A new random polynomial time algorithm for factoring sparse multivariate polynomials.

Michael Monagan

Department of Mathematics
Simon Fraser University
British Columbia

This is joint work with Baris Tuncer.

Problem: Let $a \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$.
Factor a into irreducible factors over $\mathbb{Z}$.
Example:

$$x_1{}^6 - 27\,x_2{}^3 = \left(x_1{}^2 - 3\,x_2\right)\left(x_1{}^4 + 3\,x_1{}^2 x_2 + 9\,x_2{}^2\right)$$

For the talk assume

(i) a has two factors f and g and (ii) a is monic in x_1.

Problem: Let $a \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$.
Factor a into irreducible factors over $\mathbb{Z}$.
Example:

$$x_1^{\,6} - 27\,x_2^{\,3} = \left(x_1^{\,2} - 3\,x_2\right)\left(x_1^{\,4} + 3\,x_1^{\,2}x_2 + 9\,x_2^{\,2}\right)$$

For the talk assume

(i) a has two factors f and g and (ii) a is monic in x_1.

Talk Outline:

- Wang's multivariate Hensel lifting.
- Wang's solution to the MDP $f_k g_0 + g_k f_0 = c_k$ in $\mathbb{Z}_p[x_1, \ldots, x_{j-1}]$.
- Our random polynomial time solution.
- A probabilistic analysis.
- Two benchmarks.
- Can we parallelize it?

Factoring polynomials using Wang's Hensel lifting

```
> e1 := (a = f*g);
```

$$e1 := x^5 + 20\,x^2y^6z + 5\,x^4y^3z + 30\,xy^4z^3 + 12\,xy^4z^2 + 4\,x^3y^3$$

$$+3\,x^3yz^2 + 18\,y^2z^4 + 6\,x^2yz^2$$

$$= \left(x^2 + 5\,xy^3z + 3\,yz^2\right)\left(x^3 + 4\,xy^3 + 6\,yz^2\right)$$

```
> e2 := eval(e1,z=3);
```

$$e2 := x^5 + 60\,x^2y^6 + 15\,x^4y^3 + 4\,x^3y^3 + 918\,xy^4 + 27\,x^3y + 54\,x^2y + 1458\,y^2$$

$$= \left(x^2 + 15\,xy^3 + 27\,y\right)\left(x^3 + 4\,xy^3 + 54\,y\right)$$

```
> e3 := eval(e2,y=-5);
```

$$e3 := x^5 - 1875\,x^4 - 635\,x^3 + 937230\,x^2 + 573750\,x + 36450$$

$$= \left(x^2 - 1875\,x - 135\right)\left(x^3 - 500\,x - 270\right)$$

Notes: Let h be any factor of a and let $B > \max(||h||_\infty, ||a||_\infty)$.
Multivariate Hensel Lifting (MHL) is done modulo a prime $p > 2B$.

Not all evaluation points can be used

$$f = \left(x^2 + 5\,xy^3z + 3\,yz^2\right)\left(x^3 + 4\,xy^3 + 6\,yz^2\right) + (y - z)$$

The polynomial f in $\mathbb{Z}[x, y, z]$ is irreducible over $\mathbb{Q}$ but

```
> eval(f,[z=3,y=3]);
```

$$(x^2 + 405x + 81)(x^3 + 108x + 162)$$

Theorem (Hilbert irreducibility)

If f is irreducible over $\mathbb{Q}$ and α, β are chosen from a sufficiently large set $S \subset \mathbb{Z}$ then $f(x, z = \alpha, y = \beta)$ is irreducible in $\mathbb{Q}$ with high probability.

Wang's Multivariate Hensel Lifting (MHL) : j'th step

Input $a \in \mathbb{Z}_p[x_1, ..., x_j]$, $\alpha = (\alpha_2, \ldots \alpha_j)$, $f_0, g_0 \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$ s.t.
(i) $a(x_1, ..., x_{j-1}, \alpha_j) = f_0 g_0$ and
(ii) $\gcd(f_0(x_1, \alpha), g_0(x_1, \alpha)) = 1$ in $\mathbb{Z}_p[x_1]$.

Idea: $f(x_j) = f_0 + f_1(x_j - \alpha_j) + f_2(x_j - \alpha_j)^2 + \ldots$ where $f_k = f^{(k)}(\alpha_j)/k!$

Wang's Multivariate Hensel Lifting (MHL) : j'th step

Input $a \in \mathbb{Z}_p[x_1, ..., x_j]$, $\alpha = (\alpha_2, \ldots \alpha_j)$, $f_0, g_0 \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$ s.t.
(i) $a(x_1, ..., x_{j-1}, \alpha_j) = f_0 g_0$ and
(ii) $\gcd(f_0(x_1, \alpha), g_0(x_1, \alpha)) = 1$ in $\mathbb{Z}_p[x_1]$.

Idea: $f(x_j) = f_0 + f_1(x_j - \alpha_j) + f_2(x_j - \alpha_j)^2 + \ldots$ where $f_k = f^{(k)}(\alpha_j)/k!$

Initialize: $f \leftarrow f_0; g \leftarrow g_0$ and $error := a - fg$
For $k = 1, 2, \ldots,$ while $\deg(f, x_j) + \deg(g, x_j) < \deg(a, x_j)$ do
  $c_k :=$ coeff($error$, $(x_j - \alpha_j)^k$)
  If $c_k \neq 0$ then
    Solve the MDP $f_k g_0 + g_k f_0 = c_k$ for $f_k, g_k \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$.
    Set $f \leftarrow f + f_k(x_j - \alpha_j)^k$ and $g \leftarrow g + g_k(x_j - \alpha_j)^k$.
    Set $error := a - fg$
If $error = 0$ output (f, g) else output FAIL.

Implemented in Magma, Maple, Macsyma, Mathematica and Singular $\Longrightarrow$ Sage.
Ref: Ch. 6 of **Algorithms for Computer Algebra**, Geddes, Czapor, and Labahn, 1992.

Wang's Multivariate Hensel Lifting (MHL) : j'th step

Input $a \in \mathbb{Z}_p[x_1, ..., x_j]$, $\alpha = (\alpha_2, \ldots \alpha_j)$, $f_0, g_0 \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$ s.t.
(i) $a(x_1, ..., x_{j-1}, \alpha_j) = f_0 g_0$ and
(ii) $\gcd(f_0(x_1, \alpha), g_0(x_1, \alpha)) = 1$ in $\mathbb{Z}_p[x_1]$.

Idea: $f(x_j) = f_0 + f_1(x_j - \alpha_j) + f_2(x_j - \alpha_j)^2 + \ldots$ where $f_k = f^{(k)}(\alpha_j)/k!$

Initialize: $f \leftarrow f_0; g \leftarrow g_0$ and $error := a - fg$
For $k = 1, 2, \ldots,$ while $\deg(f, x_j) + \deg(g, x_j) < \deg(a, x_j)$ do
  $c_k :=$ coeff($error$, $(x_j - \alpha_j)^k$)
  If $c_k \neq 0$ then
    Solve the MDP $f_k g_0 + g_k f_0 = c_k$ for $f_k, g_k \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$.
    Set $f \leftarrow f + f_k(x_j - \alpha_j)^k$ and $g \leftarrow g + g_k(x_j - \alpha_j)^k$.
    Set $error := a - fg$
If $error = 0$ output (f, g) else output FAIL.

Implemented in Magma, Maple, Macsyma, Mathematica and Singular $\Longrightarrow$ Sage.
Ref: Ch. 6 of **Algorithms for Computer Algebra**, Geddes, Czapor, and Labahn, 1992.
I noticed that often over 90% of the time was solving MDPs.

Wang's Multivariate Diophantine Solver

Input $A, B, C \in \mathbb{Z}_p[x_1, \ldots, x_n]$ and $\alpha = (\alpha_2, \ldots \alpha_n)$
Output $\sigma, \tau \in \mathbb{Z}_p[x_1, \ldots, x_n]$ satisfying

$$\sigma A + \tau B = C \text{ with } \deg(\sigma, x_1) < \deg(B, x_1)$$

1 If $n = 1$ solve $\sigma A + \tau B = C$ using the Euclidean algorithm in $\mathbb{Z}_p[x_1]$.

2 $(\sigma_0, \tau_0) :=$ MultiDioLift($A(x_n = \alpha_n)$, $B(x_n = \alpha_n)$, $C(x_n = \alpha_n)$, α)

Wang's Multivariate Diophantine Solver

Input $A, B, C \in \mathbb{Z}_p[x_1, \ldots, x_n]$ and $\alpha = (\alpha_2, \ldots \alpha_n)$
Output $\sigma, \tau \in \mathbb{Z}_p[x_1, \ldots, x_n]$ satisfying

$\sigma A + \tau B = C$ with $\deg(\sigma, x_1) < \deg(B, x_1)$

1 If $n = 1$ solve $\sigma A + \tau B = C$ using the Euclidean algorithm in $\mathbb{Z}_p[x_1]$.

2 $(\sigma_0, \tau_0) :=$ MultiDioLift($A(x_n = \alpha_n)$, $B(x_n = \alpha_n)$, $C(x_n = \alpha_n)$, α)

3 Initialize: $(\sigma, \tau) := (\sigma_0, \tau_0)$ $\quad$ $error := C - \sigma A - \tau B$
4 For $k = 1, 2, \ldots$ while $error \neq 0$ do
$c_k :=$ coeff($error$, $(x_n - \alpha_n)^k$)
If $c_k \neq 0$ then

Solve the MDP $\sigma_k \tau_0 + \tau_k \sigma_0 = c_k$ in $\mathbb{Z}_p[x_1, \ldots, x_{n-1}]$.

$\sigma_k, \tau_k :=$ MultiDioLift($\sigma_0, \tau_0, c_k, \alpha$)
$\sigma := \sigma + \sigma_k (x_n - \alpha_n)^k$; $\tau := \tau + \tau_k (x_n - \alpha_n)^k$
$error := error - \sigma_k (x_n - \alpha_n)^k A - \tau_k (x_n - \alpha_n)^k B$

5 output (σ, τ).

Wang's Multivariate Diophantine Solver

Input $A, B, C \in \mathbb{Z}_p[x_1, \ldots, x_n]$ and $\alpha = (\alpha_2, \ldots \alpha_n)$
Output $\sigma, \tau \in \mathbb{Z}_p[x_1, \ldots, x_n]$ satisfying

$$\sigma A + \tau B = C \text{ with } \deg(\sigma, x_1) < \deg(B, x_1)$$

1 If $n = 1$ solve $\sigma A + \tau B = C$ using the Euclidean algorithm in $\mathbb{Z}_p[x_1]$.

2 $(\sigma_0, \tau_0) :=$ MultiDioLift($A(x_n = \alpha_n)$, $B(x_n = \alpha_n)$, $C(x_n = \alpha_n)$, α)

3 Initialize: $(\sigma, \tau) := (\sigma_0, \tau_0)$ $\quad$ $error := C - \sigma A - \tau B$

4 For $k = 1, 2, \ldots$ while $error \neq 0$ do
   $c_k :=$ coeff($error$, $(x_n - \alpha_n)^k$)
   If $c_k \neq 0$ then
      Solve the MDP $\sigma_k \tau_0 + \tau_k \sigma_0 = c_k$ in $\mathbb{Z}_p[x_1, \ldots, x_{n-1}]$.
      $\sigma_k, \tau_k :=$ MultiDioLift(σ_0, τ_0, c_k, α)
      $\sigma := \sigma + \sigma_k (x_n - \alpha_n)^k$; $\tau := \tau + \tau_k (x_n - \alpha_n)^k$
      $error := error - \sigma_k (x_n - \alpha_n)^k A - \tau_k (x_n - \alpha_n)^k B$

5 output (σ, τ).

Let $M(n)$ count calls to the Euclidean algorithm and $d > \deg(C's, x_i)$.

Then $M(1) = 1, M(n) \leq M(n-1) + (d-1)M(n-1) \Longrightarrow \boxed{M(n) \leq d^{n-1}}$.

The Taylor Coefficients

$$f = x^3 - xyz^2 + y^3z^2 + z^4 - 2$$

Consider $f = f_0 + f_1(z-\alpha_3) + f_2(z-\alpha_3)^2 + f_3(z-\alpha_3)^3 + f_4(z-\alpha_3)^4$.

If $\alpha_3 = 0$ then $f(z) = \underbrace{(x^3-2)}_{f_0} + \underbrace{(y^3-xy)}_{f_2} z^2 + \underbrace{1}_{f_4} z^4.$

If $\alpha_3 = 2$ then

$$\begin{aligned} f(z) &= \underbrace{(x^3 + 4\,y^3 - 4\,xy + 14)}_{f_0} + \underbrace{(4\,y^3 - 4\,xy + 32)}_{f_1}(z-2) + \\ &\quad \underbrace{(y^3 - xy + 24)}_{f_2}(z-2)^2 + \underbrace{8}_{f_3}(z-2)^3 + \underbrace{1}_{f_4}(z-2)^4 \end{aligned}$$

The Taylor Coefficients

$$f = x^3 - xyz^2 + y^3z^2 + z^4 - 2$$

Consider $f = f_0 + f_1(z-\alpha_3) + f_2(z-\alpha_3)^2 + f_3(z-\alpha_3)^3 + f_4(z-\alpha_3)^4.$

If $\alpha_3 = 0$ then $f(z) = \underbrace{(x^3-2)}_{f_0} + \underbrace{(y^3-xy)}_{f_2} z^2 + \underbrace{1}_{f_4} z^4.$

If $\alpha_3 = 2$ then

$$f(z) = \underbrace{(x^3 + 4\,y^3 - 4\,xy + 14)}_{f_0} + \underbrace{(4\,y^3 - 4\,xy + 32)}_{f_1}(z-2) + \underbrace{(y^3 - xy + 24)}_{f_2}(z-2)^2 + \underbrace{8}_{f_3}(z-2)^3 + \underbrace{1}_{f_4}(z-2)^4$$

Lemma (MT 2016)

If α_j is chosen at random from a sufficiently large set then

$$\mathrm{Prob}[\,\mathrm{supp}(f_0) \supseteq \mathrm{supp}(f_1) \supseteq \cdots \supseteq \mathrm{supp}(f_k)\,] \text{ is high}$$

MTSHL : Our Multivariate Diophantine Solver

Solve the MDP $f_k \; g_0 + g_k \; f_0 = c_k$ for $f_k, g_k \in \mathbb{Z}_p[x_1, \ldots, x_l]$.

1: Let $f_{k-1} = \sum_{i=0}^{df} a_i(x_2, \ldots, x_l) x_1^i$. Let $t_i = |\text{supp}(a_i)|$.
Set $f_k = \sum_{i=0}^{df} \left(\sum_{M \in \text{supp}(a_i)} a_{iM} M \right) x_1^i$.

Assumes $\text{supp}(f_k) \subseteq \text{supp}(f_{k-1})$.

2: Set $t = \max(t_i)$. Pick $\beta = (\beta_2, \ldots, \beta_l) \in \mathbb{Z}_p$, all non-zero, at random.

3: For $1 \le j \le t$ solve $\sigma_j \; g_0(\beta^j) + \tau_j \; f_0(\beta^j) = c_k(\beta^j)$ for $\sigma_j, \tau_j \in \mathbb{Z}_p[x_1]$.

Needs $\gcd(g_0(\beta^j), f_0(\beta^j)) = 1$ for all $1 \le j \le t$.

4: For $1 \le j < df$ solve the $t_i \times t_i$ shifted Vandermonde linear system

$$\left\{ coeff(f_k(\beta^j), x_1^i) = coeff(\sigma_j, x_1^i) \text{ for } 1 \le j \le t_i \right\}.$$

Needs $X(\beta) \neq Y(\beta)$ for each pair (X, Y) in $\text{supp}(a_i)$ for $0 \le i < df$.

5: Do this also for g_k.

Shifted Transposed Vandermonde Systems

Let $f_k = \sum_{i=0}^{df} \left(\sum_{j=1}^{t_i} a_{ij} M_{ij}(x_2, \ldots, x_l) \right) x_1^i$, M_{ij} known, a_{ij} unknown.

Compute $m_{ij} = M_{ij}(\beta)$ for $1 \leq j \leq t_i$ and solve the t_i by t_i linear system $V_S a = b$

$$\begin{bmatrix} m_{i1} & m_{i2} & \cdots & m_{it_i} \\ m_{i1}^2 & m_{i2}^2 & \cdots & m_{it_i}^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_{i1}^{t_i} & m_{i2}^{t_i} & \cdots & m_{it_i}^{t_i} \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{it_i} \end{bmatrix} = \begin{bmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{it_i} \end{bmatrix}$$

using [Zip90] in $O(t_i^2)$ time and $O(t_i)$ space and using the factorization

$$V_i = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ m_{i1} & m_{i2} & \cdots & m_{it_i} \\ \vdots & \vdots & \vdots & \vdots \\ m_{i1}^{t_i-1} & m_{i2}^{t_i-1} & \cdots & m_{it_i}^{t_i-1} \end{bmatrix} \begin{bmatrix} m_{i1} & 0 & \cdots & 0 \\ 0 & m_{i2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & m_{it_i} \end{bmatrix}$$

The matrix V_i is invertible iff $m_{ij} \neq m_{ik}$ and $m_{ij} \neq 0$.

The Schwartz-Zippel Lemma

Let $L = \begin{bmatrix} 1 & k & k & k \\ s & 1 & 0 & 0 \\ 0 & s & 1 & 0 \\ 0 & 0 & s & sk \end{bmatrix}$ and $f = \det(L) = sk(s-1)(ks-s-1)$

If we pick α, β from a finite set S e.g. $\mathbb{F}_9$, what is $\text{Prob}[f(\alpha,\beta) = 0]$?

The Schwartz-Zippel Lemma

Let $L = \begin{bmatrix} 1 & k & k & k \\ s & 1 & 0 & 0 \\ 0 & s & 1 & 0 \\ 0 & 0 & s & sk \end{bmatrix}$ and $f = \det(L) = sk(s-1)(ks-s-1)$

If we pick α, β from a finite set S e.g. $\mathbb{F}_9$, what is $\text{Prob}[f(\alpha,\beta)=0]$? **31/81**

Lemma (Schwartz-Zippel, 1979)

See wikipedia for a proof.

Let f be a non-zero polynomial in $F[x_1,\ldots,x_n]$ where F is a field e.g. $\mathbb{Q}$ or $\mathbb{F}_q$.

If $\alpha_1,\ldots,\alpha_n$ are chosen at random from $S \subset F$ then

$$\text{Prob}[f(\alpha_1,\ldots,\alpha_n)=0] \le \frac{\deg f}{|S|}.$$

The Schwartz-Zippel Lemma

Let $L = \begin{bmatrix} 1 & k & k & k \\ s & 1 & 0 & 0 \\ 0 & s & 1 & 0 \\ 0 & 0 & s & sk \end{bmatrix}$ and $f = \det(L) = sk(s-1)(ks-s-1)$

If we pick α, β from a finite set S e.g. $\mathbb{F}_9$, what is $\text{Prob}[f(\alpha,\beta) = 0]$? **31/81**

Lemma (Schwartz-Zippel, 1979)

See wikipedia for a proof.

Let f be a non-zero polynomial in $F[x_1, \ldots, x_n]$ where F is a field e.g. $\mathbb{Q}$ or $\mathbb{F}_q$.

If $\alpha_1, \ldots, \alpha_n$ are chosen at random from $S \subset F$ then

$$\text{Prob}[f(\alpha_1, \ldots, \alpha_n) = 0] \leq \frac{\deg f}{|S|}.$$

Note: $\deg(\det L) \leq 1+1+1+2 = 5$ so the SZ bound is **5/9 = 45/81**.

Monomial evaluations

Let $f_k = \sum_{i=0}^{df} \left(\sum_{j=1}^{t_i} a_{ij} M_{ij}(x_2, \ldots, x_l) \right) x_1^i$.

2: Pick $\beta = (\beta_2, \ldots, \beta_l) \in \mathbb{Z}_p$, all non-zero, at random.
Compute $S_i = \{ m_{ij} = M_{ij}(\beta) : 1 \leq j \leq t_i \}$ for $1 \leq i \leq df$.

We need distinct monomial evaluations m_{ij} for each S_i, equivalently, $|S_i| = t_i$.

Monomial evaluations

Let $f_k = \sum_{i=0}^{df} \left(\sum_{j=1}^{t_i} a_{ij} M_{ij}(x_2, \ldots, x_l) \right) x_1^i$.

2: Pick $\beta = (\beta_2, \ldots, \beta_l) \in \mathbb{Z}_p$, all non-zero, at random.
Compute $S_i = \{m_{ij} = M_{ij}(\beta) : 1 \le j \le t_i\}$ for $1 \le i \le df$.

We need distinct monomial evaluations m_{ij} for each S_i, equivalently, $|S_i| = t_i$.

Let $R_i(x_2, \ldots, x_l) = \prod_{1 \le j < k \le t_i} (M_{ij} - M_{ik})$ for $1 \le i \le df$.

$$Prob[|S_i| < t_i] \quad = \quad Prob[R_i(\beta) = 0]$$

Monomial evaluations

Let $f_k = \sum_{i=0}^{df} \left(\sum_{j=1}^{t_i} a_{ij} M_{ij}(x_2, \ldots, x_l) \right) x_1^i$.

2: Pick $\beta = (\beta_2, \ldots, \beta_l) \in \mathbb{Z}_p$, all non-zero, at random.
Compute $S_i = \{ m_{ij} = M_{ij}(\beta) : 1 \leq j \leq t_i \}$ for $1 \leq i \leq df$.

We need distinct monomial evaluations m_{ij} for each S_i, equivalently, $|S_i| = t_i$.

Let $R_i(x_2, \ldots, x_l) = \prod_{1 \leq j < k \leq t_i} (M_{ij} - M_{ik})$ for $1 \leq i \leq df$.

$$
\begin{aligned}
Prob[|S_i| < t_i] &= Prob[R_i(\beta) = 0] \\
&\leq \frac{\deg R_i}{p-1} \quad \text{by} \quad \text{Schwartz} - \text{Zippel} \\
&\leq \frac{\binom{t_i}{2} \deg f_k}{p-1}.
\end{aligned}
$$

Benchmark 1

$n/d/T$	Wang (MDP)	Kaltofen (MDP)	MTSHL (MDP)
4/35/100	13.07 (11.95)	1.75 (1.18)	1.51 (0.24)
5/35/100	88.10 (86.28)	3.75 (2.57)	1.16 (0.36)
7/35/100	800.0 (797.0)	5.04 (4.08)	1.58 (0.59)
9/35/100	4451.6 (4449.4)	8.13 (6.22)	2.94 (0.56)
4/35/500	33.96 (26.48)	642.2 (635.1)	11.29 (0.82)
5/35/500	472.1 (402.5)	1916.2 (1899.6)	26.0 (4.86)
7/35/500	3870.5 (3842.2)	2329.4 (2305.5)	43.1 (6.84)
9/35/500	> 60000	3866.3 (3805.9)	79.6 (9.71)

$> \ M \ := \ product(\ x_i, \ i = 2..n \ );$
$> \ f \ := \ x_1^d \ + \ M \, randpoly(\ [x_1, \ldots, x_n], \ terms = T-1, \ degree = d \ );$
$> \ g \ := \ x_1^d \ + randpoly(\ [x_1, \ldots, x_n], \ terms = T-1, \ degree = d \ );$

M forces the evaluation points $x_2 \neq 0, \ldots, x_n \neq 0$.

Kaltofen, E., Sparse Hensel lifting. Proceedindgs of EUROCAL '85, LNCS **204**, pp. 4–17, 1985.

Factoring the determinants of Cyclic matrices.

Let C_n denote the $n \times n$ cyclic matrix below.

$$C_n = \begin{bmatrix} x_1 & x_2 & \dots & x_{n-1} & x_n \\ x_n & x_1 & \dots & x_{n-2} & x_{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_3 & x_4 & \dots & x_1 & x_2 \\ x_2 & x_3 & \dots & x_n & x_1 \end{bmatrix}$$

The determinant of C_n is a homogeneous polynomials in n variables.
It has dense factors – not suited to MTSHL.

Example

$$\det C_4 = (x_4 + x_3 + x_1 + x_2)(x_4 - x_3 - x_1 + x_2)$$
$$\left({x_1}^2 - 2\,x_1x_3 + {x_2}^2 - 2\,x_2x_4 + {x_3}^2 + {x_4}^2\right)$$

Benchmark 2

n	$\#d_n$	#fmax	Maple	(MDP)	MTSHL	Magma	Singular
7	246	924	0.045	90%	0.026	0.01	0.02
8	810	86	0.059	46%	0.063	0.07	0.06
9	2704	1005	0.225	74%	0.120	0.74	0.24
10	7492	715	0.853	62%	0.500	8.44	2.02
11	32066	184756	7.160	91%	0.945	104.3	11.39
12	86500	621	19.76	76%	5.121	7575.1	30.27
13	400024	2704156	263.4	92%	27.69	30871.9	??
14	1366500	27132	1664.4	77%	523.07	$> 10^6$	288463.2
15	4614524	303645	18432.	82%	7496.9	–	–

Table: Factorization timings (seconds) for det C_n evaluated at $x_n = 1$

Notes: ?? $=$ I cannot compute det(C_n) nor read in det(C_n) nor it's factors.

$$\begin{array}{ccc}
a_j(x_1, x_2, \ldots, x_j) & & f_j(x_1, x_2, x_3, \ldots, x_j) \\
f_{j-1}(x_1, x_2, \ldots, x_{j-1}) & & g_j(x_1, x_2, x_3, \ldots, x_j) \\
\text{evaluate } x_3, \ldots, x_{j-1} \text{ for } 1 \le k \le s & & \uparrow \\
\downarrow & & \text{interpolate } x_3, \ldots, x_{j-1} \\
a_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) & \text{Hensel} & f_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
f_{j-1}(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k) & \xrightarrow{\text{lift } x_j} & g_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
\text{evaluate } x_2 \text{ for } 1 \le l \le \deg(a_j, x_2) & & \uparrow \\
\downarrow & & \text{dense interpolate } x_2 \\
a_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) & \text{Hensel} & f_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
f_{j-1}(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k) & \xrightarrow{\text{lift } x_j} & g_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j)
\end{array}$$

Monagan and Tuncer ICMS 2018

$$
\begin{array}{ccc}
a_j(x_1, x_2, \ldots, x_j) & & f_j(x_1, x_2, x_3, \ldots, x_j) \\
f_{j-1}(x_1, x_2, \ldots, x_{j-1}) & & g_j(x_1, x_2, x_3, \ldots, x_j) \\
\text{evaluate } x_3, \ldots, x_{j-1} \text{ for } 1 \le k \le s & & \uparrow \\
\downarrow & & \text{interpolate } x_3, \ldots, x_{j-1} \\
a_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) & \text{Hensel} & f_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
 & \text{lift } x_j & \\
f_{j-1}(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k) & \longrightarrow & g_j(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
\text{evaluate } x_2 \text{ for } 1 \le l \le \deg(a_j, x_2) & & \uparrow \\
\downarrow & & \text{dense interpolate } x_2 \\
a_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) & \text{Hensel} & f_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j) \\
 & \text{lift } x_j & \\
f_{j-1}(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k) & \longrightarrow & g_j(x_1, \gamma_{\mathbf{m}}, \beta_3^k, \ldots, \beta_{j-1}^k, x_j)
\end{array}
$$

Parallelized evaluations $a(x_1, x_2, \beta_3^k, \ldots, \beta_{j-1}^k, x_j)$ using Cilk C
Do evaluations $x_2 = \gamma_{\mathbf{m}}$ and **bivariate Hensel lifts** in parallel.
Optimize the bivariate Hensel lifts in $\mathbb{Z}_p[x_1, x_j]$ mod $(x_j - \alpha_j)^k$.

Parallel Benchmark 1

On a server with 2 Intel Xeon E2660 8 core CPUs – 2.2 GHz(base) – 3.0 GHz(turbo)

			Maple 2018		New times (1 core)			New times (16 cores)		
n	d	t	best	worst	total	(hensel)	(eval)	total	(hensel)	(eval)
6	7	500	0.411	28.84	0.098	(0.015)	(0.042)	0.074	(0.019)	(0.008 – 5.2x)
6	7	1000	1.140	58.46	0.414	(0.025)	(0.247)	0.180	(0.027)	(0.030 – 8.2x)
6	7	2000	3.066	99.88	1.593	(0.041)	**(1.132)**	0.285	(0.042)	**(0.121 – 9.4x)**
6	7	4000	7.173	162.49	5.072	(0.069)	(4.070)	0.814	(0.074)	(0.380 – 10.7x)
9	7	500	1.171	7564.9	0.105	(0.013)	(0.040)	0.101	(0.024)	(0.010 – 4.0x)
9	7	1000	3.704	10010.4	0.524	(0.025)	(0.297)	0.233	(0.026)	(0.030 – 11.4x)
9	7	2000	13.43	NA	2.838	(0.042)	**(1.973)**	0.483	(0.045)	**(0.193 – 10.2x)**
9	7	4000	51.77	NA	18.35	(0.078)	(14.84)	2.325	(0.083)	(1.350 – 11.0x)

Table 1: Timings (real time in seconds) for Hensel lift of x_n.
Legend: $n = \#$variables, $d = \deg(f, x_j) = \deg(g, x_j)$, $t = \#f = \#g$.

Parallel Benchmark 2

			Maple 2018		New time (1 core)			New time (16 cores)		
n	d	t	best	worst	total	(hensel)	(eval)	total	(hensel)	(eval)
6	15	500	0.751	7956.5	0.134	(0.070)	(0.016)	0.093	(0.034 – 2.1x)	(0.006)
6	20	500	0.919	48610.1	0.238	(0.168)	(0.017)	0.130	(0.065 – 2.6x)	(0.005)
6	40	500	1.615	NA	1.207	**(1.128)**	(0.015)	0.282	**(0.203 – 5.6x)**	
6	80	500	4.485	NA	13.76	(13.65)	(0.016)	1.674	(1.554 – 8.8x)	(0.012)
6	15	2000	7.166	23128.5	1.616	(0.221)	(0.706)	0.413	(0.107 – 2.1x)	(0.061)
6	20	2000	9.195	NA	1.635	(0.451)	(0.480)	0.431	(0.150 – 3.0x)	(0.040)
6	40	2000	15.98	NA	4.008	**(2.993)**	(0.260)	0.854	**(0.505 – 5.9x)**	(0.038)
6	80	2000	57.33	NA	26.34	(25.25)	(0.217)	3.340	(2.839 – 8.9x)	(0.050)

Table 2: Timings (real time in seconds) for increasing degree d.
Legend: $n = \#$variables, $d = \deg(f, x_j) = \deg(g, x_j)$, $t = \#f = \#g$.

Concluding Remarks

Baris has installed the new MTSHL code into Maple for Maple 2019. This was done under a MITACS internship with Maplesoft.

- Wang, P.S., Rothschild, L.P. Factoring multivariate polynomials over the integers. *Mathematics of Computation*, **29**(131): 935–950, (1975).
- Paul Wang. An improved Multivariate Polynomial Factoring Algorithm. *Mathematics of Computation*, **32**(144):1215–1231, 1978.
- Erich Kaltofen. Sparse Hensel lifting. *Proc. EUROCAL '85*, LNCS **204**, pp. 4–17, 1985.
- Monagan and Tuncer. Using Sparse Interpolation in Hensel Lifting. *Proc. CASC 2016*. LNCS **9890**, pp. 381–400, 2016.
- Monagan and Tuncer. Factoring multivariate polynomials with many factors and huge coefficients. *Proc. CASC 2018*, LNCS **11077**, pp. 319–400, 2018.
- Monagan and Tuncer. Sparse multivariate polynomial factorization: a high-performance design and implementation. *Proc. ICMS 2018*, LNCS **10931**, pp. 359-368, 2018.
- Monagan and Tuncer. The complexity of sparse Hensel lifting and sparse polynomial factorization. To appear in *J. Symbolic Comp.*, 2019. DOI: 10.1016/j.jsc.2019.05.001