A Fast Parallel Sparse Polynomial GCD Algorithm.

Jiaxiong Hu and Michael Monagan
Department of Mathematics
Simon Fraser University.

SFU

This work is supported by NSERC of Canada and Maplesoft

The GCD Problem

Input: A and B in $\mathbb{Z}[x_0, x_1, \ldots, x_n]$.
Output: $G = \gcd(A, B)$.

Talk: assume $G = 1x_0^m + \sum_{i=0}^{m-1} c_i(x_1, \ldots, x_n)x_0^i$

The GCD Problem

Input: A and B in $\mathbb{Z}[x_0, x_1, \ldots, x_n]$.
Output: $G = \gcd(A, B)$.

Talk: assume $G = 1x_0^m + \sum_{i=0}^{m-1} c_i(x_1, \ldots, x_n) x_0^i$

Step 1 Pick a prime p and points $\alpha_j \in \mathbb{Z}_p^n$ and compute

$$\gcd(A(x_0, \alpha_j), B(x_0, \alpha_j)) \bmod p = G(x_0, \alpha_j) = x_0^m + \sum_{i=0}^{m-1} \underbrace{c_i(\alpha_j)} x_0^i$$

$$\text{for } j = 1, 2, \ldots, T \quad \text{and} \quad \textit{interpolate } c_i(x_1, \ldots, x_n)$$

Step 2 Compute $\gcd(A, B)$ modulo $p_2, p_3, \ldots$ and obtain G using Chinese remaindering.

We do we parallelize for **N** cores?

Sparse Interpolation Algorithms

Assume $G = x_0^m + \sum_{i=0}^{m-1} c_i(x_1, \ldots, x_{\mathbf{n}}) x_0^i$ is **sparse**.

Let $\mathbf{t} = \max_i \# c_i$ and $\mathbf{d} = \max_i \deg_{x_i} G$ and $\mathbf{D} = \deg G$.

Zippel [1979]	$O(ndt)$ points	$p > 2nd^2t^2 = 6.4 \times 10^9$
BenOr/Tiwari [1988]	$O(t)$ points	$p > p_n^D = 5.3 \times 10^{77}$
Monagan/Javadi [2010]	$O(nt)$ points	$p > nDt^2 = 4.8 \times 10^8$
Discrete Logs	$O(t)$ points	$p > (d+1)^n = 3.8 \times 10^{10}$

Large GCD example: $n = 8$, $d = 20$, $D = 60$ and $t = 1000$.

Sparse Interpolation Algorithms

Assume $G = x_0^m + \sum_{i=0}^{m-1} c_i(x_1, \ldots, x_n) x_0^i$ is **sparse**.

Let $\mathbf{t} = \max_i \#c_i$ and $\mathbf{d} = \max_i \deg_{x_i} G$ and $\mathbf{D} = \deg G$.

Zippel [1979]	$O(ndt)$ points	$p > 2nd^2t^2 = 6.4 \times 10^9$
BenOr/Tiwari [1988]	$O(t)$ points	$p > p_n^D = 5.3 \times 10^{77}$
Monagan/Javadi [2010]	$O(nt)$ points	$p > nDt^2 = 4.8 \times 10^8$
Discrete Logs	$O(t)$ points	$p > (d+1)^n = 3.8 \times 10^{10}$

Large GCD example: $n = 8$, $d = 20$, $D = 60$ and $t = 1000$.

Talk Outline.

1. The BenOr-Tiwari algorithm and discrete logs
2. Unlucky evaluations and Kronecker substitutions.
3. Benchmarks (in Cilk C) and Current work.

Ben-Or Tiwari Sparse Interpolation

Let $C(x_1, \ldots, x_n) = \sum_{i=1}^{t} a_i M_i(x_1, \ldots, x_n)$ where $a_i \in \mathbb{Z}$.

Step 1 compute values $v_j = C(2^j, 3^j, 5^j, \ldots, p_n^j)$ for $0 \le j < 2t$.

Step 2 determine $m_i = M_i(2, 3, 5, \ldots, p_n)$ from v_j

Step 3 factor the integers m_i to determine the monomials M_i

Step 4 determine the coefficients a_i by solving

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ m_1 & m_2 & \ldots & m_t \\ {m_1}^2 & {m_2}^2 & \ldots & {m_t}^2 \\ \vdots & \vdots & \vdots & \vdots \\ {m_1}^{t-1} & {m_2}^{t-1} & \ldots & m_t^{t-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_t \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{t-1} \end{bmatrix}$$

Do this all mod a prime $p > m_i \le p_n^D = 19^{60} = 5.3 \times 10^{77}$.

Ben-Or/Tiwari using discrete logarithms in $\mathbb{Z}_p$

[Fujise and Murao. JSC 1996, PASCO 1994.]
[Kaltofen, unpublished 1988, PASCO 2010]

- Pick a prime $p = q_1 q_2 q_3 \dots q_n + 1$ with $\gcd(q_i, q_j) = 1$ and $q_i > \deg_{x_i} G \implies p > (d+1)^n = 21^8 = 3.8 \times 10^{10}$.
- Pick a random primitive element $\alpha \in \mathbb{Z}_p$ and set $\omega_i := \alpha^{(p-1)/q_i} \implies \omega_i^{q_i} = 1$.
- Replace $(2^j, 3^j, \dots, p_n^j)$ with $(\omega_1^j, \omega_2^j, \dots, \omega_n^j)$ in BT. Hence if $M_i = \prod_{k=1}^n x_k^{d_k}$ we have $m_i = \prod_{k=1}^n \omega_k^{d_k}$.

Step 3 Compute the discrete logarithm

$$\log_\alpha m_i = d_1 q_2 q_3 \dots q_n + \dots + d_n q_1 q_2 \dots q_{n-1}$$

using Pohlig-Hellman in $O(\sum_i \sqrt{q_i})$ and solve for the d_k.

Unlucky Evaluation Points

Let $G = \gcd(A, B)$ and $\bar{A} = A/G$ and $\bar{B} = B/G$.

Definition. A point $\alpha \in \mathbb{Z}_p^n$ is **unlucky** if $\gcd(\bar{A}(x_0, \alpha), \bar{B}(x_0, \alpha)) \neq 1$.
We can't interpolate G using unlucky evaluation points.

Example. $\bar{A} = x_0^2 + (x_1 - 1)(x_2 - 9)x_0 + 1$ Unlucky α?
$\bar{B} = x_0^2 + 1$

Unlucky Evaluation Points

Let $G = \gcd(A, B)$ and $\bar{A} = A/G$ and $\bar{B} = B/G$.

Definition. A point $\alpha \in \mathbb{Z}_p^n$ is **unlucky** if $\gcd(\bar{A}(x_0, \alpha), \bar{B}(x_0, \alpha)) \neq 1$.
We can't interpolate G using unlucky evaluation points.

Example. $\bar{A} = x_0^2 + (x_1 - 1)(x_2 - 9)x_0 + 1$, $\bar{B} = x_0^2 + 1$

Unlucky α?
$(1, \blacksquare)$ and $(\blacksquare, 9)$

Theorem: If α is chosen at random from $\mathbb{Z}_p^n$ then

$$\text{Prob}[\alpha \text{ is unlucky}] \leq \frac{\deg \bar{A} \deg \bar{B}}{p}.$$

We need $2t$ consecutive unlucky evaluation points for BT.

Ben-Or Tiwari Evaluation Points

Example. $\bar{A} = x_0^2 + (x_1 - 1)(x_2 - 9)x_0 + 1$
$\bar{B} = x_0^2 + 1$

Ben-Or/Tiwari $\alpha_j = (2^j, 3^j, 5^j, \ldots, p_n^j)$ for $0 \leq j < 2t$.
$j = 0, 2$ are unlucky.

Ben-Or Tiwari Evaluation Points

Example. $\bar{A} = x_0^2 + (x_1 - 1)(x_2 - 9)x_0 + 1$
$\bar{B} = x_0^2 + 1$

Ben-Or/Tiwari $\alpha_j = (2^j, 3^j, 5^j, \ldots, p_n^j)$ for $0 \leq j < 2t$.
$j = 0, 2$ are unlucky.

Pick s with $2^s > p$ and use $s \leq j < 2t + s$.
Must solve the shifted transposed Vandermonde system

$$\begin{bmatrix} m_1^s & m_2^s & \ldots & m_t^s \\ {m_1}^{s+1} & {m_2}^{s+1} & \ldots & {m_t}^{s+1} \\ \vdots & \vdots & \vdots & \vdots \\ {m_1}^{s+t-1} & {m_2}^{s+t-1} & \ldots & m_t^{s+t-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_t \end{bmatrix} = \begin{bmatrix} v_s \\ v_{s+1} \\ \vdots \\ v_{s+t-1} \end{bmatrix}$$

Additional cost is $O(t \log s)$ multiplications (details in paper).

Discrete Logs Evaluation Points

Example. $\bar{A} = x_0^2 + (x_1 - 1)(x_2 - 9)x_0 + 1$
$\bar{B} = x_0^2 + 1$

Discrete logs uses $\alpha_j = (\omega_1^j, \omega_2^j, \ldots, \omega_n^j)$ for $1 \leq j \leq 2t$.
But $\omega_i^{q_i} = 1$ so $j = q_1, 2q_1, 3q_1, \ldots$ are unlucky.

Pick $q_i > 2t \implies p > (2t)^n = (2000)^8 = 2.5 \times 10^{27}$.
But we don't know t!

Kronecker Substitutions

For $r > 0$ define

$$K_r(G(x_0, x_1, \ldots, x_n)) = G(x, y, y^r, y^{r^2}, \ldots, y^{r^{n-1}}).$$

If $d = \deg G$ then K_r is invertible if $r > d$.

Example: GCD in $\mathbb{Z}_p[x_0, x_1, x_2]$.

$G = x_0^2 + x_1^2 + x_2^2$ $\quad$ $K_3(G) = x^2 + y^2 + y^6$

$\bar{A} = x_0^2 - x_1^2$ $\quad$ $K_3(\bar{A}) = x^2 - y^2$

$\bar{B} = x_0^4 - x_1 x_2$ $\quad$ $K_3(\bar{B}) = x^4 - y^4$

$\gcd(\bar{A}, \bar{B}) = 1$ $\quad$ $\gcd(K_3(\bar{A}), K_3(\bar{B})) = x^2 - y^2$

Kronecker Substitutions

For $r > 0$ define

$$K_r(G(x_0, x_1, \ldots, x_n)) = G(x, y, y^r, y^{r^2}, \ldots, y^{r^{n-1}}).$$

If $d = \deg G$ then K_r is invertible if $r > d$.

Example: GCD in $\mathbb{Z}_p[x_0, x_1, x_2]$.

$$\begin{array}{ll} G = x_0^2 + x_1^2 + x_2^2 & K_3(G) = x^2 + y^2 + y^6 \\ \bar{A} = x_0^2 - x_1^2 & K_3(\bar{A}) = x^2 - y^2 \\ \bar{B} = x_0^4 - x_1 x_2 & K_3(\bar{B}) = x^4 - y^4 \\ \gcd(\bar{A}, \bar{B}) = 1 & \gcd(K_3(\bar{A}), K_3(\bar{B})) = x^2 - y^2 \end{array}$$

Definition: K_r is unlucky if $\gcd(K_r(\bar{A}), K_r(\bar{B})) \neq 1$

Theorem 1: The number of unlucky K_r is $\leq (n-1)\sqrt{2 \deg \bar{A} \deg \bar{B}}$.

Try K_r for $r = d+1, d+2, \ldots$ until we get a lucky one.

Kronecker substitutions and unlucky evaluation points

Example

$$G = x_0 + x_1^d + x_2^d + \cdots + x_n^d$$
$$\bar{A} = x_0 + x_1 + \cdots + x_{n-1} + x_n^{d+1}$$
$$\bar{B} = x_0 + x_1 + \cdots + x_{n-1} + 1$$
$$R = res_{x_0}(\bar{A}, \bar{B}) = 1 - x_n^{d+1} \text{ and } K_{d+1}(R) = 1 - y^{(d+1)^n}$$

$$\text{Prob}[\alpha \text{ is unlucky}] \leq \frac{\deg K(R)}{p} \leq \frac{(d+1)^n}{p}.$$

Kronecker substitutions and unlucky evaluation points

Example

$$G = x_0 + x_1^d + x_2^d + \cdots + x_n^d$$
$$\bar{A} = x_0 + x_1 + \cdots + x_{n-1} + x_n^{d+1}$$
$$\bar{B} = x_0 + x_1 + \cdots + x_{n-1} + 1$$
$$R = res_{x_0}(\bar{A}, \bar{B}) = 1 - x_n^{d+1} \text{ and } K_{d+1}(R) = 1 - y^{(d+1)^n}$$

$$\text{Prob}[\alpha \text{ is unlucky}] \leq \frac{\deg K(R)}{p} \leq \frac{(d+1)^n}{p}.$$

Theorem 2

Over $\mathbb{F}_p$ let $A = x^m + \sum_{i=0}^{m-1} a_i(y)x^i$, and $B = x^n + \sum_{i=0}^{n-1} b_i(y)x^i$.
Let $X = |\{0 \leq \beta < p : \gcd(A(x,\beta), B(x,\beta)) \neq 1\}|$.
If $m > 0$ and $n > 0$ and $\deg a_i(y), b_i(y) \leq d$ then

$$E[X] =$$

Kronecker substitutions and unlucky evaluation points

Example

$$G = x_0 + x_1^d + x_2^d + \cdots + x_n^d$$
$$\bar{A} = x_0 + x_1 + \cdots + x_{n-1} + x_n^{d+1}$$
$$\bar{B} = x_0 + x_1 + \cdots + x_{n-1} + 1$$
$$R = res_{x_0}(\bar{A}, \bar{B}) = 1 - x_n^{d+1} \text{ and } K_{d+1}(R) = 1 - y^{(d+1)^n}$$

$$\text{Prob}[\alpha \text{ is unlucky}] \le \frac{\deg K(R)}{p} \le \frac{(d+1)^n}{p}.$$

Theorem 2

Over $\mathbb{F}_p$ let $A = x^m + \sum_{i=0}^{m-1} a_i(y)x^i$, and $B = x^n + \sum_{i=0}^{n-1} b_i(y)x^i$.
Let $X = |\{0 \le \beta < p : \gcd(A(x,\beta), B(x,\beta)) \ne 1\}|$.
If $m > 0$ and $n > 0$ and $\deg a_i(y), b_i(y) \le d$ then

$$E[X] = 1 \implies \text{Prob}[\alpha \text{ is unlucky}] = \frac{1}{p}.$$

Try $p > 2(d+1)^n$. If unlucky evaluations occur increase p.

Benchmark

New algorithm coded in Cilk C codes for 31, 63 and 127 bit primes.
Benchmark: $n = 8$, $d = 20 \geq \deg_{x_i} G, \bar{A}, \bar{B}$, $D = 60 \geq \deg G, \bar{A}, \bar{B}$.
Coefficients of $G, \bar{A}, \bar{B}$ generated at random on $[0, 2^{31})$.

			New algorithm $p = 29 \cdot 2^{57} + 1$		Zippel's algorithm	
#G	#A	t	1 core (eval)	16 cores	Maple	Magma
10^3	10^5	113	0.66s (68%)	0.100s (6.6x)	341.9s	63.55s
10^3	10^6	130	5.66s (90%)	0.717s (9.4x)	5553.5s	FAIL
10^4	10^6	1198	48.44s (87%)	4.474s (10.2x)	62520.1s	FAIL
10^3	10^7	122	52.102 (92%)	4.591s (11.3x)	NA	NA
10^4	10^7	1212	428.96s (98%)	37.43s (11.5x)	NA	NA
10^5	10^7	11867	3705.4s (98%)	311.60s (11.9x)	NA	NA
10^6	10^7	117508	47568.0s (90%)	3835.9s (12.4x)	NA	NA

Timings (in seconds) on two Xeon E5-2680 CPUs, 8 cores, 2.2GHz/3.0GHz.
Maximum parallel speedup $= 16 \times 2.2/3.0 = 11.7$ x.

Evaluation: If $G = \gcd(A, B)$ usually $(\mathbf{s} = \#A + \#B) \gg \#G \gg t$.

Improvements

- Evaluation: $O(sn + nd + st) \longrightarrow O(sn + nd + s\log^2 t)$ ops in $\mathbb{Z}_p$.
- Bivariate Images: Let
  $K_r(A(x_0, x_1, x_2, \ldots, x_n) = A(x, y, z, z^r, z^{r^2}, \ldots)$.
  Interpolate $K_r(G)$ from

$$\gcd(K_r(A)(x, y, z = \alpha^j), K_r(B)(x, y, z = \alpha^j)).$$

  Gain? $t : 1198 \longrightarrow 122$. Cost? $O(40^2) \rightarrow O(40^3)$.
  Time (1 core): $48.44s \rightarrow 7.27s$ Time (16 cores): $4.47s \rightarrow 0.66s$.

Final Remarks

- Algorithm: Input (A, B). Output $(p, G = \gcd(A, B) \mod p)$ w.h.p.
- The paper treats the general case G not monic.
- ?∃ enough smooth primes to find one which is not unlucky?

Current Work: Bivariate Images

Let $G = x_0^m + \sum_{i=0}^{m-1} \sum_{j=0} c_{ij}(x_2, \ldots, x_n) x_0^i x_1^j$ in $\mathbb{Z}[x_2, \ldots, x_n][x_0, x_1]$.
Gain? reduces t.
Cost? $O(d^2) \rightarrow O(d^3)$ per image using Brown's dense GCD algorithm.

Current Work: Bivariate Images

Let $G = x_0^m + \sum_{i=0}^{m-1} \sum_{j=0} c_{ij}(x_2, \ldots, x_n) x_0^i x_1^j$ in $\mathbb{Z}[x_2, \ldots, x_n][x_0, x_1]$.
Gain? reduces t.
Cost? $O(d^2) \to O(d^3)$ per image using Brown's dense GCD algorithm.

			New algorithm $p = 29 \cdot 2^{57} + 1$		Zippel's algorithm	
$\#G$	$\#A$	t	1 core (eval)	16 cores	Maple	Magma
10^3	10^5	113	0.66s (68%)	0.100s (6.6x)	341.9s	63.55s
		13	0.31s (55%)	0.066s (4.5x)		
10^3	10^6	130	5.66s (90%)	0.717s (9.4x)	5553.5s	FAIL
		14	1.68s (68%)	0.268 (4.3x)		
10^4	10^6	1198	48.44s (87%)	4.474s (10.2x)	62520.1s	FAIL
		122	7.27s (74%)	0.656s (11.2x)		
10^4	10^7	1212	428.96s (98%)	37.43s (11.5x)	NA	NA
		122	57.21s (90%)	5.10s (11.2x)		
10^5	10^7	11867	3705.4s (98%)	311.60s (11.9x)	NA	NA
		1114	438.87s(90%)	34.40s (12.7x)		
10^6	10^7	117508	47568s (90%)	3835.9s (12.4x)	NA	NA
		11002	4794.5s (83%)	346.1s (13.8x)		

Kronecker substitutions + discrete logarithms

Before interpolate c_i in $G = x_0^m + \sum_{i=0}^{m-1} c_i(x_1, \ldots, x_n)x_0^i$.
Now interpolate y in $K_r(G) = x_0^m + \sum_{i=0}^{m-1} K_r(c_i)(y)x_0^i$.

Pick a smooth prime p with $p > r^n$.
Pick a random generator α from $\mathbb{Z}_p$. Interpolate $K_r(G)$ from

$$\gcd(K_r(A)(x, y = \alpha^j), K_r(B)(x, y = \alpha^j)) \quad \text{for } j = 1, 2, \ldots, T.$$