# Fast parallel multi-point evaluation of sparse polynomials.

Michael Monagan
Department of Mathematics
Simon Fraser University.

Milestones in Computer Algebra
Celebrating the research contributions of Erich Kaltofen
Waterloo, Ontario, July 18, 2016.

This is joint work with Alan Wong.

# My Connection with Erich

First Maple retreat, Sparrow lake, July 1983

# My Connection with Erich

PhD defense, October 1989.

Signatures + Abstract Types = Computer Algebra − Intermediate Expression Swell.

# My Connection with Erich

PhD defense, October 1989.

Signatures + Abstract Types = Computer Algebra − Intermediate Expression Swell.

Erich's contributions to ISSAC

General chair 1992, 2001    Program chair 2009

# My Connection with Erich

PhD defense, October 1989.
Signatures + Abstract Types = Computer Algebra − Intermediate Expression Swell.

Erich's contributions to ISSAC
General chair 1992, 2001     Program chair 2009

DBLP – Erich has 81 conference papers
Of these 42 + 2(2016) are ISSAC papers

# Motivation

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

Assume: $G = \sum_{i=0}^{m-1} \sum_{j=0} c_{ij}(x_2, \ldots, x_n) x_0^i x_1^j + x_0^n$ for talk.
Let $d = \max_{i=2}^{n} \deg_{x_i} G$ and $D = \max_{i=2}^{n} \deg_{x_i} A, \deg_{x_i} B$.

# Motivation

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

Assume: $G = \sum_{i=0}^{m-1} \sum_{j=0} c_{ij}(x_2, \ldots, x_n) x_0^i x_1^j + x_0^n$ for talk.
Let $d = \max_{i=2}^{n} \deg_{x_i} G$ and $D = \max_{i=2}^{n} \deg_{x_i} A, \deg_{x_i} B$.

Apply a Kronecker substitution to $A$ and $B$
$K_r(A(x_0, x_1, \ldots, x_n)) = A(x, y, z, z^r, \ldots, z^{r^{n-2}})$ for $r > d$.

Interpolate $z$ in $K(c_{ij})(z)$ using sparse interpolation modulo a smooth prime $p > r^{n-1}$ from $T$ bivariate images

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

We need $T = 2\,\mathbf{t} + \delta$ points where $\mathbf{t} = \max \# c_{ij}$ for BenOr-Tiwari.

What can we **parallelize** for **N** cores?

# Motivation

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

Assume: $G = \sum_{i=0}^{m-1} \sum_{j=0} c_{ij}(x_2, \ldots, x_n) x_0^i x_1^j + x_0^n$ for talk.
Let $d = \max_{i=2}^{n} \deg_{x_i} G$ and $D = \max_{i=2}^{n} \deg_{x_i} A, \deg_{x_i} B$.

Apply a Kronecker substitution to $A$ and $B$
$K_r(A(x_0, x_1, \ldots, x_n)) = A(x, y, z, z^r, \ldots, z^{r^{n-2}})$ for $r > d$.

Interpolate $z$ in $K(c_{ij})(z)$ using sparse interpolation modulo a smooth prime $p > r^{n-1}$ from $T$ bivariate images

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

We need $T = 2\,\mathbf{t} + \delta$ points where $\mathbf{t} = \max \# c_{ij}$ for BenOr-Tiwari.

What can we **parallelize** for **N** cores?

Observation: $S = \#A + \#B \gg \#G \gg t$, that is, $S \gg T$.
**Evaluation:** $\leq (n-1)D + (n-1)S + ST$ multiplications in $\mathbb{Z}_p$.
$\longrightarrow (n-1)D + (n-1)S + O(S \log^2 T)$.

A. Bostan, G. Lecerf, E. Schost, Telegens principle into practice, ISSAC 2003.

# Fast sparse evaluation

Let $f(z) = \sum_{i=1}^{s} a_i z^{d_i}$ be the sparse input.

Compute $y_j = f(\alpha^j) \mod p$ for $j = 0, 1, \ldots, T-1$ ?

Do this for $T \in \{2, 4, 8, 16, \ldots\}$ because we don't know $t$.

# Fast sparse evaluation

Let $f(z) = \sum_{i=1}^{s} a_i z^{d_i}$ be the sparse input.
Compute $y_j = f(\alpha^j) \mod p$ for $j = 0, 1, \ldots, T-1$ ?
Do this for $T \in \{2, 4, 8, 16, \ldots\}$ because we don't know $t$.

First compute $b_i := \alpha^{d_i}$ . Observe that

$$F(u) = \sum_{i=1}^{s} \frac{a_i}{1 - b_i u} = y_0 + y_1 u + y_2 u^2 + \cdots + y_{T-1} u^{T-1} + \ldots$$

# Fast sparse evaluation

Let $f(z) = \sum_{i=1}^{S} a_i z^{d_i}$ be the sparse input.
Compute $y_j = f(\alpha^j) \mod p$ for $j = 0, 1, \ldots, T-1$ ?
Do this for $T \in \{2, 4, 8, 16, \ldots\}$ because we don't know $t$.

First compute $b_i := \alpha^{d_i}$ . Observe that

$$F(u) = \sum_{i=1}^{S} \frac{a_i}{1 - b_i u} = y_0 + y_1 u + y_2 u^2 + \cdots + y_{T-1} u^{T-1} + \ldots$$

For $S > T$ split $F$ up into $K = \lceil \frac{S}{T} \rceil$ blocks of size $\leq T$.

$$F(u) \quad = \quad B_1(u) + B_2(u) + \cdots + B_K(u)$$

Then expand $B_i(u)$ to $O(u^T)$ and add up.

$$
\begin{aligned}
B_1 &= \frac{a_1}{1-b_1 u} + \frac{a_2}{1-b_2 u} + \cdots + \frac{a_T}{1-b_T u} \\
&= \underbrace{\sum_{i=1}^{\frac{T}{2}} \frac{a_i}{(1-b_i u)}}_{\text{compute recursively}} + \underbrace{\sum_{i=\frac{T}{2}+1}^{T} \frac{a_i}{(1-b_i u)}}_{\text{compute recursively}} \\
&= \frac{N_1}{D_1} + \frac{N_2}{D_2} = \underbrace{\frac{N_1 D_2 + N_2 D_1}{D_1 D_2}}_{\text{use fast multiplication}} = \frac{N}{D} \\
&= \underbrace{N(u) \cdot D(u)^{-1} = y_0 + y_1 u + \cdots + y_{T-1} u^{T-1} + O(y^T)}_{\text{use fast inversion}}
\end{aligned}
$$

$$
\begin{aligned}
B_1 &= \frac{a_1}{1-b_1 u} + \frac{a_2}{1-b_2 u} + \cdots + \frac{a_T}{1-b_T u} \\
&= \underbrace{\sum_{i=1}^{\frac{T}{2}} \frac{a_i}{(1-b_i u)}}_{\text{compute recursively}} + \underbrace{\sum_{i=\frac{T}{2}+1}^{T} \frac{a_i}{(1-b_i u)}}_{\text{compute recursively}} \\
&= \frac{N_1}{D_1} + \frac{N_2}{D_2} = \underbrace{\frac{N_1 D_2 + N_2 D_1}{D_1 D_2}}_{\text{use fast multiplication}} = \frac{N}{D} \\
&= \underbrace{N(u) \cdot D(u)^{-1} = y_0 + y_1 u + \cdots + y_{T-1} u^{T-1} + O(y^T)}_{\text{use fast inversion}}
\end{aligned}
$$

Let $C_1(T)$ be the cost of computing $N(u)/D(u)$ to $O(u^T)$. Then

$$
C_1(T) \leq 3M(T/2) + 2M(T) + 1M(T) \in O(M(T))
$$

where $M(T)$ is the cost of multiplying two polynomials of degree $\leq T$.

Evaluate $A(x, y, z = \alpha^j) \bmod p$ for $j = 0, 1, \ldots, T-1$.

$A = (3z^6)x^2y + (z^{13} + 8z^2 + 14z^{14} + 12)x^3 + (5z^7 + z^4 + 11z)xy$

$p = 17$, $\alpha = 3$, $S = \#A = 8$

| | $x^2y$ | $x^3$ | | | | $xy$ | | |
|---|---|---|---|---|---|---|---|---|
| | $3y^6$ | $y^{13}$ | $8y^2$ | $14y^{14}$ | $12$ | $5y^7$ | $y^4$ | $11y$ |
| $T=1$ | $\frac{3}{1-\alpha^6 u}$ <br> $3 + O(u)$ | $\frac{1}{1-\alpha^{13} u}$ <br> $1 + O(u)$ | + $\frac{8}{1-\alpha^2 u}$ <br> $8 + O(u)$ | + $\frac{14}{1-\alpha^{14} u}$ <br> $14 + O(u)$ | + $\frac{12}{1-u}$ <br> $12 + O(u)$ | $\frac{5}{1-\alpha^7 u}$ <br> $5 + O(u)$ | + $\frac{1}{1-\alpha^4 u}$ <br> $1 + O(u)$ | + $\frac{11}{1-\alpha}$ <br> $11 + O(u)$ |
| $T=2$ | $3 + 11u + O(u^2)$ | $\frac{14u+9}{6u^2+13u+1}$ <br> $9 + 16u + O(u^2)$ | | + $\frac{13u+9}{2u^2+14u+1}$ <br> $9 + 6u + O(u^2)$ | | $\frac{9u+6}{7u^2+10u+1}$ <br> $6 + 0u + O(u^2)$ | | + $11 + 16u + O(u^2)$ |
| $T=4$ | $3 + 11u + 12u^2$ $+10u^3 + O(u^4)$ | $\frac{4u^3+12u^2+15u+1}{12u^4+8u^3+3u^2+10u+1}$ <br> $1 + 5u + 10u^2 + 0u^3 + O(u^4)$ | | | | $\frac{16u^2+16u}{13u^3+11u^2+7u+1}$ <br> $0 + 16u + 6u^2 + 3u^3 + O(u^4)$ | | |

Cost at each level is $O(\frac{S}{T}M(T)) = O(S(1 + \log_2 T))$ with FFT in $\mathbb{Z}_p$

parallelize each level for $N$ cores?

# Parallel implementation in Cilk C for 63 bit primes.

Random sparse inputs in 9 variables with $S$ terms.
Degree at most $d = 10$ in each variable, total degree $\leq 60$.
Compute $T$ evaluations modulo $p = 29 \cdot 2^{57} + 1$.

| | | Matrix $O(nd + nS + ST)$ | | | Fast $O(nd + nS + S\log^2 T)$ | | |
|---|---|---|---|---|---|---|---|
| $S$ | $T$ | 1 core | 16 cores | speedup | 1 core | 16 cores | speedup |
| $10^7$ | $10^2$ | 7.35 | 0.73 | 10.0x | 11.18 | 1.45 | 7.7x |
| $10^7$ | $10^3$ | 64.32 | 5.29 | 12.2x | 38.94 | 3.63 | 10.7x |
| $10^7$ | $10^4$ | 633.51 | 51.43 | 12.3x | 92.25 | 7.77 | 11.9x |
| $10^7$ | $10^5$ | 6335.26 | 516.44 | 12.3x | 155.58 | 12.72 | 12.2x |
| $10^7$ | 500 | 32.67 | 2.71 | 12.0x | 27.83 | 2.77 | 10.1x |
| $10^8$ | $10^4$ | 6198.68 | 553.84 | 11.2x | 890.20 | 74.48 | 12.0x |
| $10^8$ | $10^5$ | - | 5852.47 | - | 1374.74 | 112.52 | 12.2x |
| $10^8$ | $10^6$ | - | - | - | 2045.96 | 164.96 | 12.4x |

Timings in CPU 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.

Why 63 bit primes and not 64 bit primes?

```
long sub(a,b,p) { long t = a-b; t += (t>>63) & p; return t; }
```

# Does this help speed up our polynomial GCD computations?

Random sparse inputs in 9 variables, monic in $x_0$.
Degree at most 20 in each variable, total degree $\leq 60$.
Compute $\gcd(A, B) \mod p = 29 \cdot 2^{57} + 1$ uses $T$ evaluations.

| $\#A$ | $\#G$ | $T$ | Fast (eval) | Matrix (eval) | Maple | Magma |
|---|---|---|---|---|---|---|
| $10^5$ | $10^3$ | 36 | 0.1 (76%) | 0.1 (55%) | 341.9 | 63.6 |
| $10^6$ | $10^3$ | 40 | 0.5 (88%) | 0.2 (66%) | 5553.5 | FAIL |
| $10^6$ | $10^4$ | 264 | 0.8 (82%) | 0.6 (74%) | 62520.1 | FAIL |
| $10^7$ | $10^4$ | 256 | 5.8 (90%) | 4.5 (88%) | - | - |
| $10^7$ | $10^5$ | 2334 | 13.5 (77%) | 36.1 (91%) | - | - |
| $10^7$ | $10^6$ | 24214 | 91.1 (32%) | 395.7 (85%) | - | - |
| $10^8$ | $10^5$ | 2328 | 96.3 (92%) | 369.2 (98%) | - | - |
| $10^8$ | $10^6$ | 24214 | 214.9 (69%) | 3691.1 (98%) | - | - |
| $10^8$ | $10^7$ | 242574 | 3058.1 (11%) | 39643.0 (93%) | - | - |

Time for GCD($A$, $B$) modulo a smooth prime $p$ on 16 cores.

Happy birthday Erich