Computing GCDs of Multivariate Polynomials over Algebraic Number Fields with Multiple Extensions

Mahsa Ansari
Michael Monagan

Department of Mathematics,
Simon Fraser University,
Canada

Outline

Algebraic Number Fields

Algebraic Number Field

Let $\alpha_1, \ldots, \alpha_n$ be algebraic numbers. The **algebraic number field** $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ is the smallest field containing $\mathbb{Q}$ and $\alpha_1, \ldots, \alpha_n$.

Theorem (1)

Let $\alpha_1, \ldots, \alpha_n$ be algebraic numbers. There exists $\gamma \in \mathbb{C}$ s.t

$$\mathbb{Q}(\alpha_1, \ldots, \alpha_n) = \mathbb{Q}(\gamma).$$

*We call γ a **primitive element**.*

Example

Let $\alpha = \sqrt{2}$, $\beta = \sqrt{3}$ and $\gamma = \alpha + \beta$. Then, $\mathbb{Q}(\alpha, \beta) = \mathbb{Q}(\gamma)$.

Outline

History

1987-Langemyr and McCallum

Designed a modular gcd algorithm for $\mathbb{Q}(\alpha)[x]$.

History

1987-Langemyr and McCallum

Designed a modular gcd algorithm for $\mathbb{Q}(\alpha)[x]$.

1995- Encarnacion

Used rational number reconstruction to make the algorithm for $\mathbb{Q}(\alpha)[x]$ output sensitive.

History

1987-Langemyr and McCallum

Designed a modular gcd algorithm for $\mathbb{Q}(\alpha)[x]$.

1995- Encarnacion

Used rational number reconstruction to make the algorithm for $\mathbb{Q}(\alpha)[x]$ output sensitive.

2002-Monagan and Van Hoeij

Introduced another algorithm to compute the GCDs of two polynomials over $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x]$.

Our Contributions

Let $f_1, f_2 \in \mathbb{Q}(\alpha_1, \ldots, \alpha_n)[x_1, \ldots, x_k]$.

1. We designed a modular gcd algorithm called MGCD to compute the monic $\gcd(f_1, f_2)$.
2. To speed up our algorithm, we use linear algebra to map $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ into $\mathbb{Q}(\gamma)$, where γ is a primitive element. We do this mod a prime to avoid expression swell.
3. A Maple implementation using a recursive dense representation for polynomials.
4. Analysis of the expected time complexity.

Outline

Preliminaries

In order to improve computational efficiency, in a preprocessing step in MGCD, we eliminate fractions from the input polynomials f_1 and f_2 and the minimal polynomials $M_1, \ldots, M_n$.

Denominator and Semi-associate

Let $L_{\mathbb{Z}} = \mathbb{Z}[\alpha_1, \ldots, \alpha_n]$.
Let $f = \frac{3}{2}\alpha_1 x + \alpha_2 \in \mathbb{Q}(\alpha_1, \alpha_2)[x]$ where $\alpha_1 = \sqrt{2}$ and $\alpha_2 = \sqrt{3}$.

- The denominator of f, denoted by $den(f)$, is the smallest positive integer for which $den(f)f \in L_{\mathbb{Z}}[x]$. Here, $den(f) = 2$.
- The semi-associate of f is $\check{f} = rf$ where r is the smallest rational number such that $den(rf) = 1$. Here, $r = 2$ and $\check{f} = 3\alpha_1 x + 2\alpha_2$.

Computation over $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$

Question: How can we do computation over $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$?
Let

$$\begin{aligned} L_0 &= \mathbb{Q} \\ L_i &= L_{i-1}/\langle M_i(z_i) \rangle \quad for\ 1 \le i \le n \\ L_n &= L \end{aligned}$$

where $M_i(z_i)$ is the minimal polynomial of α_i over L_{i-1}.

$$\mathbb{Q}(\alpha_1, \ldots, \alpha_n) \cong L$$

Answer: We map the operands from $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ to L and do the computation over L.

Remark

In our algorithm MGCD we suppose that we are given the minimal polynomials $M_1(z_1), \ldots, M_n(z_n)$ of algebraic numbers $\alpha_1, \ldots, \alpha_n$.

- L can be specified as a $\mathbb{Q}-$vector space.
- Let $d_i = \deg(M_i(z_i))$ and $\prod_{i=1}^{n} d_i = d$. Then, $B_L := \{\prod_{i=1}^{n}(z_i)^{e_i} | \ 0 \leq e_i < d_i\}$ is a basis for L.
- $| B_L | = [L : \mathbb{Q}] = d$

Example

We are given the field $L = \mathbb{Q}[z_1, z_2]/\langle z_1^2 - 2, z_2^2 - 3\rangle$ with basis $B_L = \{1, z_2, z_1, z_1 z_2\}$. If $f = 2z_1 x + y + z_1 + z_1 z_2 \in L[x, y]$, then $[f]_{B_L} = [y, 0, 2x + 1, 1]^T$.

Outline

MGCD

Let $f_1, f_2 \in L[x_1, \ldots, x_k]$, MGCD computes $g = \textit{monic}\,\gcd(f_1, f_2)$.

$$
\begin{array}{ccc}
\check{f}_1, \check{f}_2 \in L_{\mathbb{Z}}[x_1, \ldots, x_k] & \longrightarrow & \gcd(f_1, f_2) \in L[x_1, \ldots, x_k] \\
\phi_p \; for \; p \in \{p_1, p_2, \ldots\} \Big\downarrow p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i) \cdot \mathrm{lc}(\check{f}_1) & & \Big\uparrow CRT,\ RNR,\ Division\ test \\
\check{f}_1, \check{f}_2 \in L_p[x_1, \ldots, x_k] & & \gcd(\check{f}_1, \check{f}_2) \in L_p[x_1, \ldots, x_k] \\
\Big\downarrow \phi_\gamma & & \Big\uparrow \phi_\gamma^{-1} \\
\check{f}_1, \check{f}_2 \in \bar{L}_p[x_1, \ldots, x_k] & \xrightarrow{PGCD} & \gcd(\check{f}_1, \check{f}_2) \in \bar{L}_p[x_1, \ldots, x_k]
\end{array}
$$

$L = \mathbb{Q}[z_1, \ldots, z_n]/\langle M_1(z_1), \ldots, M_n(z_n)\rangle$
$L_{\mathbb{Z}} = \mathbb{Z}[\alpha_1, \ldots, \alpha_n]$
$L_p = \mathbb{Z}_p[z_1, \ldots, z_n]/\langle m_1(z_1), \ldots, m_n(z_n)\rangle$ s.t $m_i(z_i) = \check{M}_i(z_i) \mod p$
$\bar{L}_p = \mathbb{Z}_p[z]/\langle M(z)\rangle$

Reduction mod p

The modular homomorphism

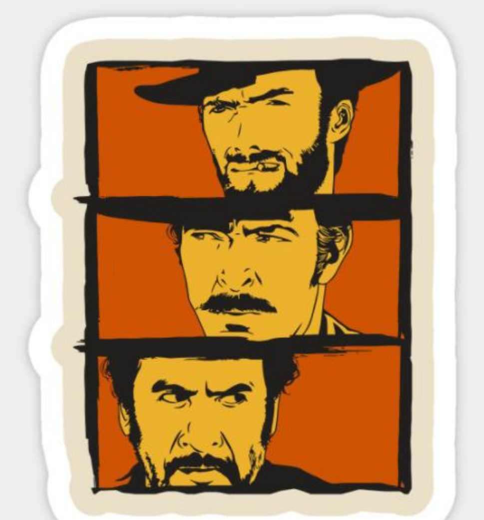
$\phi_p : \mathbb{Z} \longrightarrow \mathbb{Z}_p$ maps integers into their remainder modulo p. We choose p to be a prime so $\mathbb{Z}_p$ is a finite field.

$$
\begin{array}{ccc}
\check{f}_1, \check{f}_2 \in L_{\mathbb{Z}}[x_1, \ldots, x_k] & \longrightarrow & \gcd(f_1, f_2) \in L[x_1, \ldots, x_k] \\
\phi_p \ for \ p \in \{p_1, p_2, \ldots\} \Big\downarrow p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i) \cdot \mathrm{lc}(\check{f}_1) & & \Big\uparrow CRT,\ RNR,\ Division\ test \\
\check{f}_1, \check{f}_2 \in L_p[x_1, \ldots, x_k] & & \gcd(\check{f}_1, \check{f}_2) \in L_p[x_1, \ldots, x_k] \\
\Big\downarrow \phi_\gamma & & \Big\uparrow \phi_\gamma^{-1} \\
\check{f}_1, \check{f}_2 \in \bar{L}_p[x_1, \ldots, x_k] & \xrightarrow{PGCD} & \gcd(\check{f}_1, \check{f}_2) \in \bar{L}_p[x_1, \ldots, x_k]
\end{array}
$$

Lc-bad, Zero-divisor, and Unlucky Primes

Question: Can we use any prime?

Answer: No!!!



Lc-bad, Det-bad, Zero-divisor, Unlucky primes

- Lc-bad Prime. If p divides $\mathrm{lc}(\check{f}_1)$ or any $\mathrm{lc}(\check{M}_1(z_1)), \ldots, \mathrm{lc}(\check{M}_n(z_n))$, then we call p an lc-bad prime.
- Det-bad Prime. Let B_{L_p} be a basis of L_p and γ be a primitive element and $A = \begin{bmatrix} \begin{bmatrix} \vdots \\ 1 \\ \vdots \end{bmatrix}_{B_{L_p}} & \cdots & \begin{bmatrix} \vdots \\ \gamma^{d-1} \\ \vdots \end{bmatrix}_{B_{L_p}} \end{bmatrix}$.

  If $\det(A) \mod p = 0$, then p is called a det-bad prime.
- Zero-divisor Prime. If p is neither an lc-bad nor a det-bad prime and the PGCD algorithm tries to invert a zero-divisor over $\bar{L}_p = \mathbb{Z}_p[z]/\langle M(z) \rangle$, we call p a zero-divisor prime.
- Unlucky Prime. Let $g_p = \gcd(\phi_p(\check{f}_1), \phi_p(\check{f}_2))$. If $\mathrm{lm}(g_p) > \mathrm{lm}(\gcd(f_1, f_2))$, then we call p an unlucky prime. The results of these primes must be ignored.
- Good Prime. If prime p is not an lc-bad, det-bad, unlucky, or zero-divisor prime, we define it as a good prime.

Example

Let $L = \mathbb{Q}[z, w]/\langle z^2 - 2, w^2 - 3\rangle$. Let

$$f_1 = (x + w)(5x + 2w + z)xw$$
$$f_2 = (x + w)(5x + 9w + z)$$

be two polynomials in $L[x, y]$. Let fix the lexicographic order with $x > y$.

- $p_1 = 5 \Longrightarrow \phi_5(\mathrm{lc}(\check{f}_1)) = 0 \mod p_1$. Hence, p_1 is an lc-bad prime.
- $p_2 = 7 \Longrightarrow g_{p_2} = \gcd(\phi_{p_2}(\check{f}_1), \phi_{p_2}(\check{f}_2)) = (x + w)(5x + 2w + z)$ while $g = \gcd(f_1, f_2) = (x + w)$. Since $\mathrm{lm}(g_{p_2}) > \mathrm{lm}(g)$, we can conclude that p_2 is an unlucky prime.
- $p_3 = 3 \Longrightarrow \mathrm{lc}(\check{f}_1) = 2w$ and $w^2 - 3 = w^2 \mod p_3$ so w is not invertible. Thus, p_3 is a zero-divisor prime.

The isomorphism ϕ_γ

$\phi_\gamma : L_p[x_1,\ldots,x_k] \longrightarrow \bar{L}_p[x_1,\ldots,x_k]$ maps f over $\mathbb{Z}_p[z_1,\ldots,z_n]/\langle m_1(z_1),\ldots,m_n(z_n)\rangle$ to its corresponding polynomial over $\mathbb{Z}_p[z]/\langle M(z)\rangle$.

$$
\begin{array}{ccc}
\check{f}_1, \check{f}_2 \in L_{\mathbb{Z}}[x_1,\ldots,x_k] & \longrightarrow & \gcd(f_1,f_2) \in L[x_1,\ldots,x_k] \\
\phi_p \ for \ p \in \{p_1,p_2,\ldots\} \Big\downarrow p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i)\cdot \mathrm{lc}(\check{f}_1) & & \Big\uparrow CRT,\ RNR,\ Division\ test \\
\check{f}_1, \check{f}_2 \in L_p[x_1,\ldots,x_k] & & \gcd(\check{f}_1,\check{f}_2) \in L_p[x_1,\ldots,x_k] \\
\Big\downarrow \phi_\gamma & & \Big\uparrow \phi_\gamma^{-1} \\
\check{f}_1, \check{f}_2 \in \bar{L}_p[x_1,\ldots,x_k] & \xrightarrow{PGCD} & \gcd(\check{f}_1,\check{f}_2) \in \bar{L}_p[x_1,\ldots,x_k]
\end{array}
$$

How to compute a primitive element γ and its minimal polynomial $M(z)$

We use the LAminpoly algorithm:

1. Choose $C_1, \ldots, C_{n-1} \in \mathbb{Z}$ randomly from the interval $[1, p)$ where p is a large prime. Set $\gamma = \alpha_1 + \sum_{i=2}^{n} C_{i-1}\alpha_i$.
2. Let B_L be a basis for L. We build the $d \times d$ matrix
$$A = \left[\begin{bmatrix} \vdots \\ 1 \\ \vdots \end{bmatrix}_{B_L} \quad \cdots \quad \begin{bmatrix} \vdots \\ \gamma^{d-1} \\ \vdots \end{bmatrix}_{B_L} \right].$$
3. If $\det(A) \neq 0$, then γ is a primitive element and we can create its minimal polynomial by solving $\boxed{A \cdot q = -[\gamma^d]_{B_L}}$ and setting $M(z) = z^d + \sum_{i=1}^{d} q_i z^{i-1}$.

We can do these computations over two ground fields $F = \mathbb{Q}$ and $\boxed{F = \mathbb{Z}_p}$.

Computation over $F = \mathbb{Q}$

Theorem (3)

Let $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$ *have degree* d *and* $C_1, \ldots, C_{n-1} \in \mathbb{Z}$ *be chosen randomly from* $[1, p)$ *where* p *is a large prime. Define* $\gamma = \alpha_1 + \sum_{i=2}^{n} C_{i-1}\alpha_i$, *and let* B_L *be a basis for* $\mathbb{Q}(\alpha_1, \ldots, \alpha_n)$. *Let* A *be the* $d \times d$ *matrix whose* i*th column is* $[\gamma^{i-1}]_{B_L}$ *for* $1 \le i \le d$.
γ *is a primitive element for* $\mathbb{Q}(\alpha_1, \ldots, \alpha_n) \iff \det(A) \neq 0$.

Corollary

Under the assumptions of Theorem 3, if $\det(A) \neq 0$ *and* $q = [q_1, \ldots, q_d]^T$ *be the solution of the linear system* $A \cdot q = -[\gamma^d]_{B_L}$, *the polynomial* $M(z) = z^d + \sum_{i=1}^{d} q_i z^{i-1}$ *is the minimal polynomial of* γ.

Computation over $F = \mathbb{Z}_p$

If p is not an lc-bad prime and $\det(A) \mod p \neq 0$

$$\begin{aligned} m_i(z_i) &= M_i(z_i) \mod p \\ L_p &= \mathbb{Z}_p[z_1, \ldots, z_n]/\langle m_1(z_1), \ldots, m_n(z_n)\rangle \\ \bar{L}_p &= \mathbb{Z}_p[z]/\langle M(z)\rangle \end{aligned}$$

Remark

It is likely that one or more of $m_i(z_i)$ are reducible and so $M(z)$ is likely to be reducible over $\mathbb{Z}_p$. That is L_p and $\bar{L}_p$ may not be fields.

$M(z)$ still generates the quotient ring $\bar{L}_p$ such that $L_p \cong \bar{L}_p$.

The isomorphism ϕ_γ

Let B_{L_p} and $B_{\bar{L}_p}$ be two basis of L_p and $\bar{L}_p$ respectively.

$$C : L_p \longrightarrow \mathbb{Z}_p^d \quad s.t \quad C(a) = [a]_{B_{L_p}}$$

$$D : \bar{L}_p \longrightarrow \mathbb{Z}_p^d \quad s.t \quad D(b) = [b]_{B_{\bar{L}_p}}$$

If $\det(A) \mod p \neq 0$, then

$$\phi : L_p \longrightarrow \bar{L}_p \quad s.t \quad \phi(a) = D^{-1}(A^{-1} \cdot C(a))$$

$$\phi^{-1} : \bar{L}_p \longrightarrow L_p \quad s.t \quad \phi^{-1}(b) = C^{-1}(A \cdot D(b))$$

Theorem (4)

If $\det(A) \mod p \neq 0$, *then there exists a ring isomorphism* $\phi : L_p \longrightarrow \bar{L}_p$ *which induces the natural isomorphism* $\phi_\gamma : L_p[x_1, \ldots, x_k] \longrightarrow \bar{L}_p[x_1, \ldots, x_k]$.

PGCD

PGCD

Algorithm PGCD is a recursive algorithm to find the monic gcd of two polynomials f_1, f_2 in $\bar{L}_p[x_1, \ldots, x_k]$ where p is a prime and $k \geq 2$.

$$
\begin{array}{ccc}
\check{f}_1, \check{f}_2 \in L_{\mathbb{Z}}[x_1, \ldots, x_k] & \longrightarrow & \gcd(f_1, f_2) \in L[x_1, \ldots, x_k] \\
\phi_p \ for \ p \in \{p_1, p_2, \ldots\} \Big\downarrow p \nmid \prod_{i=1}^{n} \mathrm{lc}(\check{M}_i) \cdot \mathrm{lc}(\check{f}_1) & & \Big\uparrow CRT, \ RNR, \ Division \ test \\
\check{f}_1, \check{f}_2 \in L_p[x_1, \ldots, x_k] & & \gcd(\check{f}_1, \check{f}_2) \in L_p[x_1, \ldots, x_k] \\
\Big\downarrow \phi_\gamma & & \Big\uparrow \phi_\gamma^{-1} \\
\check{f}_1, \check{f}_2 \in \bar{L}_p[x_1, \ldots, x_k] & \xrightarrow{PGCD} & \gcd(\check{f}_1, \check{f}_2) \in \bar{L}_p[x_1, \ldots, x_k]
\end{array}
$$

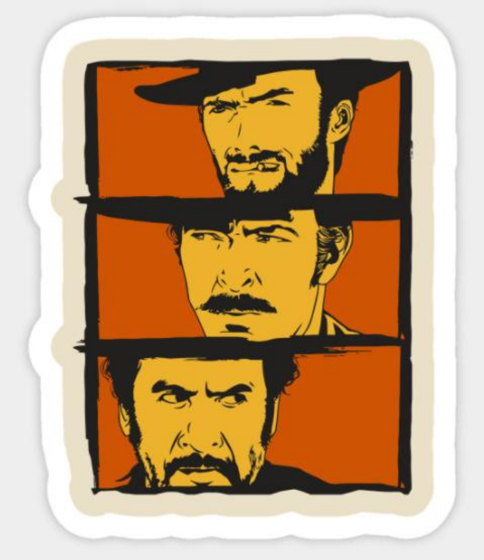
PGCD

Let $f_1, f_2 \in \bar{L}_p[x_1, \ldots, x_k]$ and $g = monic(\gcd(f_1, f_2))$.

1. If $k = 1$, then PGCD calls the MEA to find $\gcd(f_1, f_2) \in \bar{L}_p[x_1]$. PGCD might fail.
2. PGCD uses dense evaluation and interpolation to recover $x_2, \ldots, x_k$.

Question: Can we use any evaluation point?

Answer: No!!!



Outline

Benchmark

Let $L = \mathbb{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7}, \sqrt{11})$ have degree 32. The input polynomials $f_1, f_2 \in L[x, y]$ have degree d in x and y and their gcd g has degree 2 in x and y.

d	New MGCD			Old MGCD	
	time	LAMP	PGCD	time	PGCD
4	0.119	0.023	0.027	0.114	0.100
6	0.137	0.016	0.034	0.184	0.156
8	0.217	0.018	0.045	0.330	0.244
10	0.252	0.018	0.087	0.479	0.400
12	0.352	0.018	0.078	0.714	0.511
16	0.599	0.017	0.129	1.244	1.008
20	0.767	0.017	0.161	1.965	1.643
24	1.103	0.019	0.220	2.896	2.342
28	1.890	0.023	0.358	4.487	3.897
32	2.002	0.020	0.392	5.416	4.454
36	2.461	0.017	0.595	6.944	5.883
40	3.298	0.019	0.772	9.492	7.960

Outline

Complexity

Let $f \in L[x_1, \cdots, x_k]$.

1. The height of f, denoted by $H(f)$, is the magnitude of the largest integer coefficient of $\check{f}$.
2. Let $\#f$ denote the number of terms of f.
3. We assume that multiplication and inverses in $\bar{L}_p$ cost $O(d^2)$.

Theorem

The expected time complexity of our MGCD algorithm is

$$O(N(M + CT_f)d + Nd^2(d + T_f + T_g) + Nd^2D^{k+1} + N^2dT_g)$$

- N is the number of good primes needed to reconstruct the monic gcd g
- $T_f = \max(\#f_1, \#f_2)$ and $T_g = \#g$
- $M = \log \max_{i=1}^{n} H(\check{m}_i)$ and $C = \log \max(H(\check{f}_1), H(\check{f}_2))$.
- $D = \max_{i=1}^{k} \max(\deg(f_1, x_i), \deg(f_2, x_i))$ and $d = [L : \mathbb{Q}]$.

Outline

Future Work

1. Compute the probability of getting unlucky, zero-divisor, bad primes, and evaluation points.
2. Compute the probability that MGCD obtains an incorrect answer.
3. Apply Fast Multiplication and Fast Division algorithms in $\mathbb{Z}_p[z]$ to speed up arithmetic over $\bar{L}_p = \mathbb{Z}_p[z]/\langle M(z)\rangle$.

Thank you!