



Sparse Modular GCD algorithm

Mahdi Javadi
sjavadi@sfu.ca

Department of Computing Science
Simon Fraser University

Problem Statement

- We want to compute $g = \gcd(f_1, f_2)$.
- $f_1, f_2 \in L[x_1, \dots, x_n]$.
- $L = \mathbb{Q}(t_1, \dots, t_k)[z_1, \dots, z_r] / \langle m_1, \dots, m_r \rangle$.
- $m_i \in \mathbb{Q}(t_1, \dots, t_k)[z_1, \dots, z_{i-1}][z_i]$.

Example:

$$f_1 = x_1^2 - \frac{2t_1}{t_2} z_2 x_2^2 + 3t_1^2 x_1 z_1 + 5t_2 x_1 x_2 + \frac{(1-t_2)}{t_1} z_2.$$
$$m_1(z_1) = z_1^2 - t_1, m_2(z_2) = z_2^3 - z_1 z_2 + t_1 t_2^2 - 1.$$

Problem Statement (contd.)

- Euclidean Algorithm blows up on large examples.
- Is not directly applicable to multivariate polynomials.
- Brown's modular GCD algorithm: $\mathbb{Z}[x_1, \dots, x_n]$
- Zippel's Sparse Interpolation: $\mathbb{Z}[x_1, \dots, x_n]$
- Encarnacion algorithm: $\mathbb{Q}(\alpha)[x]$
- ModGcd (Monagan and van Hoeij): $L[x]$, one field extension
- SparseModGcd: $L[x]$, multiple field extensions



Outline

- Encarnacion's algorithm
- Theory for ModGcd algorithm over $L[x]$
- SparseModGcd algorithm
- Bottlenecks of implementation
- Benchmarks

Encarnacion's algorithm

- Works for a number field and with one field extension
- $g = \gcd(f_1, f_2) \pmod{p}$ is unique up to a scalar multiple.
- Encarnacion's algorithm uses rational number reconstruction.

Example: Let $m(z) = z^2 - 2$ and $L = \mathbb{Q}(z)$.

Let $g = \gcd(f_1, f_2) = 3x^2 + (2z - 2)x + z - 1$ and $p_1 = 11$.

The first image is: $h = x^2 + (8z + 3)x + 4z + 7$.

Rational number reconstruction: $h' = x^2 + \left(\frac{2}{3}z - \frac{2}{3}\right)x + \frac{1}{3}z - \frac{1}{3}$.

Clearing the denominator: $g_{11} = 3h' = 3x^2 + (2z - 2)x + z - 1$.

Encarnacion's algorithm (contd.)

- Zero Divisors. **Example:** Let $f_1 = x^4 + (z - 2)x^2 + zx + 1$, $f_2 = (z - 3)x^3 + x + 2z$ and $m(z) = z^2 - 2$. Choose $p_1 = 7$, divide $f_1 \bmod p_1$ by $f_2 \bmod p_1$. Hit a zero divisor while inverting $\text{lc}(f_2) = z - 3$.
- Notation: $f \in L[x] \Rightarrow \check{f} = \text{den}(f)f \in \mathbb{Z}[z, x]$.
- Let g be the monic $\text{gcd}(f_1, f_2) \in \mathbb{Q}[z, x]$. We require that $p \nmid \text{lc}(\check{g}) \in \mathbb{Z}$.
- Encarnacion (1994): suffices $p \nmid l\Delta$ where $\Delta = \text{res}_z(\check{m}, \check{m}')$ and $l = \text{res}_z(\text{lc}(\check{f}_2), \check{m}(z))$
- Monagan and van Hoeij (2002): Suffices to require $\text{lc}(\check{f}_2) \not\equiv 0 \pmod{p}, \text{lc}(\check{m}) \not\equiv 0 \pmod{p}$.



ModGcd Algorithm

- ModGcd is a modular algorithm.
- Uses polynomial evaluation and interpolation.
- Euclidean algorithm is used to compute univariate gcds.
- Uses Rational number and univariate rational function reconstruction.
- Is output sensitive , Uses Trial Division
- Time complexity is of $O(d^k)$ which is exponential in the number of parameters

ModGcd Algorithm (contd.)

- Let $D = \mathbb{Z}[t_1, \dots, t_k]$.
- *denominator* of f : $\text{den}(f) \in D$ of least total degree in (t_1, \dots, t_k) and with smallest integer content such that $\text{den}(f)f \in D[z_1, \dots, z_r, x_1, \dots, x_n]$.
- *primitive associate* \check{f} : associate of $\text{den}(f)f$ which is primitive in $D[z_1, \dots, z_r, x_1, \dots, x_n]$ and has positive leading coefficient in a term ordering.
 - **Example:** $f = 2x^2 + \frac{2}{t}zx + \frac{2}{3} \Rightarrow \check{f} = 3tx^2 + 3zx + t.$

ModGcd Algorithm (contd.)

■ Bad primes and evaluation points

- p is bad $\Leftrightarrow \text{lc}_x(\check{f}_1)$ or $\text{lc}_x(\check{f}_2)$ or $\text{lc}_{z_i}(\check{m}_i)$ vanishes mod p .
- α is bad $\Leftrightarrow \text{lc}_x(\check{f}_i(\alpha)) = 0$ or $\text{lc}_{z_i}(\check{m}_i(\alpha)) = 0$.

■ Unlucky primes and evaluation points

- p is unlucky $\Leftrightarrow \deg_x(g \bmod p) > \deg_x(g)$.
- α is unlucky $\Leftrightarrow \deg_x(\gcd(\check{f}_1(\alpha), \check{f}_2(\alpha))) > \deg_x(g)$.

■ Zero Divisors

- p is large $\Rightarrow \text{Prob}(\textit{Hitting a zero divisor})$ is small.

ModGcd Algorithm (contd.)

- Lemma: Let $f_1, f_2 \in L[x]$ where $L = F[z_1, \dots, z_r] / \langle m_1, \dots, m_r \rangle$ and $F = \mathbb{Q}(t_1, \dots, t_k)$. Let $\check{g} = \gcd(\check{f}_1, \check{f}_2)$. Let p be a prime and $\alpha = (t_1 = \alpha_1, \dots, t_k = \alpha_k) \in \mathbb{Z}_p^k$. Suppose that the Euclidean algorithm applied to $\check{f}_1(\alpha, x)$ and $\check{f}_2(\alpha, x)$ modulo p does not fail and outputs g_p . If α is not a bad evaluation point and p is not a bad prime, $\deg_x(g_p) \geq \deg_x(\check{g})$. Moreover if $\deg_x(g_p) = \deg_x(\check{g})$ then $g_p = \text{monic}(\gcd(\check{f}_1(\alpha, x), \check{f}_2(\alpha, x)) \bmod p)$.
- Using this we reconstruct g .

SparseModGcd Algorithm

- Motivation for sparse interpolation

$d_i = \deg_{t_i}(g) \Rightarrow$ Euclidean algorithm is called approximately $O(\prod_{i=1..k} d_i)$ times.

Example: Let $\check{g} = x_1^2 + t_1^{100}zx + t_2^{100}x + t_3^{100} \Rightarrow$ about 1000000 calls.

- Using sparse interpolation, we only need ~ 300 calls.
- Euclidean algorithm is the most time consuming part of the algorithm.

SparseModGcd Algorithm (contd.)

- **Example:** Let $g = x^2 + (10s - 16t)zx + 17x - 2st + 23$, $f_1 = (x + 1)g$, $f_2 = (x + z)g$ and $m(z) = z^2 - t$. Let $p_1 = 11$.
- Choose $\alpha = 2$ at random. Evaluate f_1 and f_2 at $s = \alpha$. Take random evaluation points $t = 1$ and $t = 5$.
- Compute the univariate gcds
$$g_1 = \gcd(f_1(x, 1, 2), f_2(x, 1, 2)) \bmod p_1 = x^2 + 4zx + 6x + 8$$
 and
$$g_2 = \gcd(f_1(x, 5, 2), f_2(x, 5, 2)) \bmod p_1 = x^2 + 6zx + 6x + 3.$$
- Using the dense interpolation we get
$$g_3 = \gcd(f_1(x, t, 2), f_2(x, t, 2)) \bmod p_1 = x^2 + (9 + 6t)zx + 6x + 7t + 1.$$

SparseModGcd Algorithm (contd.)

- Take the next evaluation point $s = 3$. From
 $g_3 \Rightarrow g(x, t, 3) \bmod p_1 = x^2 + (A + Bt)zx + Cx + (Dt + E)$.
- Choose evaluation points $t = -1$ and $t = 6$
 - $t = -1 \Rightarrow g(x, -1, 3) = x^2 + 2zx + 6x + 7 \Rightarrow \{C = 6, A - B = 2, -D + E = 7\} \bmod p_1$
 - $t = 6 \Rightarrow g(x, 6, 3) = x^2 + 6zx + 9 \Rightarrow \{C = 6, A + 6B = 0, 6D + E = 9\} \bmod p_1$
 - Solving this system of linear equations
 $\Rightarrow A = 8, B = 6, C = 6, E = 1, D = 5$.
 - $\Rightarrow g(x, t, 3) = x^2 + (8 + 6t)zx + 6x + 5t + 1$
- Using the same method, we can find more images if needed, to interpolate s .

SparseModGcd Algorithm (contd.)

- Interpolate s , apply RFR

$$\Rightarrow g \bmod p_1 = x^2 + (10s + 6t)zx + 6x + 9st + 1.$$

- For $p_2 = 17$ we have

$$g \bmod p_2 = x^2 + (As + Bt)zx + Cx + Dst + E, \dots$$

- After computing enough images: Apply CRT and RNR.

SparseModGcd Algorithm (contd.)

- Zippel's algorithm works when the gcd is monic in the main variable.
- LINZIP : Adding scaling factors
- g_f is the assumed form
 $\Rightarrow m_i g(\alpha_1, \dots, \alpha_n, x) \bmod p = g_f(\alpha_1, \dots, \alpha_n, x).$
- Each univariate image, introduces a new unknown m_i , hence we might need more images.
- The solution expense for the multiple scaling factor is still the same.

SparseModGcd Algorithm (contd.)

- Let T = the number of terms in the assumed form g_f .
- Let c_i be the coefficient of x^i in g_f and n_i be the number of terms in c_i .
- Let n_u be the maximum over all n_i 's.
- The number of univariate images needed is:
 - n_u when g is monic,
 - $\max(n_u, \left\lceil \frac{\sum_1^T n_i - 1}{T-1} \right\rceil)$ otherwise.
- Worst case: g is sparse ($T = 2$) and each coefficient has a lot of terms.

SparseModGcd Algorithm (contd.)

- Algorithm SparseModGcd
- Supports multiple field extensions
- Bad forms, missing terms : p introduces missing terms if $g \bmod p$ has less terms than g .
Example: $g = tx^2 + 26zx + t - 8 \Rightarrow p = 13, t = 0, 8$.
- Problem: Choosing an assumed form based on an image which is computed mod such prime.
- Can not detect them in advance.
- A good idea: No term in the inputs should vanish mod any of our primes.

SparseModGcd Algorithm (contd.)

- Prob (p introducing missing term) decreases for larger primes.
- Unlucky contents
- p introduces unlucky content if $\text{cont}_x(g) = 1$ but $\text{cont}_x(g \bmod p) \neq 1$. **Example:** $g = (12s + t)x + (s + 12t)$.
- Problem: Choosing an assumed form based on an image which is computed mod such prime.
- Such primes and evaluation points are rare so we do not detect them in advance.



Bottlenecks

- Univariate gcd computation: Euclidean algorithm
- Sparse Interpolation : polynomial evaluation and solving the system of equations
- LinearAlgebra :- Modular :- Mod and LinearAlgebra :- Modular :- RowReduce
- Trial Division is very slow.
- In order to increase the probability that the trial division succeeds we use Maximal Quotient Rational Reconstruction.

Benchmarks

Let

$$m(z) = z^3 - (s + r)z^2 - (t + v)^2z - 5 - 3u,$$

$$g = sx_1^n + tx_2^n + ux_3^n + \sum_{j=1}^4 \sum_{i=0}^{n-1} r_{ij}^{(1)} z^{j-1} x_j^i + \sum_{w=[r,s,t,u,v]} \sum_{k=0}^n r_{w_k}^{(1)} w^k,$$

$$a = tx_1^n + ux_2^n + sx_3^n + \sum_{j=1}^4 \sum_{i=0}^{n-1} r_{ij}^{(2)} z^{j-1} x_j^i + \sum_{w=[r,s,t,u,v]} \sum_{k=0}^n r_{w_k}^{(2)} w^k,$$

$$b = ux_1^n + sx_2^n + tx_3^n + \sum_{j=1}^4 \sum_{i=0}^{n-1} r_{ij}^{(3)} z^{j-1} x_j^i + \sum_{w=[r,s,t,u,v]} \sum_{k=0}^n r_{w_k}^{(3)} w^k.$$

Benchmarks (contd.)

n	SparseModGcd	ModGcd
1	0.40	8.70
2	1.29	114.78
3	2.40	879.26
4	4.46	> 2000
5	7.57	NA
6	12.51	NA
7	20.25	NA
8	29.73	NA
9	43.03	NA
10	61.87	NA

Benchmarks (contd.)

- Suppose g , a and b are three randomly chosen polynomials in variables x_1, x_2, s and z of total degree n which are dense.
- The term $x_1^{d_1} x_2^{d_2} s^{d_3} z^{d_4}$ with $d_1 + d_2 + d_3 + d_4 \leq n$ is present in each of these three polynomials.
- Each of them has exactly $\sum_{i=0}^n \binom{n+4}{4}$ number of terms.
- For $n = 1, 2, \dots, 10, 15$, let $f_1 = g \times a$ and $f_2 = g \times b$.
- Since in this set of problems the gcd g is dense, ModGcd algorithm is expected to perform better.

Benchmarks (contd.)

n	SparseModGcd	ModGcd
1	0.033	0.029
2	0.072	0.058
3	0.151	0.141
4	0.313	0.307
5	0.498	0.557
6	0.921	1.272
7	1.584	2.091
8	2.527	3.244
9	4.191	5.024
10	7.704	7.437
15	62.758	50.228



Summary

- Encarnacion's algorithm: polynomials over a number field
- ModGcd algorithm: polynomials over an algebraic function field with one field extension
- Design and implementation of SparseModGcd algorithm
- The first sparse algorithm for multiple field extensions
- SparseModGcd is very efficient for sparse polynomials and is competitive with ModGcd on dense problems.
- Future work: Improve the algorithm!



Questions?