lifting and one based on Fault Tolerant Rational Reconstruction by Abbott [1]. Finally, in section 5 we give some details for our Maple implementation of our algorithm and make some concluding remarks.

2 Triangular Sets

We begin with some notation. All computations will be done in the ring $k[z_1, \dots, z_n]$ endowed with the monomial ordering $z_i < z_{i+1}$ and k a field. Let $f \in k[z_1, \dots, z_n]$ be non-constant. The *main variable* $\mathrm{mvar}(f)$ of f is the largest variable with nonzero degree in f, and the *main degree* of f is $\mathrm{mdeg}(f) = \deg_{\mathrm{mvar}(f)}(f)$.

As noted in the introduction, triangular sets will be of key interest in this paper. Further, they are to be viewed as a generalization of an algebraic number field with multiple extensions. For this reason, we impose extra structure than is standard:

Definition 1. A *triangular set* T is a set of non-constant polynomials in $k[z_1, \dots, z_n]$ with distinct main variables. Further:

(i) $|T| = n$,
(ii) $T = \{t_1, \dots, t_n\}$ where $\mathrm{mvar}(t_i) = z_i$,
(iii) t_i is monic with respect to z_i, and
(iv) $\deg_{z_j}(t_i) < \mathrm{mdeg}(t_j)$ for $j < i$.

The degree of T is $\prod_{i=1}^{n} \mathrm{mdeg}(t_i)$. Also, $T = \emptyset$ is a triangular set.

Condition (i) states there are no unused variables. This is be equivalent to T being zero-dimensional. Condition (ii) gives a standard notation that will be used throughout this paper. Conditions (iii) and (iv) relates the definition to that of minimal polynomials. Condition (iv) is commonly referred to as a reduced triangular set as seen in [2]. The degree of T is akin to the degree of an extension.

Example 4. The polynomials $\{z_1^3 + 4z_1, z_2^2 + (z_1 + 1)z_2 + 4\}$ form a triangular set. However, $\{z_2^2 + (z_1 + 1)z_2 + 4\}$ wouldn't since there's no polynomial with z_1 as a main variable. Also, $\{t_1 = z_1^3 + 4z_1,\ t_2 = z_2^2 + z_1^4 z_2 + 3\}$ isn't because $\deg_{z_1}(t_2) = 4 > \mathrm{mdeg}(t_1)$.

Given a triangular set T, we define $T_i = \{t_1, \dots, t_i\}$ and $T_0 = \emptyset$. For example, let $T = \{z_1^3 + 1,\ z_2^3 + 2,\ z_3^3 + 3\}$. Then, $T_3 = T$, $T_2 = \{z_1^3 + 1, z_2^3 + 2\}$, $T_1 = \{z_1^3 + 1\}$.

Proposition 1. Let $T \subset k[z_1, \dots, z_n]$ be a triangular set. Then T forms a Groebner basis with respect to the ordering $z_1 < z_2 < \cdots < z_n$.

Proof. Let $t_i, t_j \in T$. Observe that $\mathrm{lt}(t_i) = z_i^{\mathrm{mdeg}(t_i)}$ and $\mathrm{lt}(t_j) = z_j^{\mathrm{mdeg}(t_j)}$. They are relatively prime, and so Proposition 4 of Sec 2.9 of Cox, Little, O'Shea [5] completes the proof. □

It follows that $k[z_1, \dots, z_i] \cap \langle T \rangle = \langle T_i \rangle$ when $\langle T_i \rangle$ is viewed as an ideal of $k[z_1, \dots, z_i]$; this is a standard result of elimination theory, see Cox, Little, O'Shea [5].

2.1 Useful Lemmas About Rings

The section will contain useful lemmas about commutative rings that will be used throughout this paper. Some are standard exercises in a class on commutative algebra, and the others are straightforward to verify, see [3].

Proposition 2. Suppose $\psi\colon R \to R_1 \times R_2$ is a ring isomorphism. Let π_1, π_2 be the canonical projections $R_1 \times R_2$ to R_1, R_2, respectively. Let $a, b \in R$ and $g_1 = \gcd(\pi_1\psi(a), \pi_1\psi(b))$, $g_2 = \gcd(\pi_2\psi(a), \pi_2\psi(b))$. Then, $g = \psi^{-1}(g_1, g_2)$ is a gcd of a and b.

Proof. This follows directly from the definition of a gcd. For a formal proof, see the appendix. □

Proposition 3. Suppose n is a positive integer and $\psi\colon R \to \prod_{j=1}^n R_j$ is a ring isomorphism. Let $\pi_i\colon \prod_{j=1}^n R_j \to R_i$ be the canonical projections. Let $a, b \in R$ and $g_i = \gcd(\pi_i\psi(a), \pi_i\psi(b))$. Then, $g = \psi^{-1}(g_1, g_2, \ldots, g_n)$ is a gcd of a and b.

Proof. Follows by induction using Proposition 2. □

Proposition 4. Let R be a finite dimensional F-algebra where F is a field. Then any nonzero element of R is either a unit or zero divisor.

Proof. Let $\omega_1, \ldots, \omega_n$ be a basis for R over F. Let u be a nonzero element of R. Then, there exists $a_{ij} \in F$ satisfying $u\omega_i = \sum_{j=1}^n a_{ij}\omega_j$. Let $1 = b_1\omega_1 + \cdots + b_n\omega_n$ with $b_i \in F$. Let $A = (a_{ij})$, $b = [b_1, \ldots, b_n]$, and $x = [x_1, \ldots, x_n]$ where x_i are variables. Suppose u isn't a unit. Then, $u(x_1\omega_1 + \cdots + x_n\omega_n) = 1$ has no solution. Equivalently, there's no solution to the linear system $Ax = b$. This means A is not invertible and so $\ker(A)$ is nontrivial; i.e., $a(x_1\omega_1 + \cdots + x_n\omega_n) = 0$ has a nontrivial solution. □

Lemma 1 (CRT for ideals)**.** Let R be a ring with ideals $I_1, \ldots, I_n$ satisfying $I_i + I_j = R$ for $i \neq j$. Then there is a canonical isomorphism $R/\prod_{i=1}^n I_i \cong \prod_{i=1}^n R/I_i$ where the Cartesian product of the rings R/I_i is viewed as a ring under componentwise addition and multiplication.

Lemma 2. Let R be a commutative ring with unity. Then, $\sqrt{R[x]} = \sqrt{R} \cdot R[x]$. Furthermore,

$$R[x]^* = R^* + \sqrt{R} \cdot \langle x \rangle$$

where the right hand side is to be interpreted as the set of all polynomials $\sum_{i=0}^d a_i x^i$ where $a_0 \in R^*$ and $a_i \in \mathrm{rad}(R)$ for $i > 0$.

2.2 Radical Triangular Sets

To start, we give a structure theorem for triangular sets. The given proof is more difficult than necessary. For example, one could prove this more generally by using the associated primes of T. But, it allows us to introduce common ideas used throughout the paper. Since it is quite long, see the appendix for the proof.

Proposition 5. Let $I = \langle f \rangle \subset k[x]$ be an ideal. Then, $k[x]/I$ is isomorphic to a direct product of fields if and only if f is square-free.

Proof. Use the CRT. □

Theorem 1. Let $T \subseteq k[z_1, \ldots, z_n]$ be a triangular set and $I = \langle T \rangle$. Then, $k[z_1, \ldots, z_n]/I$ is isomorphic to a direct product of fields if and only if I zero-dimensional and radical.

Example 5. This example illustrates a nonradical triangular set. Consider $T = \{z_1^2 - 1, z_2^2 + 2z_1z_2 + 1\}$. Observe that $z_1 + z_2 \notin T$, but

$$(z_2 + z_1)^2 = z_2^2 + 2z_1z_2 + z_1^2 = (z_2^2 + (2z_1)z_2 + 1) + (z_1^2 - 1) \in T.$$

This shows $z_1 + z_2$ is nilpotent modulo T and so T isn't radical.

The structure theorem above gives many powerful corollaries:

Corollary 1. Let $T \subset k[z_1, \ldots, z_n]$ be a radical, zero-dimensional triangular set and $R = k[z_1, \ldots, z_n]/T$. Let $a, b \in R[x]$. Then a greatest common divisor of a and b exists.

Proof. Use Theorem 1 and Proposition 3. □

Corollary 2 (Extended Euclidean Representation)**.** Let $T \subset k[z_1, \ldots, z_n]$ be a radical, zero-dimensional triangular set and $R = k[z_1, \ldots, z_n]/T$. Let $a, b \in R[x]$ with $g = \gcd(a, b)$. Then, there exists polynomial $A, B \in R[x]$ such that $aA + bB = g$.

Proof. Note that $R \cong \prod F_i$ where F_i is a field, and we can extend this to $R[x] \cong \prod F_i[x]$. Let $a \mapsto (a_i)_i$ and $b \mapsto (b_i)_i$. Define $h_i = \gcd(a_i, b_i)$ in $F_i[x]$. By the extended Euclidean algorithm, there exists $A_i, B_i \in F_i[x]$ such that $a_iA_i + b_iB_i = h_i$. Let $h \mapsto (h_i)_i$ and $A \mapsto (A_i)_i$ and $B \mapsto (B_i)_i$. Clearly, $aA + bB = h$ in $R[x]$. Since $h \mid g$, we can multiply through by the quotient to write g as a linear combination of a and b. □

It should be noted that Corollary 2 works even if $\mathrm{lc}(g)$ is a zero divisor. This shows it's more powerful than the extended Euclidean algorithm.

Corollary 3 (Division Algorithm)**.** Let $T \subset k[z_1, \ldots, z_n]$ be a radical, zero-dimensional triangular set and $R = k[z_1, \ldots, z_n]/T$. Let $a, b \in R[x]$ with $\deg(a) \geq \deg(b) \geq 1$. Suppose b isn't a zero divisor. Then there exists a quotient q and remainder r satisfying $a = bq + r$ and $\deg(r) < \deg(b)$. The remainder r is unique if and only if $\mathrm{lc}(b)$ is a unit.

Proof. Note that $R \cong \prod F_i$ is isomorphic to a product of fields, and we can extend this to $R[x] \cong \prod F_i[x]$. Let $a \mapsto (a_i)_i$ and $b \mapsto (b_i)_i$. Note that all $b_i \neq 0$ or else b would a zero divisor. Apply the division algorithm over a field to get q_i, $r_i \in F_i[x]$ with $a_i = q_ib_i + r_i$ and $r_i = 0$ or $\deg(r_i) < \deg(b_i)$. The element $r \mapsto (r_i)_i$ will have degree less than b since all of its images do. This shows existence. For uniqueness, first let $\mathrm{lc}(b)$ be a unit. Suppose $a = bq_1 + r_1$ and $a = bq_2 + r_2$. Then, $b(q_1 - q_2) = r_2 - r_1$. Since $\mathrm{lc}(b)$ is a unit, $\deg(b(q_1 - q_2)) \geq \deg(b)$ unless $q_1 - q_2 = 0$. Clearly, $\deg(r_2 - r_1) < \deg(b)$ which leaves $q_1 - q_2 = 0$ as the only possibility. It follows that $r_1 = r_2$. Conversely, suppose $\mathrm{lc}(b)$ isn't a unit, and hence a zero divisor by Proposition 4. Let $v \in R$ be such that with $\mathrm{lc}(b)v = 0$. Note that $\deg(bv) < \deg(b)$. Given a remainder r and quotient q, algebraic manipulation gives

$$a = bq + r = (q - v)b + bv + r.$$

Since $\deg(bv + r) < \deg(b)$, this gives two distinct remainders as long as $bv \neq 0$, which must be the case as b isn't a zero divisor. □

3 The Euclidean Algorithm over a Ring

3.1 The Monic Euclidean Algorithm

In this section, we will assume we're working over a commutative ring R with unity where every non-zero element of the ring is either a unit or a zero divisor, and that there's an algorithm that

```
Algorithm 1: MonicEuclideanAlgorithm
Input  : A ring $R$ as specified in the opening of the section, and two polynomials
         $a, b \in R[x]$. Assume $\deg_x(a) \geq \deg_x(b) \geq 0$.
Output: Either $\gcd(a, b)$ or an error if a zero divisor is encountered.
Set $r_0 := a$ and $r_1 := b$;
$i := 1$;
while $r_i \neq 0$ do
    if $\mathrm{lc}(r_i)$ is a zero divisor then return $ZERODIVISOR(\mathrm{lc}(r_i))$;
    $r_i := \mathrm{lc}(r_i)^{-1} r_i$;
    Set $r_{i+1}$ to be the remainder of $r_{i-1}$ divided by $r_i$;
    $i = i + 1$;
end
return $r_{i-1}$
```

can decide if an element is a unit, and we have a method to compute inverses. This sufficient to be able to use the monic Euclidean algorithm in $R[x]$.

Proposition 6. Let $a, b \in R[x]$ with $\deg(a) \geq \deg(b)$. Suppose no zero divisors are encountered when running the monic Euclidean algorithm on a and b. Then the output is a $\gcd(a, b)$.

Proof. Let $g = r_{i-1}$, the last nonzero remainder. We have to show that (i) $g \mid a$ and $g \mid b$, and (ii) any common divisor $d \mid a$ and $d \mid b$ also divides g. It will be useful to index the quotient q_j and leading coefficient $c_j = \mathrm{lc}(r_j)$ at the jth iteration; so $r_{j-2} = q_j r_{j-1} + c_j r_j$. With that in mind, for (i), note that $r_{i-2} = q_i r_{i-1}$, and $g \mid r_{i-2}$. This implies $g \mid q_{i-1} r_{i-2} + c_{i-1} r_{i-1} = r_{i-3}$ as well. We can repeat the above argument $i - 3$ more times to get $g \mid r_0$ and $g \mid r_1$. Clearly, $g \mid \mathrm{lc}(b) r_1 = b$ as well. This completes (i). For (ii), consider a common divisor d. Then, $d \mid r_0 = a$ and $d \mid \mathrm{lc}(b)^{-1} b = r_1$. This implies $d \mid r_0 - q_2 r_1 = r_2$. Apply this argument $i - 2$ more times to get $d \mid r_{i-1}$, as desired. □

One could use this algorithm for $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ when T is radical and zero-dimensional. However, this will lead to coefficient growth. Instead, we will use it over $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$ for a prime number p. The details of the implementation will be given later, but the general idea is that every non-zero element is a zero divisor or unit since R is a finite-dimensional $\mathbb{Z}_p$-algebra; and recursively running the extended Euclidean algorithm with parameters $\mathrm{lc}(r_i)$ and t_n can be used for computing $\mathrm{lc}(r_i)^{-1}$ and testing for invertibility.

4 The Modular Algorithm

The main content of this section is to fully present and show the correctness of our modular algorithm. First, let's suppose a zero divisor w over $\mathbb{Q}$ is found while running the algorithm. It will be used to factor $t_k = uv \mod \langle T_{k-1} \rangle$ where u and v are monic with main variable z_k. From here, the algorithm proceeds to split T into $T^{(u)}$ and $T^{(v)}$ where t_k is replaced with u in $T^{(u)}$ and t_k is replaced with v in $T^{(v)}$. Of course t_i is reduced for $i > k$ as well. The algorithm then continues recursively. Once the recursive calls are finished, we could use the CRT to combine gcds into a single gcd, but this would be very time consuming. Instead, it's better to just return both gcds along with the associated triangular sets. This approach is similar to Hubert's in [10] which she calls a pseudo-gcd. Here, we refer to this as a component-wise gcd, or c-gcd for short:

Definition 2. Let R be a commutative ring with unity such that $R \cong \prod_{i=1}^{r} R_i$ and $a, b \in R[x]$. Let $\pi_i \colon R \to R_i$ be the natural projections. A component-wise gcd of a and b is a tuple $(g_1, \ldots, g_r) \in \prod_{i=1}^{r} R_i[x]$ where each $g_i = \gcd(\pi_i(a), \pi_i(b))$ and $\mathrm{lc}(g_i)$ is a unit.

The modular algorithm's goal will be to compute c-gcd(a, b) given $a, b \in R[x]$ where $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ and $T \subset \mathbb{Q}[z_1, \ldots, z_n]$ is a radical triangular set.

Section 4.1 will describe the primes the modular algorithm chooses. We have to make sure for each prime chosen, the triangular set T remains radical modulo p. We will prove all but finitely many prime numbers enjoy this property. Further, we will give an algorithm that can determine if T remains radical modulo a prime p. Also, we need the the modular image of the gcd to be the gcd of the modular images; i.e., we will need the chosen prime p to not be unlucky, see section 4.1.2. We will prove finiteness in this case as well. In section 4.2, we give an overview of the modular algorithm. It is based on Brown's classical modular algorithm from [4] (see also [7, 8]), but care has to be taken when zero divisors are encountered. In section 4.3, we discuss using Hensel lifting for handling zero divisors. We give details of the algorithm as well as proofs. In section 4.4, we consider Fault Tolerant Rational Reconstruction (FTRR) for handling zero divisors. This method is based on a new algorithm by Abbott in [1]. We refer to the original article for the details of FTRR. Proofs will be provided showing its correctness in our application.

4.1 Primes

4.1.1 Radical Primes

Consider a radical triangular set $T \subset \mathbb{Q}[z_1, \ldots, z_n]$. For many reasons, the modular algorithm will cease to work if T doesn't remain radical modulo a prime p. This leads to the definition of a radical prime:

Definition 3. Let $T \subset \mathbb{Q}[z_1, \ldots, z_n]$ be a radical triangular set. A prime number p is a radical prime if p doesn't appear as a denominator of any of the polynomials in T, and if $T \mod p \subset \mathbb{Z}_p[z_1, \ldots, z_n]$ remains radical.

Example 6. The triangular set $\{z_1^2 - 3\}$ is radical over $\mathbb{Q}$. Since the discriminant of $z_1^2 - 3$ is 12, it follows that $2, 3$ aren't radical primes, but all other primes are.

If there were an infinite family of nonradical primes, it would present a problem for the algorithm. The following proofs show this won't happen.

Lemma 3. *Let $T \subset k[z_1, \ldots, z_n]$ be a zero-dimensional triangular set over a perfect field k. Then T is radical if and only if $\gcd(t_i, t_i') = 1 \pmod{T_{i-1}}$ for all i.*

Proof. ($\Longrightarrow$) Suppose T_{j-1} is radical. Then, $k[z_1, \ldots, z_j]/T_{j-1} \cong \prod F_i[z_j]$ for some fields F_i. Let $t_j \mapsto (t_{ji})$ and $g_i = \gcd(t_{ji}, t_{ji}')$ over F_i. We claim all g_i are units. If any g_i weren't a unit, then $F_i[z_j]/t_{ji}$ would contain a nilpotent element because t_{ji} wouldn't be square-free. However, T_j is radical and so $k[z_1, \ldots, z_j]/T_j \cong \prod F_i[z_j]/t_{ji}$ contains no nilpotents. Well, $g \mapsto (g_i)$ would then also be a unit as well as a gcd of t_j and t_j' modulo T_{j-1}. Since gcds divide each other, $\gcd(t_j, t_j') = 1 \pmod{T_{j-1}}$.

($\Longleftarrow$) Proceed by induction on n. If $n = 1$ we have $\gcd(t_1, t_1') = 1$ in $\mathbb{Q}[z_1]$ which implies t_1 is square-free hence $T_1 = \{t_1\}$ is radical. The induction hypothesis asserts T_{n-1} is radical, and so $\mathbb{Q}[z_1, \ldots, z_{n-1}/T_{n-1} \cong \prod F_i$ where F_i are finite extensions of k, and hence perfect as well. Let $t_n \mapsto (t_{ni})$ under this isomorphism. Let $g_i = \gcd(t_{ni}, t_{ni}')$ over F_i and $g \mapsto (g_i)$. Then, g would be a common divisor of $\gcd(t_n, t_n') = 1$ and so g is a unit. Since homomorphisms preserve units, g_i is a unit and we may assume $g_i = 1$. It follows that t_{ni} is square-free because F_i is a perfect field.

Then, $F_i[z_n]/t_{ni}$ contains no nilpotent elements, and so $k[z_1,\dots,z_n]/T \cong \prod F_i[z_n]/t_{ni}$ contains no nilpotents as well. □

Theorem 2. Let $T \subset \mathbb{Q}[z_1,\dots,z_n]$ be a radical, zero-dimensional triangular set. All but finitely many primes are radical primes.

Proof. By Lemma 3, $\gcd(t_i, t_i') = 1$. By the extended Euclidean representation (Corollary 2), there exist polynomials $A_i, B_i \in (\mathbb{Q}[z_1,\dots,z_{i-1}]/T_{i-1})[z_i]$ where $A_i t_i + B_i t_i' = 1 \mod \langle T_{i-1}\rangle$. Take any prime p that doesn't divide the denominator of any A_i, B_i, t_i, t_i'. This means one can reduce this equation modulo p and so $A_i t_i + B_i t_i' \mod \langle T_{i-1}, p\rangle$. This implies $\gcd(t_i, t_i') = 1 \mod \langle T_{i-1}, p\rangle$ and so T remains radical modulo p by Lemma 3. There are only a finite amount of primes that divide the denominator of any of these polynomials. □

Lemma 3 also gives the main idea of an algorithm to test if a prime is radical:

```
Algorithm 2: isRadical
Input  : A zero-dimensional, radical triangular set $T \subset \mathbb{Q}[z_1,\dots,z_n]$ and a prime number $p$
         where $p \nmid \mathrm{den}(T)$.
Output: A boolean indicating if $T$ remains radical modulo $p$, or a zero divisor.
for $i = 1,\dots,n$ do
    $dt := \frac{\partial}{\partial z_i} T[i]$;
    $g := \gcd(T[i], dt)$ over $\mathbb{Z}_p[z_1,\dots,z_i]/T_{i-1}$;
    if $g =$ ZERODIVISOR($u$) then return ZERODIVISOR($u$);
    if $g \neq 1$ then return False;
end
return True;
```

4.1.2 Unlucky Primes

As with all modular algorithms, it's possible that some primes are unlucky. We also prove this only happens for a finite amount of cases.

Definition 4. Let $T \subset \mathbb{Q}[z_1,\dots,z_n]$ be a radical triangular set, and $R = \mathbb{Q}[z_1,\dots,z_n]/T$. Let $a, b \in R[x]$ and $g = \text{c-gcd}(a,b)$. A prime number p is an *unlucky prime* if g doesn't remain a componentwise greatest common divisor of a and b modulo p. Additionally, a prime is *bad* if the it divides any denominator in T, any denominator in a or b, or if $\mathrm{lc}(a)$ or $\mathrm{lc}(b)$ vanishes modulo p.

Theorem 3. Let $T \subset \mathbb{Q}[z_1,\dots,z_n]$ be a radical triangular set, and $R = \mathbb{Q}[z_1,\dots,z_n]/T$. Let $a, b \in R[x]$ and $g = \text{c-gcd}(a,b)$. Only finitely many primes are unlucky.

Proof. Let $R[x] \cong \prod R_i[x]$ where $(g_i) = \text{c-gcd}(a,b) \in R_i[x]$. Let $a \mapsto (a_i)$ and $b \mapsto (b_i)$. If $g_i = 0$, then $a_i = 0$ and $b_i = 0$ and no primes are unlucky since, $\gcd(0,0) \equiv 0 \pmod p$. Suppose $g_i = \gcd(a_i, b_i)$ is nonzero and monic. Let $\overline{a}_i$ and $\overline{b}_i$ be the cofactors $a_i = g_i\overline{a}_i$ and $b_i = g_i\overline{b}_i$. I claim $\gcd(\overline{a}_i, \overline{b}_i) = 1$. To show this, consider a common divisor f of $\overline{a}_i$ and $\overline{b}_i$. Note that $fg_i \mid a_i$ and $fg_i \mid b_i$. Since $g_i = \gcd(a_i, b_i)$, it follows that $fg_i \mid g_i$; so there exists $q \in R_i[x]$ where $fg_iq = g_i$. Rewrite this equation as $(fq-1)g_i = 0$. Well, g_i is monic in x, and so can't be a zero divisor. This implies $fq - 1 = 0$ and so indeed f is a unit. Thus, $\gcd(\overline{a}_i, \overline{b}_i) = 1$. By the extended Euclidean representation (Corollary 2), there exists $A_i, B_i \in R_i[x]$ where $\overline{a}_iA_i + \overline{b}_iB_i = 1$.

Let p be a prime where p doesn't divide any of the denominators in $a_i, \overline{a}_i, A_i, b_i, \overline{b}_i, B_i, g_i$. Then, we can reduce the equations

$$\overline{a}_i A_i + \overline{b}_i B_i = 1 \pmod p \tag{1}$$

$$a_i = g_i \overline{a}_i \pmod p, \qquad b_i = g_i \overline{b}_i \pmod p. \tag{2}$$

We will now show that $g_i = \gcd(a_i, b_i) \pmod p$. By (2), we get g_i is a common divisor of a_i and b_i. Consider a common divisor c of a_i and b_i. Multiplying equation (1) through by g_i gives $a_i A_i + b_i B_i = g_i$. Clearly, $c \mid g_i$. Thus, g_i is indeed a greatest common divisor of a_i and b_i. As there are finitely many primes that can divide the denominators of fractions in the polynomials $a_i, \overline{a}_i, A_i, b_i, \overline{b}_i, B_i, g_i$, there are indeed finitely many unlucky primes. □

4.2 The Modular Algorithm

This main content of this section is Algorithm ModularC-GCD. It has Monagan and van Hoeij's algorithm as its backbone but handles zero divisors differently. This is because we have to account for the case where the Euclidean algorithm over $\mathbb{Q}$ encounters a zero divisor. We will give two ways to handle this zero divisor problem later, for now we leave a blackbox algorithm HandleZeroDivisor(u) that gives ModularC-GCD instructions on how to proceed.

Example 7. This example illustrates how the IsRadical function can run into a zero divisor. Consider $T = \{z_1^2 - 1, z_2^2 + 2(z_1 - 1)z_2 + 1\}$. We will be running the algorithm over $\mathbb{Q}$. First, it would determine that $T_1 = \{z_1^2 - 1\}$ is radical. Now, when it is running the Euclidean algorithm on $t_2 = z_2^2 + 2(z_1 - 1)z_2 + 1$ and $t_2' = 2z_2 + 2(z_1 - 1)$, the first remainder would be $(z_1 - 1)z_2 + 1$. However, $z_1 - 1$ is a zero divisor, so the algorithm would output ZERODIVISOR($z_1 - 1$). This same zero divisor will show up for every prime besides 2. This explains why we can't just simply pick a new prime in Algorithm ModularC-GCD if IsRadical encounters a zero divisor.

The crux of ModularC-GCD is an algorithm to compute $\gcd(a, b)$ for two polynomials $a, b \in (\mathbb{Z}_p[z_1, \ldots, z_n]/T)[x]$. The algorithm we'll be using for this is EuclideanC-GCD. It is a variant of the monic Euclidean algorithm. For computing inverses, the Extended Euclidean algorithm (EEA) can be used; modifying EuclideanC-GCD to do this is straightforward.

A short discussion about the zero divisors that may appear is warranted. To compute an inverse, the modular algorithm will be using the Extended Euclidean algorithm. The first step would be to invert a leading coefficient u of some polynomial. This requires a recursive call to ExtendedEuclideanC-GCD$(u, t_k) \mod \langle T_{k-1} \rangle$ where $z_k = \text{mvar}(u)$. If u isn't monic, then it would again attempt to invert $\text{lc}(u)$. Because of the recursive nature, it will keep inverting leading coefficients until it succeeds or a monic zero divisor is found. The main point is that we may assume that the zero divisors encountered are monic.

4.3 Zero-Divisors: Hensel Lifting

The main content of this section will be to show how a variant of Hensel lifting can be used for handling the zero divisor problem. First, section 4.3.1 will show that Hensel lifting can be done in the given ring. Next, section 4.3.2 will give the idea for using it to solve the zero divisor problem and a proof of correctness.

4.3.1 Hensel Lifting

A general factorization $ab = f \pmod p$ for $a, b, f \in R[x]$ will not be liftable. Certain conditions are needed for both existence and uniqueness of each lifting step. For one, we will need $\gcd(a, b) = 1$

Algorithm 3: ModularC-GCD

```
Input  : A zero-dimensional, radical triangular set $T \subset \mathbb{Q}[z_1, \ldots, z_n]$ and two polynomials
         $a, b \in R[x]$ where $R = \mathbb{Q}[z_1, \ldots, z_n]/T$. Assume $\deg(a) \geq \deg(b) \geq 0$.
Output: A tuple consisting of comaximal triangular sets $T^{(i)}$ such that $T = \bigcap T^{(i)}$ and
         $g_i = \gcd(a, b) \mod \langle T^{(i)} \rangle$ where $g_i = 0$ or $\mathrm{lc}(g_i)$ is a unit.
Initialize $dg := \deg(b)$, $P = 1$;
Main Loop: Pick a prime $p$ that isn't bad (see Definition 4);
Test if $p$ is a radical prime: $B :=$ isRadical$(T, p)$;
if $B =$ZERODIVISOR$(u)$ then
    Call HandleZeroDivisor$(u)$ and react accordingly;
else if $B =$False then
    Pick a new prime: Go to Main Loop;
end
Set $g := \gcd(a, b) \mod \langle T, p \rangle$ using algorithm MonicEuclideanAlgorithm;
if $g =$ZERODIVISOR$(u)$ then
    Call HandleZeroDivisor$(u)$ and react accordingly;
else
    if $\deg(g) = dg$ then
        The chosen prime seems to be lucky;
        Use CRT to combine $g$ other gcds (if any), store the result in $G$ and set $P := P \times p$;
    else if $\deg(g) > dg$ then
        The chosen prime was unlucky, discard $g$;
        Pick a new prime: Go to Main Loop;
    else if $\deg(g) < dg$ then
        All previous primes were unlucky, discard $G$;
        Set $G := g$ and $P := p$;
    end
    Set $h :=$RationalReconstruction$(G \pmod{P})$;
    if $h \neq$ FAIL and $h \mid a$ and $h \mid b$ then return $(T, h)$;
    Pick a new prime: Go to Main Loop;
end
```

Algorithm 4: EuclideanC-GCD

```
Input  : A zero-dimensional, radical triangular set $T \subset k[z_1, \ldots, z_n]$ and two polynomials
         $a, b \in R[x]$ where $R = k[z_1, \ldots, z_n]/T$. Assume $\deg(a) \geq \deg(b) \geq 0$.
Output: Either $\gcd(a, b) \pmod{T}$ or a zero divisor.
Initialize $r_0 := a$, $r_1 := b$ and $i := 1$;
while $r_i \neq 0$ do
    Compute $s := \mathrm{lc}(r_i)^{-1} \mod \langle T_{k-1} \rangle$ using the EEA;
    if $s =$ ZERODIVISOR$(u)$ then return ZERODIVISOR$(u)$ else $r_i := s \times r_i$;
    Let $r_{i+1}$ be the remainder of $r_{i-1}$ divided by $r_i$;
    $i = i + 1$;
end
return $r_{i-1}$
```

(mod p) as is required in the case with no extensions to satisfy existence. Further, we will need both a and b to be monic to satisfy uniqueness. The following lemma gives a uniqueness criteria to the extended Euclidean representation. It is a generalization of theorem 26 in Geddes, Czapor, Labahn [8] from $F[x]$ to $R[x]$.

Lemma 4. Let $T \subset k[z_1, \dots, z_n]$ be a radical, zero-dimensional triangular set and $R = k[z_1, \dots, z_n]/T$. Let $a, b \in R[x]$ be nonzero and monic with $1 = \gcd(a, b)$. Then, for any polynomial $c \in R[x]$, there exist unique polynomials $\sigma, \tau \in R[x]$ such that

$$a\sigma + b\tau = c, \quad \deg(\sigma) < \deg(b).$$

Proof. Existence: By Corollary 2, there exist polynomials A, B satisfying $aA+bB = 1$. Multiplying through by c gives $a(cA) + b(cB) = c$. Dividing cA by b, which we can do since b is monic, gives $cA = qb + r$ with $r = 0$ or $\deg(r) < \deg(b)$. Define $\sigma = r$ and $\tau = cB + qa$. Observe that

$$a\sigma + b\tau = ar + b(cB + qa) = ar + bcB + abq = a(r + bq) + bcB = acA + bcB = c(aA + bB) = c$$

thus σ and τ satisfy the conditions of the Lemma. *Uniqueness*: Suppose both pairs σ_1, τ_1 and σ_2, τ_2 satisfy $a\sigma_i + b\tau_i = c$ with the desired degree constraint. This yields

$$(\sigma_1 - \sigma_2)a = b(\tau_2 - \tau_1).$$

Since $\gcd(a, b) = 1$, it follows that $b \mid \sigma_1 - \sigma_2$. However, since b is monic and $\deg(\sigma_1 - \sigma_2) < \deg(b)$, this is only possible if $\sigma_1 - \sigma_2 = 0$. Thus $0 = b(\tau_2 - \tau_1)$. Next, since b is not a zero divisor (because it's monic), this can only happen if $\tau_2 - \tau_1 = 0$ as well. □

We're also particularly interested in trying to factor t_n modulo T_{n-1}. That's because a zero divisor leads to such a factorization. That is, if w is a zero divisor with main variable z_n, we can write $u = \gcd(t_n, w)$ and then $t_n = uv \mod \langle T_{n-1} \rangle$ by the division algorithm. We know $u \neq 1$ since w is a zero divisor. As long as T is radical, the next lemma shows we automatically get $\gcd(u, v) = 1$.

Lemma 5. Let $T \subset k[z_1, \dots, z_n]$ be a radical, zero-dimensional triangular set. Suppose $uv \equiv t_n \pmod{T_{n-1}}$. Then, $1 = \gcd(u, v) \pmod{T_{n-1}}$.

Proof. Let $u = \overline{u}g \pmod{T_{n-1}}$ and $v = \overline{v}g \pmod{T_{n-1}}$. Note that $t_n \equiv \overline{u}\overline{v}g^2 \pmod{T_{n-1}}$. This would imply $(\overline{u}\overline{v}g)^2 \equiv 0 \pmod{T}$; that is, $\overline{u}\overline{v}g$ is a nilpotent element. However, since nilpotent elements don't exist modulo a radical ideal, $\overline{u}\overline{v}g \equiv 0 \pmod{T}$. This would imply $\overline{u}\overline{v}g \equiv qt_n \pmod{T_{n-1}}$ for some polynomial q. Then,

$$(gq - 1)t_n \equiv gqt_n - t_n \equiv g\overline{u}\overline{v}g - t_n \equiv 0 \pmod{T_{n-1}}.$$

Since t_n is monic in z_n, it can't be a zero divisor modulo T_{n-1}. Therefore, $gq - 1 \equiv 0 \pmod{T_{n-1}}$. Thus, g is a unit modulo T_{n-1} and so indeed $1 = \gcd(u, v) \pmod{T_{n-1}}$. □

Finally, the next proposition shows that lifting is possible. The proof given is simply the Hensel construction.

Proposition 7. Let $T \subset \mathbb{Z}_p[z_1, \dots, z_n]$ be a zero-dimensional, radical triangular set with p a prime number. Suppose $t_n \equiv u_0v_0 \pmod{T_{n-1}, p}$ where u_0 and v_0 are monic. Then, there exist unique monic polynomials u_k, v_k such that $t_n \equiv u_kv_k \mod \langle T_{n-1}, p^k \rangle$ and $u_k \equiv u_0 \mod \langle T_{n-1}, p \rangle$ and $v_k \equiv v_0 \mod \langle T_{n-1}, p \rangle$ for all $k \geq 1$.

Proof. Work by induction on k. The base case is clear. For the inductive step, we want to be able to write $u_k = u_{k-1} + p^{k-1}a \mod \langle T_{n-1}, p^k \rangle$ and $v_k = v_{k-1} + p^{k-1}b \mod \langle T_{n-1}, p^k \rangle$ satisfying

$$t_n \equiv u_k v_k \mod \langle T_{n-1}, p^k \rangle.$$

Multiplying out u_k, v_k gives

$$t_n \equiv u_k v_k \equiv u_{k-1}v_{k-1} + p^{k-1}(av_{k-1} + bu_{k-1}) \mod \langle T_{n-1}, p^k \rangle.$$

Subtracting $u_{k-1}v_{k-1}$ on both sides and dividing through by p^{k-1} gives

$$\frac{t_n - u_{k-1}v_{k-1}}{p^{k-1}} \equiv av_0 + bu_0 \mod \langle T_{n-1}, p \rangle.$$

Note that $\gcd(u_0, v_0) = 1 \mod \langle T_{n-1}, p \rangle$. Let $c = \frac{t_n - u_{k-1}v_{k-1}}{p^{k-1}}$. By Lemma 4, there exists unique polynomials σ, τ such that $u_0\sigma + v_0\tau \equiv c \mod \langle T_{n-1}, p \rangle$ with $\deg(\sigma) < \deg(v_0)$ and $\deg(\tau) < \deg(u_0)$ since certainly $\deg(c) = \deg(t_n - u_{k-1}v_{k-1}) < \deg(t_n) = \deg(u_0) + \deg(v_0)$. Set $a = \tau$ and $b = \sigma$. Because of these degree constraints, $u_k = u_{k-1} + ap^{k-1}$ has the same leading coefficient as u_{k-1} and hence u_0; in particular u_k is monic. Similarly, v_k is monic as well. By uniqueness of σ and τ, we get uniqueness of u_k and v_k. □

What follows is the formal presentation of the Hensel construction. Algorithm HenselLift takes input $u_0, v_0, f \in R/\langle p \rangle[x]$ where u_0, v_0 are monic and $f = u_0 v_0 \pmod p$. It also requires a bound B that's used to notify termination of the Hensel construction and output FAIL. The HenselLift algorithm can also output ZERODIVISOR(u) if it encounters a zero divisor $u \in R/\langle p \rangle$ in the course of its run.

```
Algorithm 5: HenselLift
Input  : A zero-dimensional, radical triangular set $T \subset \mathbb{Q}[z_1, \ldots, z_n]$, a radical prime $p$,
         polynomials $f \in R[x]$ and $a_0, b_0 \in R/\langle p \rangle[x]$ where $R = \mathbb{Q}[z_1, \ldots, z_n]/T$, and a
         bound $B$. Further, assume $f \equiv a_0 b_0 \pmod p$ and $\gcd(a_0, b_0) = 1$.
Output: Either polynomials $a, b \in R[x]$ where $f = ab$, FAIL if the bound $B$ is reached, or
         ZERODIVISOR($w$) if a zero divisor $w \in R/\langle p \rangle$ is encountered.
Solve $sa_0 + tb_0 = 1$ using the Monic extended Euclidean algorithm for $s, t \in \mathbb{R}/\langle p \rangle[x]$;
if a zero divisor $w$ is encountered then return ZERODIVISOR($w$);
Initialize $u = a_0, v = b_0$ and lift $u$ and $v$ from $R/\langle p \rangle$ to $R$;
for $i = 1, 2, \ldots$ do
    Apply rational reconstruction mod $p^i$ to the coefficients of $u$;
    if rational reconstruction succeeded with output $a$ and $a|f$ in $R[x]$ then return ($a$,$f/a$);
    if $p^i > 2B$ then return FAIL;
    Compute $e := f - uv$ in $R[x]$;
    Set $c := (e/p^i) \mod p$ ;
    Solve $\sigma a_0 + \tau b_0 = c$ for $\sigma, \tau \in R/\langle p \rangle[x]$ using $sa_0 + tb_0 = 1$;
    Lift $\sigma$ and $\tau$ from $R/\langle p \rangle$ to $R$ and set $u := u + \tau p^i$ and $v := v + \sigma p^i$;
end
```

In general f will have fractions thus the error e in our Hensel lifting algorithm will also have fractions and hence it can never become 0. Note the size of the rational coefficients of e grow linearly with i as f is fixed and the magnitude of the integer coefficients in the product uv is bounded by $p^{2i}(1 + \deg u)$.

The standard implementation of Hensel lifting requires a bound on the coefficients of the factors of the polynomial $f \in R[x]$. For the base case $n = 0$ where $R[x] = \mathbb{Q}[x]$ one can use the Mignotte bound (see [7]). For the case $n = 1$ Weinberger and Rothschild [15] give a bound but note that it is large. We do not know of any bounds for the general case $n > 1$. Therefore a more "engineering" based approach is suitable. Since we do not know whether the input zero divisor a_0 is the image of a monic factor of f, we repeat the Hensel lifting each time a zero divisor is encountered in our modular GCD algorithm, first using a bound of 2^{60}, then 2^{120}, then 2^{240} and so on, until the rational coefficients of any monic factor of f can be recovered using rational number reconstruction. In section 4.4 we prove that the Euclidean algorithm over $\mathbb{Q}$ agrees with the Euclidean algorithm modulo a prime for all but finitely many primes, which is enough to show that this strategy terminates.

4.3.2 Lifting Zero-Divisors

The prime application of Hensel lifting will be as a solution to the zero divisor problem. Suppose that we're using the Hensel construction on the factorization $t_n = uv \mod \langle T_{n-1}, p\rangle$. Part of the Hensel construction is running the extended Euclidean algorithm (see Step 1). It's possible that a new zero divisor is encountered. This has to be accounted for.

```
Algorithm 6: HandleZeroDivisorHensel
Input  : A zero-dimensional, radical triangular set $T \subset k[z_1, \dots, z_n]$ and a zero divisor
         $u_0 \in R$ where $R = k[z_1, \dots, z_n]/T$. Assume mvar$(u) = n$.
Output: A message telling ModularC-GCD what to do and any important parameters;
Set $v_0$ :=Quotient$(t_n, u_0)$ (mod $T_{n-1}, p$);
if $v_0$ = ZERODIVISOR$(w)$ then return HandleZeroDivisorHensel$(w)$;
if the global variable $B$ is unassigned then set $B := 2^{60}$ else set $B := B^2$;
Set $u, v$ :=HenselLift$(t_n, u_0, v_0, B)$;
if $u$ = ZERODIVISOR$(w)$ then
    return HandleZeroDivisorHensel$(w)$
else if $u$ =FAIL then
    Tell ModularC-GCD to pick a new prime;
else
    Create two new triangular sets $T^{(u)}$ and $T^{(v)}$ where $t_n$ is replaced by $u$ and $v$;
    Tell ModularC-GCD to recursively compute c-gcd$(a, b)$ modulo $T^{(u)}$ and $T^{(v)}$;
end
```

Now that all algorithms have been given, the following series of lemmatta leads to a proof of correctness for ModularC-GCD.

Lemma 6. Let $R = k[z_1, \dots, z_n]/T$ where T is a radical zero-dimensional triangular set and let $a, b \in R[x]$. If $g = \gcd(a, b)$ is monic, any other $\gcd(a, b)$ has the same degree. In particular, g is the unique monic $\gcd(a, b)$.

Proof. Let h be a $\gcd(a, b)$. Then, $h \mid g$ and $g \mid h$. This implies the existence of u, v where $hu = g$ and $gv = h$. Basic algebraic manipulation gives $g(uv - 1) = 0$. Since g is monic, it can't be a zero divisor. Therefore, v is a unit and so $\deg_x(v) = 0$ by Lemma 2. Thus, $\deg_x(h) = \deg_x(g)$. In particular, if h were monic, then $v = 1$ by comparing leading coefficients of $gv = h$. □

Lemma 7. Let $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ where T is a radical zero-dimensional triangular set. Put $a, b \in R[x]$. Suppose p is a radical prime and $\text{denom}(a)\text{denom}(b)\text{denom}(T)\text{lc}(a)\text{lc}(b) \not\equiv 0 \pmod{p}$. Let $g = \gcd(a, b)$ over $\mathbb{Q}$ be monic, and let $g_p = \gcd(a, b) \pmod{p}$. Then, $\deg_x(g_p) \geq \deg_x(g)$.

Proof. Note that $g \mid a$ and $g \mid b$. Let $\pi \colon \mathbb{Z}_{\langle p \rangle} \to \mathbb{Z}/p\mathbb{Z}$ be reduction by p. Then, we may reduce these equations modulo p to get $\pi(g) \mid \pi(a)$ and $\pi(g) \mid \pi(b)$. This shows $\pi(g)$ is a common divisor of a and b modulo p. Therefore, $\pi(g) \mid g_p$. Since g is monic, $\deg_x(g) = \deg_x(\pi(g)) \leq \deg(g_p)$. □

Theorem 4. Let $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ where T is a radical zero-dimensional triangular set. Put $a, b \in R[x]$. A finite number of zero divisors are encountered when running ModularC-GCD(a, b).

Proof. First, there are a finite number of non-radical primes. So we may assume that T remains radical modulo any chosen prime. Second, consider (theoretically) running the Euclidean algorithm over $\mathbb{Q}$ where we split the triangular set if a zero divisor is encountered. In this process, a finite number of primes divide either denominators or leading coefficients; so we may assume the algorithm isn't choosing these primes without loss of generality.

We use induction on the degree of the extension $\delta = d_1 \cdots d_n$ where $d_i = \text{mdeg}(t_i)$. If $\delta = 1$, then $R = \mathbb{Q}$ so no zero divisors occur. Now, suppose a prime p is chosen by the algorithm and a zero divisor u_p is encountered modulo p at some point of the algorithm. This implies $\gcd(u_p, t_k) \not\equiv 1 \mod \langle T_{k-1}, p \rangle$. We may assume that $u_p = \gcd(u_p, t_k) \mod \langle T_{k-1}, p \rangle$ and that u_p is monic; this is because the Euclidean algorithm with error handling will only output such zero divisors. If u_p lifts to a zero divisor over $\mathbb{Q}$, the algorithm constructs two triangular sets, each with degree smaller than δ. So by induction, a finite number of zero divisors occur in each recursive call. Now, suppose lifting fails. This implies there is some polynomial u over $\mathbb{Q}$ that reduces to u_p modulo p and appears in the theoretical run of the Euclidean algorithm over $\mathbb{Q}$. Note that $\gcd(u, t_k) = 1 \mod \langle T_{k-1} \rangle$ over $\mathbb{Q}$ since we're assuming the lifting failed. By Theorem 3, this happens for only a finite amount of primes. Thus, a finite number of zero divisors are encountered. □

Theorem 5. Let $R = \mathbb{Q}[z_1, \ldots, z_n]/T$ where T is a radical zero-dimensional triangular set and let $a, b \in R[x]$. The modular algorithm using Hensel lifting to handle zero divisors outputs a correct c-gcd if run on a and b.

Proof. It is enough to prove this for a single component of the decomposition. In particular, let h be the monic polynomial returned from the modular algorithm modulo a triangular set T. First, we claim that running EuclideanC-GCD on a and b over $\mathbb{Q}$ terminates without encountering a zero divisor. For if it did, the image of that zero divisor would be encountered modulo a prime p; this is against our assumption of the modular algorithm returning a gcd in this particular component of the decomposition. Now, let $g = \gcd(a, b) \pmod{T}$ be the monic polynomial that's output from running EuclideanC-GCD on a and b over $\mathbb{Q}$. Since h passed the trial division, it follows that $h \mid g$. Since g is monic, $\deg(h) \geq \deg(g)$ by Lemma 7, and $\deg(h) \leq \deg(g)$ since $h \mid g$. Therefore, h must be an associate of g and so is a gcd of a and b. □

4.3.3 Zero-Divisors: FTRR

When trying to recover a polynomial from $\mathbb{Z}_p$ to $\mathbb{Q}$, the techniques often employed in computer algebra are Chinese remaindering and Hensel lifting. Using Chinese remaindering here is not straightforward. To motivate, consider the following example:

Example 8. Consider the triangular set $T = \{z^2 + 14z + 24\}$ and polynomials $a = x^4 + x^3 + (z+3)\,x^2 + (z+4)\,x + 3\,z + 1$ and $b = x^2 + x + z$. The remainder of a divided by b modulo T is $(z + 1)x + 1$. Here, $z + 1$ isn't a zero divisor since $z^2 + 14z + 24 = (z + 2)(z + 12)$. But, if we're working modulo 11, it becomes a zero divisor. Any attempt at combining zero divisors using Chinese remaindering would fail if $p = 11$ was one of the chosen primes.

Abbott's new algorithm Fault Tolerant Rational Reconstruction (FTRR) in [1] circumvents this problem. It can still find the desired value if there are enough correct images. In particular, we use the heuristic algorithm HRR given in [1] which requires a 2-to-1 correct to incorrect images ratio.

Algorithm 7: HandleZeroDivisorHRR

```
Input  : A zero-dimensional, radical triangular set $\mathbb{Z}_p \subset k[z_1, \dots, z_n]$ and a zero divisor
         $u \in R$ where $R = \mathbb{Z}_p[z_1, \dots, z_n]/T$. Assume mvar$(u) = n$.
Output: A message telling modular_cgcd what to do and any important parameters;
Use CRT to combine $u$ with previous zero divisors (if any);
Set $w$ :=HRR$(u)$;
if $w \neq$ FAIL and $w \mid t_n$ over $\mathbb{Q}$ then
    Create two new triangular sets $T^{(w)}$ and $T^{(v)}$ where $t_n$ is replaced by $w$ and $v := t_n/w$;
    return A message to ModularC-GCD instructing it to recursively compute c-gcd$(a, b)$
      modulo $T^{(w)}$ and $T^{(v)}$;
end
return A message to ModularC-GCD instructing it to pick a new prime;
```

Proving this variant of the modular algorithm works is based entirely on the idea of the Euclidean algorithm over $\mathbb{Q}$ agreeing with the Euclidean algorithm over $\mathbb{Z}_p$ for all but finitely many primes. We give a thorough proof of this.

Lemma 8. The Euclidean algorithm over $\mathbb{Q}$ agrees with the Euclidean algorithm modulo a prime for all but finitely many primes.

Proof. The main idea is that the Euclidean algorithm only encounters a finite amount of prime numbers. For a formal proof, see the Appendix. □

Theorem 6. Let $R = \mathbb{Q}[z_1, \dots, z_n]/T$ where T is a radical zero-dimensional triangular set. Put $a, b \in R[x]$. The modular algorithm using HRR to handle zero divisors outputs a correct c-gcd if run on a and b.

Proof. The modular algorithm would eventually encounter a sufficient number of primes from the infinite family of primes that agree with $\mathbb{Q}$ when running the Euclidean algorithm. □

4.4 Implementation Notes and Closing Remarks

We have implemented both variants of the modular algorithm described above. We used Maple's RECDEN package. It provides a recursive dense data structure and support for polynomial computation modulo a triangular set in characteristic 0 and p. Details of RECDEN can be found in Monagan and van Hoeij's paper [9].

Our software is available at `http://www.cecm.sfu.ca/CAG/code/MODGCD`. The reader will find several examples there for running our algorithm. Whenever a zero divisor w is encountered in one of our algorithms we generate an error (a non-local goto) containing u and catch it (using the traperror command in Maple) in the main algorithm (ModularC-GCD) and process it there and print a message.

The purpose of the following example is to illustrate the strengths and weaknesses of the two ways of handling zero divisors. First, let's consider Hensel lifting. Suppose u and v are zero divisors over $\mathbb{Q}$, and the Euclidean algorithm (when run over $\mathbb{Q}$) encounters a leading coefficient $v + pq$. Afterward, it then encounters the zero divisor u and stops. Here, if the modular algorithm chooses p, it will encounter v. This will cause a lifting to occur and it will lift to the actual zero divisor v and not u.

Example 9. This example illustrates a situation in which handling zero divisors with Hensel lifting is superior to FTRR. Suppose we are working in $\mathbb{Q}[z_1, z_2][x]/T$ where $T = \{z_1^2 + 1, z_2^2 + 1\}$ and we want to compute $\gcd(a, b)$ where

$$a = 229x^3 + (-182 + 19z_1 - 17z_2 + 2z_1z_2)x^2 + (-2z_1z_2 - 14z_1 + 16z_2 + 182)x + 34,$$
$$b = (z_1 - z_2 + 15)x^2 - 15x + 15$$

via the monic Euclidean algorithm. Initialize $r_0 = a$ and $r_1 = b$. First, it would determine that $(z_1 - z_2 + 15)^{-1}$ exists and proceed with Euclidean division giving

$$r_3 = (z_1 + z_2)x - 11.$$

Next, it would try inverting $z_1 + z_2$ and determine that it's a zero divisor. The algorithm would then terminate with the error ZERODIVISOR($z_1 + z_2$).

Now, if we were working over $\mathbb{Z}_3$, the first step would be try to invert $z_1+2z_2 \pmod 3$. However, this is a zero divisor too. Oddly enough, it does actually lift to a zero divisor over $\mathbb{Q}$. The Hensel lifting algorithm would compute this and proceed recursively. The FTRR algorithm would store the zero divisor and proceed to the next prime. Assuming it chooses $p = 5$, it would again find an earlier zero divisor $z_1 + 4z_2 \pmod 5$. Next, any other prime will find the image of the zero divisor $z_1 + z_2$ over $\mathbb{Q}$. However, HRR will require at least 4 correct zero divisor images before it can actually combine with the faulty ones.

There are of course situations when FTRR outperforms Hensel lifting. For instance, suppose the Euclidean algorithm over $\mathbb{Q}$ doesn't encounter a single divisor. It's possible that a zero divisor is encountered modulo a prime p. The FTRR would store the found zero divisor and move on the next prime. However, Hensel lifting would attempt to lift the zero divisor, wasting time.

However, if our algorithm uses moderately large primes, say 63 bit primes on a 64 bit machine, then the chance of hitting an unlucky prime or an unlucky zero divisor is very low. That is, if the monic Euclidean algorithm when run modulo p it encounters a zero divisor, it is very likely the image of a zero divisor over $\mathbb{Q}$. Thus it is best to use the Hensel lifting approach to determine the zero divisor over $\mathbb{Q}$.

We note that if the Euclidean algorithm does not encounter a zero divisor over $\mathbb{Q}$ then the additional cost of our algorithm over the modular GCD algorithm of Monagan and van Hoeij is the cost of checking if the prime is a radical prime (Algorithm isRadical) which is small. For a gcd problem with 3 extensions of degree 5, and for inputs $a = g\bar{a}$ and $b = g\bar{b}$, we generated $g, \bar{a}, \bar{b}$ of degree 5 in x, dense, with 10 digit random coefficients. Three 31 bit primes were sufficient to recover g and less than 0.5% of the time was spent checking if the primes were radical.

References

[1] John Abbott. Fault-tolerant modular reconstruction of rational numbers. Journal of Symbolic Computation Volume 80, pages 707 – 718. May-June 2017.

[2] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. J. Symb. Comp., **28**: 105 – 124, 1999.

[3] S. Bosch. Algebraic Geometry and Commutative Algebra. Springer-Verlag London. 2013.

[4] W. S. Brown. On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. J. ACM **18**: 478 – 504. 1971.

[5] D. Cox, J. Little, D. O'Shea. Ideals, Varieties and Algorithms. Springer-Verlag, 1991.

[6] M. J. Encarnacion. Computing GCDs of Polynomials over Algebraic Number Fields, J. Symb. Comp. **20**: 299 – 313, 1995.

[7] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed., Cambridge University Press, 2013.

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

[9] Mark van Hoeij and Michael Monagan, A modular GCD algorithm over number fields presented with multiple extensions. Proceedings of ISSAC 02, ACM Press, pp. 109 – 116. 2002.

[10] E. Hubert. Notes on Triangular Sets and Triangulation-Decomposition Algorithms I: Polynomial Systems. In Symbolic and Numerical Scientific Computing edited by F. Winkler and U. Langer. Lecture Notes in Computer Science 2630, pp. 1 – 39. 2003.

[11] L. Langemyr, S. McCallum. The Computation of Polynomial GCDs over an Algebraic Number Field, J. Symbolic Computation 8, pp. 429 – 448. 1989.

[12] Xin Li, Marc Moreno Maza, and Wei Pan. Gcd computations modulo regular chains. Technical report, Univ. Western Ontario. 2009.

[13] M. B. Monagan. Maximal Quotient Rational Reconstruction: An Almost Optimal Algorithm for Rational Reconstruction. *Proceedings of ISSAC '2004*, ACM Press, pp. 243–249, 2004.

[14] Paul S. Wang, M.J.T. Guy, and J.H. Davenport. P-adic reconstruction of rational numbers. ACM SIGSAM Bulletin **16**(2): 2–3, 1982.

[15] P.J. Weinberger and L.P. Rothschild. Factoring Polynomials over Algebraic Number Fields. *ACM Trans. on Math. Soft.* **2**(4): 335–350, 1976.

Appendix

Proposition 2. Suppose $\psi\colon R \to R_1 \times R_2$ is a ring isomorphism. Let π_1, π_2 be the canonical projections $R_1 \times R_2$ to R_1, R_2, respectively. Let $a, b \in R$ and $g_1 = \gcd(\pi_1\psi(a), \pi_1\psi(b))$, $g_2 = \gcd(\pi_2\psi(a), \pi_2\psi(b))$. Then, $g = \psi^{-1}(g_1, g_2)$ is a gcd of a and b.

Proof. There are two things to show: (i) g is a common divisor of a and b, (ii) any common divisor d of a and b is a divisor of g. For (i), note that $g_1q_1 = \pi_1\psi(a)$ for some $q_1 \in R_1$, and $g_2q_2 = \pi_2\psi(a)$ for some $q_2 \in R_2$. Then,

$$a = \psi^{-1}(\pi_1\psi(a), \pi_2\psi(a)) = \psi^{-1}(g_1q_1, g_2q_2) = \psi^{-1}(g_1, g_2)\psi^{-1}(q_1, q_2) = gq$$

where $q = \psi^{-1}(q_1, q_2)$. This shows $g \mid a$, and similarly $g \mid b$. For (ii), consider $d \in R$ where $d \mid a$ and $d \mid b$. Then, $\pi_i\psi(d) \mid \pi_i\psi(a)$ and $\pi_i\psi(d) \mid \pi_i\psi(b)$ since π_i, ψ are homomorphisms. By properties of greatest common divisors, $d \mid g_1$ and $d \mid g_2$. Then, there exists $(q_1, q_2) \in R_1 \times R_2$ where $(g_1, g_2) = (\pi_1\psi(d)q_1, \pi_2\psi(d)q_2)$. Next,

$$g = \psi^{-1}(g_1, g_2) = \psi^{-1}(\pi_1\psi(d)q_1, \pi_2\psi(d)q_2) = \psi^{-1}(\pi_1\psi(d), \pi_2\psi(d))\psi^{-1}(q_1, q_2) = dq$$

where $q = \psi^{-1}(q_1, q_2)$. This shows $d \mid g$, and hence $g = \gcd(a, b)$. □

Theorem 1. Let $T \subseteq k[z_1, \ldots, z_n]$ be a triangular set and $I = \langle T \rangle$. Then, $k[z_1, \ldots, z_n]/I$ is isomorphic to a direct product of fields if and only if I zero-dimensional and radical.

Proof. ($\Longleftarrow$) Induct on the number of variables n. Our induction hypothesis will give more details on the fields that occur:

> **Induction hypothesis**: Let $S \subseteq k[z_1, \ldots, z_{n-1}]$ be a triangular set. Then, $k[z_1, \ldots, z_{n-1}]/\langle S \rangle$ is isomorphic to a direct product of fields if and only if $\langle S \rangle$ is zero-dimensional and radical. Further, the fields are of the form $k[z_1, \ldots, z_{n-1}]/\langle P \rangle$ where P is a triangular set made up of polynomials m_i where $\text{mvar}(m_i) = z_i$ and z_i divides the polynomial in T with main variable z_i.

If $n = 1$, use the previous proposition. Let $J = I \cap k[z_1, \ldots, z_{n-1}]$, an ideal of $k[z_1, \ldots, z_{n-1}]$. I claim J is radical. To see this, suppose $g^k \in J$. Then, $g^k \in I$ which implies $g \in I$ by I being radical. Also, $g \in k[z_1, \ldots, z_{n-1}]$ by assumption. Therefore, $g \in J$. I also claim that $\mathbf{V}(J)$ is finite. To verify, if $\mathbf{V}(J)$ were infinite, then we could extend zeros of J to zeros of I. So, $k[z_1, \ldots, z_{n-1}]/J \cong \prod F_i$ where F_i are as in the induction hypothesis. Then,

$$(k[z_1, \ldots, z_{n-1}]/J)[z_n] \cong (\prod F_i)[z_n] = \prod F_i[z_n].$$

Recall that $f \in T$ is the polynomial with $\text{mvar}(f) = z_n$. Observe that

$$k[z_1, \ldots, z_n]/I = (k[z_1, \ldots, z_{n-1}]/J)[z_n]/f \cong (\prod F_i[z_n])/f \cong \prod F_i[z_n]/f$$

where we are assuming that f is reduced appropriately in F_i. I claim that f is square-free and nonzero in $F_i[z_n]$. If f evaluated to 0 in F_i, then taking the roots associated to the field F_i and any other field element for z_n would produce infinitely many zeros for $V(T)$, which contradicts the finiteness assumption. Next, if f weren't square-free, there would be some $g^k \mid f$ in F_i and so $(0, \ldots, g, \ldots, 0)$, where g is in the ith coordinate, would be nilpotent. This implies $k[z_1, \ldots, z_n]/I$ has a nilpotent element, contradicting that I is radical. Thus, by the previous theorem each $F_i[z_n]/f$ will be isomorphic to a product of fields by the previous proposition. This completes the inductive step and this direction of the proof.

($\Longrightarrow$) Since $k[z_1, \ldots, z_n]/I \cong \prod F_i$ is isomorphic to a direct product of fields, it will have no nilpotent elements. Therefore, I is radical. Any subfield of $k[z_1, \ldots, z_n]/I$ will be of the form $k[z_1, \ldots, z_n]/\langle f_1, \ldots, f_k \rangle$ for some polynomials $f_i \in k[z_1, \ldots, z_n]$. Therefore, it will be a finitely generated algebraic extension of k, and hence a finite extension. Since each of fields F_i will be finite extensions, we can view $k[z_1, \ldots, z_n]/I$ as a k-vector space of finite dimension. Therefore, by Theorem 6 on page 234 of Ideals, Varieties, and Algorithms, $V(T)$ is finite. □

Lemma 8. The Euclidean algorithm over $\mathbb{Q}$ agrees with the Euclidean algorithm modulo a prime for all but finitely many primes.

Proof. Note that the Euclidean algorithm consists of a finite amount of divisions; so it enough to show that a single division over $\mathbb{Q}$ agrees with a single division modulo all but finitely many primes. In symbols, let $a, b \in R[x]$ with $\deg_x(b) \leq \deg_x(a)$ and remainder r when a is divided by b. Work by induction on n, the number of extensions. Consider the base case $n = 0$. Let $a = bq + r$ and p be a prime number. As long as p doesn't divide any of the relevant denominators, this equation can be reduced modulo p. Further, if $p \not| \operatorname{lc}(b)$, then $\deg(r \mod p) < \deg(b \mod p)$. Since there are finitely many primes that cause these issues, the base case is satisfied. Now consider the nth case. In the process of dividing a by b over $\mathbb{Q}$, numerous smaller degree divisions must be done, in particular, they all only use $n - 1$ or less extensions. Use the induction hypothesis here to rule out a set of finitely many primes S. Take a prime number $p \notin S$. If a zero divisor is encountered in one of the smaller divisions, the algorithm would terminate both over $\mathbb{Q}$ and $\mathbb{Z}_p$ with the same zero divisor after reduction modulo p. So suppose no zero divisors are encountered in any of these smaller divisions. In particular, $\operatorname{lc}(b)$ is a unit over $\mathbb{Q}$ and $\mathbb{Z}_p$. Next, $\deg_x(b \mod p) < \deg_x(b)$ is only true for primes $p \mid \operatorname{lc}(b)$, which happens for finitely many primes. We may safely disregard these. After that, the uniqueness of remainders establishes that the remainders are the same after reduction by p. □