# A New Sparse Polynomial GCD by Separating Terms

Michael Monagan

This is joint work with Qiao-Long Huang from Shandong Univerity.

Let $A, B \in \mathbb{Z}[x_1, \ldots, x_n]$.
Goal is to compute $G = \gcd(A, B)$ for $A, B, G$ sparse.
Main application; simplify $A/B$

Pick a new 63 bit prime $p$.
Compute $\gcd(A \mod p, B \mod p)$ in $\mathbb{F}_p[x_1, \ldots, x_n]$.
Repeat this and use Chinese remaindering to recover the integer coefficients of $G$.

**The Problem:** How to compute $\gcd(A, B)$ in $\mathbb{F}_p[x_1, \ldots, x_n]$ ?

Zippel 1979: sparse interpolation of $x_2, \ldots, x_n$, probabilistic
Trager and Kaltofen 1988: black-box
Diaz and Kaltofen 1995: black box
Hu and Monagan 2016: Kronecker substitution + Ben-Or/Tiwari

Gao and Huang ISSAC 2023: Let $A \in \mathbb{Z}[x_1, \ldots, x_n]$. Factor $A$

for $T = 2, 4, 8, 16, \ldots$ do

1. pick primes $p_1, p_2, \ldots, p_n$ randomly from $\mathbb{P}_{32}$
2. pick a single $s \in [0, T)^n$ at random
3. factor $A(x_i = p_i y^{s_i})$ in $\mathbb{Z}[y]$

Huang and Monagan ISSAC 2024: Let $A, B \in \mathbb{F}_p[x_1, \ldots, x_n]$. Compute $\gcd(A, B)$

for $T = 2, 4, 8, 16, \ldots$ do

1. pick $\gamma \in \mathbb{F}_p^n$ and $\alpha \in \mathbb{F}_p^n$ at random.
2. for $\log_2 T$ choices of $s \in [0, T)^n$ chosen at random compute $\gcd(A(x_i = (\gamma_i z - \alpha_i) y^{s_i}), B(x_i = (\gamma_i z - \alpha_i) y^{s_i}))$ in $\mathbb{F}_p[z, y]$.

$$G = x_1^2 + 3x_1x_2 + 2x_2^3 - x_2$$

Fix $\gamma = [1,1], \alpha = [2,5]$ and try $s = [2,3]$.

$$G((z-\alpha_1)y^{s_1}, (z-\alpha_2)y^{s_2}) = (z-2)^2y^4 + 3(z-2)(z-5)y^5 + 2(z-5)^3y^9 - (z-5)y^3.$$

To separate the terms of $G$ we need $T \in O(t^2)$ where $t = \#G$
Idea: use several much smaller choices for $s$

Try $s = [1,1]$. Let $\phi_1(f) = f((z-2)y, (z-5)y)$.

$$G_1 = \phi_1(G) = (z-2)(4z-17)y^2 + 2(z-5)^3y^3 - (z-5)y.$$
$$G_1^* = \mathit{COLLISION} + 2x_2^3 - x_2.$$

Try $s = [2,1]$. Let $\phi_2(f) = f((z-2)y^2, (z-5)y)$.

$$G_2 = \phi_2(G) = (z-2)^2y^4 + (z-5)(2z^2 - 17z + 44)y^3 - (z-5)y.$$
$$G_2^* = x_1^2 + \mathit{COLLISION} - x_2.$$

We are missing $+3x_1x_2$. Try another $s$?

In $\mathbb{F}_p[z,y]$ GCDs are unique up to a scalar in $\mathbb{F}_p$.

$$\begin{aligned} G_1 &= 2(z-5)^3y^3 + (z-2)(4z-17)y^2 - (z-5)y \\ &\sim 1(z-5)^3y^3 + \tfrac{1}{2}(z-2)(4z-17)y^2 - \tfrac{1}{2}(z-5)y \\ G_1^* &:= x_2^2 - \tfrac{1}{2}x_2 \\ G_2 &= 1(z-2)^2y^4 + (z-5)(2z^2-17z+44)y^3 - (z-5)y \\ G_2^* &:= x_1^2 - x_2 \end{aligned}$$

Notice the monomial $x_2$ is common to both $G_1^*$ and $G_2^*$. Compute instead (Key idea)

$H_2 := \frac{1}{2}G_2 - \phi_2(G_1^*) = \frac{1}{2}(z-2)^2y^4 + \frac{3}{2}(z-2)(z-5)y^3.$

$H_2^* = \frac{1}{2}x_1^2 + \frac{3}{2}x_1x_2.$

$G_1^* := G_1^* + H_2^* = x_2^2 - \frac{1}{2}x_2 + \frac{1}{2}x_1^2 + \frac{3}{2}x_1x_2.$

Stop since $H_2$ had no collisions and output $G_1^*$ which equals $G$ w.h.p.

## The Algorithm

Tries $T = 2, 4, 8, \ldots$ until $\log_2 T$ choices for $s$ are enough to recover all terms of $G$.

| $G_1^*$ | $G_2^*$ | $G_3^*$ | $G_4^*$ | $G_5^*$ | $G_6^*$ |
|---|---|---|---|---|---|
| 279 | 309 | 226 | 118 | 56 | 8 |

Table: For $\#G = 996$, $n = 9$, $\deg G = 30$ at step $T = 64$

Theorem. Let $A, B \in \mathbb{F}_p[x_1, \ldots, x_n]$ and $d = \max_{i=1}^{n} \max(\deg(A, x_i), \deg(B, x_i))$.
Our algorithm outputs $G = \gcd(A, B)$ with probability $\geq 11/12$ using

$$\tilde{O}(n T_{inp} d \log p \log^3 T_{out} + n^2 d^2 T_{out} \log p)$$

bit operations where $T_{inp} = \#A + \#B$, $T_{out} = \#G$.

# Implementation in Maple + C

Do not compute

$A_k := A(x_i = (\gamma_i z - \alpha_i) y^{s_i})$
$B_k := B(x_i = (\gamma_i z - \alpha_i) y^{s_i})$
$G_k := \gcd(A_k, B_k)$ in $\mathbb{F}_p[z, y]$.

in $\mathbb{F}_p[z, y]$. Instead, we compute

Let $D = \min(\deg A, \deg B) \geq \deg(G_k, z)$
Pick $\beta_0 \neq \beta_1 \neq \ldots \beta_D$ from $\mathbb{F}_p$ at random
For $k = 0, 1, \ldots, D$ do
    $g_k := \gcd(A(x_i = (\gamma_i \beta_k - \alpha_i) y^{s_i}), B(x_i = (\gamma_i \beta_k - \alpha_i) y^{s_i})$ in $\mathbb{F}_p[y]$.
Interpolate $z$ in $G_k(y, z)$.

**Optimization 1** To speed up evaluation of $A = \sum_{j=1}^{\#A} a_j M_j(x_1, \ldots, x_n)$ compute

$$m_{jk} = M_j(x_i = \gamma_i \beta_k - \alpha_i) \quad \text{for} \quad 1 \leq j \leq \#A \text{ and } 0 \leq k \leq D$$

and resuse the $m_{jk}$ for each choice of $s$.

# Benchmark 1

Let $A = G\bar{A}$, $B = G\bar{B}$ in $\mathbb{Z}[x_1, \ldots, x_n]$ where $G = \gcd(A, B)$.
$G$ has $t$ terms $\quad \bar{A}, \bar{B}$ have $s$ terms.
Monomials in $n = 9$ variables $\quad G, \bar{A}, \bar{B}$ have total degree 30.

| $s$ | $t$ | MGCD | $T$ | eval | Maple | Magma | MonHu |
|---|---|---|---|---|---|---|---|
| $10^5$ | $10^1$ | 11.69 | 4 | 10.22 | 78.92 | 39.90 | 0.661 |
| $10^4$ | $10^2$ | 13.55 | 8 | 10.24 | 197.6 | 9.98 | 1.488 |
| $10^3$ | $10^3$ | 29.65 | 64 | 10.27 | 1054.9 | 37.49 | 6.868 |
| $10^2$ | $10^4$ | 13.43 | 16 | 10.24 | 14568. | 27.68 | 1.087 |
| $10^1$ | $10^5$ | 10.67 | 4 | 9.47 | NA | 144.8 | 0.696 |

Table: Benchmark 1: Timings in CPU seconds for $n$=9

Maple and Magma are using Zippel's algorithm.
MGCD used two 63 bit primes.
Notice $T$ is small: $T < \min(s, t)$.

## Benchmark 2

Same as benchmark 1 except $n = 18$.
Here MonHu cannot use a 64 bit prime. It uses a 128 bit prime.

| $s$ | $t$ | MGCD | $T$ | eval | Maple | Magma | MonHu |
|---|---|---|---|---|---|---|---|
| $10^5$ | $10^1$ | 38.17 | 4 | 22.67 | 494.9 | 166.5 | 310.2 |
| $10^4$ | $10^2$ | 48.76 | 16 | 24.62 | 1473.2 | 79.50 | 450.8 |
| $10^3$ | $10^3$ | 92.60 | 128 | 24.68 | 14287. | 447.8 | 4358. |
| $10^2$ | $10^4$ | 50.54 | 16 | 24.64 | NA | 76.73 | 605.7 |
| $10^1$ | $10^5$ | 39.61 | 4 | 22.59 | NA | 188.1 | 150.6 |

Table: Benchmark 2 timings in CPU seconds for $n$=18

# Benchmark 3

Let $\Gamma = \gcd(LC(A, x_1), LC(B, x_1)$ and $\Delta = \Gamma / LC(\gcd(A, B))$.
Zippel and Monagan/Hu scale by images of $\Gamma$ so interpolate $\Delta G$ which is not good if $\#\Delta \gg 1$. For example

$$G = x + y, \bar{A} = (y^2 - 1)x + y + 1, \bar{B} = (y^3 - 1)x + y - 1$$

Here $\Delta = y - 1$ and $\Delta G = (y - 1)x + y^2 - 1$.
In benchmark 3 $\#\Delta = t/10$.

| $t$ | $\#A$ | MGCD | $T$ | Maple | Magma | MonHu | tmax |
|---|---|---|---|---|---|---|---|
| 50 | 22100 | 0.945 | 4 | 82.86 | 2.17 | 0.279 | 150 |
| 100 | 169096 | 6.359 | 8 | 2794.8 | 6.15 | 5.718 | 829 |
| 150 | 573732 | 21.36 | 16 | 25407. | 148.2 | 37.10 | 1848 |
| 200 | 1352967 | 46.57 | 16 | NA | 1058.9 | 136.8 | 3349 |
| 300 | 4538198 | 143.5 | 32 | NA | 13752. | 1079.1 | 7409 |
| 500 | 20849989 | 671.9 | 32 | NA | NA | 12400. | 20656 |

## Conclusion and Future Work

- We have a new GCD algorithm for $\mathbb{F}_p[x_1, \ldots, x_n]$ that has polynomial complexity in $T_{inp} = \#A + \#B$ and $T_{out} = \#G$.
- It works well in practice for all inputs and 32 bit primes are good.
- It's easy to code; it does not need any fast arithmetic for good performance.
- TODO: how big does $T$ need to be for the algorithm succeeds?
- TODO: parallelize $\gcd(A(x_i = (\gamma_i\beta_k - \alpha_i)y^{s_i}), B(x_i = (\gamma_i\beta_k - \alpha_i)y^{s_i}))$ on $k$.

# Thank you for coming.