Factoring Multivariate Polynomials Given by Black Boxes

Michael Monagan

Department of Mathematics,
Simon Fraser University, Canada

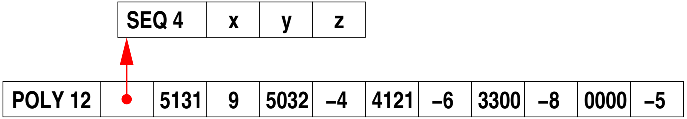
Joint work with Tian Chen, Simon Fraser University

Sparse Polynomials and Sparse Polynomial Representations

Let R be a ring and $f = \sum_{i=1}^{t} c_i M_i(x_1, \ldots, x_n)$ be a polynomial in $R[x_1, x_2, \ldots, x_n]$ where the coefficients $c_i \neq 0$ and the monomials M_i are distinct, so that t is the number of terms of f.
If f has total degree $d = \deg f$, then $t \leq \binom{n+d}{d}$.

Definition: We say f is sparse if $t \leq \sqrt{\binom{n+d}{d}}$, otherwise f is dense.
Example: $f = 9xy^3z - 4y^3z^2 - 6xy^2z - 8x^3 - 5$, $n = 3$, $d = 5$, $t = 5$, $\binom{n+d}{d} = 56$.

Definition: In a sparse representation of a polynomial, there are no 0 coefficients.
Example: Maple's POLY data structure for $f = 9xy^3z - 4y^3z^2 - 6xy^2z - 8x^3 - 5$.



Black Boxes in Computer Algebra

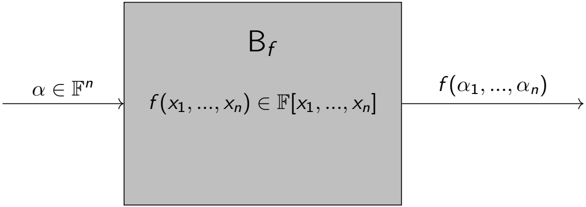


Figure: A black box for a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ where $\mathbb{F}$ is a field

Erich Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Cmpt.* **9**(3), 301–320. Elsevier (1990)

Consider the polynomial $f = (x_1 - x_2)(x_1 - x_3) \times \cdots \times (x_1 - x_n)$.

```
Magma V2.28-23    Wed Feb  4 2026 17:14:00 on cecm-maple [Seed = 3109226947]
Type ? for help.  Type <Ctrl>-D to quit.
> Z := IntegerRing();
> P<x1,x2,x3,x4> := PolynomialRing(Z,4);
> f := (x1-x2)*(x1-x3)*(x1-x4);
> f;
x1^3 - x1^2*x2 - x1^2*x3 - x1^2*x4 + x1*x2*x3 + x1*x2*x4 + x1*x3*x4 - x2*x3*x4
```

Consider the polynomial $f = (x_1 - x_2)(x_1 - x_3) \times \cdots \times (x_1 - x_n)$.

```
Magma V2.28-23    Wed Feb  4 2026 17:14:00 on cecm-maple [Seed = 3109226947]
Type ? for help.  Type <Ctrl>-D to quit.
> Z := IntegerRing();
> P<x1,x2,x3,x4> := PolynomialRing(Z,4);
> f := (x1-x2)*(x1-x3)*(x1-x4);
> f;
x1^3 - x1^2*x2 - x1^2*x3 - x1^2*x4 + x1*x2*x3 + x1*x2*x4 + x1*x3*x4 - x2*x3*x4
```

```
> B := proc(x::list) local f,i;
>      f := 1;
>      for i from 2 to nops(x) do f := f*(x[1]-x[i]); od;
>      f;
> end:
> B([1,3,5,7]);
                                      -48

> B([x1,x2,x3,x4]);
                         (x1 - x2) (x1 - x3) (x1 - x4)
```

Modular Black Boxes

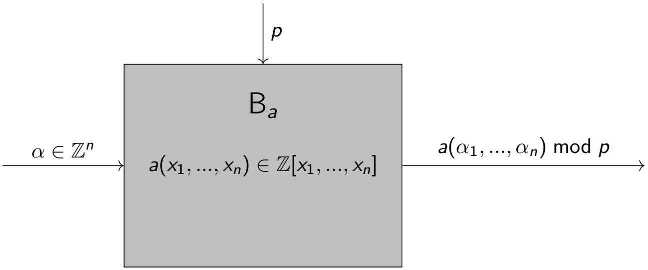


Figure: A modular black box for $a \in \mathbb{Z}[x_1, \ldots, x_n]$

Symmetric Toeplitz Matrix Determinants

```
> T4 := Matrix( [[x[1],x[2],x[3],x[4]],[x[2],x[1],x[2],x[3]],
>               [x[3],x[2],x[1],x[2]],[x[4],x[3],x[2],x[1]]]);
```

$$T4 := \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_1 & x_2 & x_3 \\ x_3 & x_2 & x_1 & x_2 \\ x_4 & x_3 & x_2 & x_1 \end{bmatrix}$$

```
> det := LinearAlgebra[Determinant](T4);
```

$$det := x_1^4 - 3x_1^2x_2^2 - 2x_1^2x_3^2 - x_1^2x_4^2 + 4x_1x_2^2x_3 + 4x_1x_2x_3x_4 + x_2^4 - 2x_2^3x_4 - 2x_2^2x_3^2 + x_2^2x_4^2 - 2x_2x_3^2x_4 + x_3^4$$

Here $n = 4$, $d = 4$, $t = 12$ and $\binom{n+d}{d} = 70$ so $\det(T_4)$ is dense.

```
> factor(det);
```

$$\left(x_1^2 - x_1x_2 - x_1x_4 - x_2^2 + 2x_2x_3 + x_2x_4 - x_3^2\right)\left(x_1^2 + x_1x_2 + x_1x_4 - x_2^2 - 2x_2x_3 + x_2x_4 - x_3^2\right)$$

A Black Box for det(A)

```
> MakeBDet := proc(A::Matrix,X::list(name))
>     proc(x::list,p::prime) local n,B,i,det;
>         n := nops(X);
>         B := Eval(A,{seq(X[i]=x[i],i=1..n)}) mod p;
>         det := Det(B) mod p;  # Gaussian elimination in Zp
>     end;
> end:
>
> T4 := Matrix( [[x[1],x[2],x[3],x[4]],[x[2],x[1],x[2],x[3]],
>                [x[3],x[2],x[1],x[2]],[x[4],x[3],x[2],x[1]]]):
> B := MakeBDet(T4,[x[1],x[2],x[3],x[4]]):
> B([1,3,5,7],101);
                                        47
```

Computational Problems with Black Boxes

Let $B : \mathbb{F}^n \to \mathbb{F}$ be a black box for $f \in \mathbb{F}[x_1, \ldots, x_n]$.

1. Is $f = 0$?
2. Compute $\deg(f, x_i)$ for some $1 \leq i \leq n$.
3. Compute $\deg(f)$.
4. For $\alpha \in \mathbb{F}^n$ compute $\partial f(\alpha)/\partial x_i$.
5. Compute the factors of f over $\mathbb{F}$ in the sparse reprentation.
6. Deterime t the number of terms of f in the expanded representation.
7. Interpolate $f(x_1, \ldots, x_n)$ in the sparse representation.

Arithmetic is easy.

```
> BBmultiply := prob(A::procedure,B::procedure)
>     proc(alpha::list) A(alpha)*B(alpha) end;
> end:
```

The Schwartz-Zippel Lemma

Let $B : \mathbb{F}^n \rightarrow \mathbb{F}$ be a black box for $f \in \mathbb{F}[x_1, \ldots, x_n]$.

Problem 1: Is $f = 0$? Idea: pick $\alpha \in \mathbb{F}^n$.
If $B(\alpha) \neq 0$ then $f \neq 0$. But if $B(\alpha) = 0$ then $f \neq 0$ is possible.

The Schwartz-Zippel Lemma

Let $B : \mathbb{F}^n \to \mathbb{F}$ be a black box for $f \in \mathbb{F}[x_1, \ldots, x_n]$.

Problem 1: Is $f = 0$? Idea: pick $\alpha \in \mathbb{F}^n$.
If $B(\alpha) \neq 0$ then $f \neq 0$. But if $B(\alpha) = 0$ then $f \neq 0$ is possible.

Theorem (Schwartz-Zippel) Let S be a finite subset of $\mathbb{F}$ and let $d = \deg f$.
Suppose we pick $\alpha \in S^n$ at random. Then

(i) If $f \neq 0$ then f has at most $d|S|^{n-1}$ roots in S^n.

(ii) If $f \neq 0$ then $\Pr(f(\alpha) = 0) \leq d/|S|$.

If we pick S with $|S| > 10^9$ and $\beta \neq \alpha$ from S^n at random then

$$\Pr(f(\alpha) = 0 \text{ and } f(\beta) = 0) \leq \frac{d^2}{|S|(|S|-1)}.$$

Problem 2: Computing $\deg(f, x_1)$

Let $B : \mathbb{F}^n \to \mathbb{F}$ be a black box for $f \in \mathbb{F}[x_1, \ldots, x_n]$.

Suppose $f \neq 0$ and let $d = \deg(f, x_1)$. Let

$$f = a_d(x_2, \ldots, x_n)x_1^d + \cdots + a_1(x_2, \ldots, x_n)x + a_0(x_2, \ldots, x_n).$$

Pick $\alpha \in S^n$ at random and let $g(x) = f(x, \alpha_2, \ldots, \alpha_n)$. Then

$$\Pr(\deg(g, x) < \deg(f, x_1)) = \Pr(a_d(\alpha) = 0) \leq \frac{\deg(a_d)}{|S|}.$$

Idea: construct a blackbox for $g(x)$ from the black box for $f(x_1, \ldots, x_n)$ and interpolate $g(x)$ to determine $\deg g$.

Interpolating $g(x)$ from a black box

Suppose $\deg(g) = d$. Let $\alpha_0, \alpha_1, \ldots, \alpha_d$ be distinct in $\mathbb{F}$. Then

$$g(x) = \underbrace{v_0 + v_1(x-\alpha_0) + \cdots + v_k \prod_{i=0}^{k-1}(x-\alpha_i)}_{h_k(x)} + \cdots + v_d \prod_{i=0}^{d-1}(x-\alpha_i) + v_{d+1} \prod_{i=0}^{d}(x-\alpha_i)$$

for some $v_i \in \mathbb{F}$ where $v_d \neq 0$, $v_{d+1} = 0$. We have $v_k = (g(\alpha_k) - h_{k-1}(\alpha_k)) / \prod_{i=0}^{k-1}(\alpha_k - \alpha_i)$.

Algorithm: Pick distinct $\alpha_0, \alpha_1, \alpha_2, \ldots$ **at random** from $S \subset \mathbb{F}$.
Compute $v_0, v_1, \ldots v_k, \ldots$, stop when $v_k = 0$, and output $k-1$.

$$\begin{aligned} \Pr(v_k = 0 \text{ for } 0 \le k \le d) &= \Pr(g(\alpha_k) - h_k(\alpha_k) = 0 \text{ for some } 0 \le k \le d) \\ &\le \sum_{k=0}^{d} \frac{\deg(g)}{|S| - k} \le \frac{(d+1)d}{|S| - (d-1)}. \end{aligned}$$

Interpolating bivariate polynomials from a black box

Let $B : \mathbb{F}^n \rightarrow \mathbb{F}$ be a black box for $f \in \mathbb{F}[x_1, \ldots, x_n]$. For $\alpha \in \mathbb{F}^n$ how can we interpolate

$$g(x_1, x_j) = f(x_1, \alpha_2, \ldots, \alpha_{j-1}, x_j, \alpha_{j+1}, \ldots, x_n).$$

$f(x,y)$	$(5y+3)\,x^2$	$+(9y^2+4y+8)x$	$+18x^3+101x+28$
$f(x,1)$	$8\,x^2$	$+21x$	$+10$
$f(x,2)$	$13\,x^2$	$+52x$	$+19$
$f(x,3)$	$18\,x^2$	$+101x$	$+28$

If $\deg(f,x) = d_x$ and $\deg(f,y) = d_y$ we need $2(d_x+1)(d_y+1)$ probes to **B**
Using Newton or Lagrange this does $O(d_x^2 d_y + d_y^2 d_x)$ field operations in $\mathbb{F}$.

A brief history of multivariate polynomial factorization

Given a polynomial $a \in \mathbb{Z}[x_1, \cdots, x_n]$, compute the irreducible factors of a with coefficients in $\mathbb{Z}$. Note that the integer content is not factored. E.g., $6x^2 - 6y^2 = 6(x+y)(x-y)$.

- Zassenhaus (1969): Hensel lifting for univariate polynomials in $\mathbb{Z}[x]$.
- Yun (1974), Wang (1975), (1978): **Multivariate Hensel lifting**.
  (can be exponential in the number of variables).

Sparse Hensel lifting

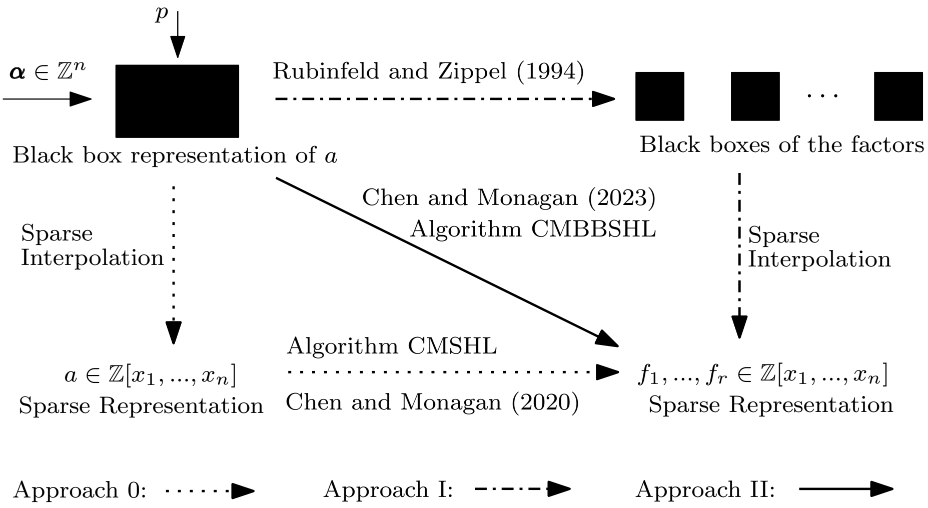
- Zippel (1981), Kaltofen (1985)
- Monagan and Tuncer (2016), (2018): MTSHL uses sparse interpolation.
- Chen and Monagan (2020): CMSHL uses many bivariate Hensel lifts.
  Dominating cost is evaluating the input polynomial –> black box representation

A brief history of black box polynomial factorization

Black box factorization

- Kaltofen and Trager (1990): First computes the black boxes of the factors, then uses sparse polynomial interpolation to recover the sparse representation of the factors.
- Diaz and Kaltofen (1998): FOXBOX. Implemented in C++.
- Rubinfeld and Zippel (1994): For factoring $a \in \mathbb{Z}[x_1, \cdots, x_n]$.
- Chen and Monagan (2022), (2023): A modular algorithm. Output factors in the sparse representation directly. Requires fewer probes to the black box than Rubinfeld and Zippel's algorithm and only one factorization in $\mathbb{Z}[x]$.

Factoring $a \in \mathbb{Z}[x_1, \cdots, x_n]$ represented by a black box



Factoring the determinant of a Toeplitz matrix

$$T_n = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_2 & x_1 & x_2 & & \\ x_3 & x_2 & x_1 & & \\ \vdots & & & \ddots & \vdots \\ x_n & & & \cdots & x_1 \end{pmatrix}.$$

For example,
$\det(T_4) = (x_1^2 - x_1x_2 - x_1x_4 - x_2^2 + 2x_2x_3 + x_2x_4 - x_3^2)(x_1^2 + x_1x_2 + x_1x_4 - x_2^2 - 2x_2x_3 + x_2x_4 - x_3^2)$.

n	$\#\det(T_n)$	$\#f_i$
8	1628	167, 167
9	6090	294, 153
10	23797	931, 931
11	90296	1730, 849
12	350726	5579, 5579
13	1338076	10611, 4983
14	5165957	34937, 34937
15	19732508	66684, 30458
16	76020346	221854, 221854
17	291057539	191164, 424292
18	–	1419659, 1419659
19	–	1209612, 2714726

Table: Number of terms of $\det(T_n)$ and its factors.

Example: Factor $a = \det(T_4)$ in $\mathbb{Z}[x_1, x_2, x_3, x_4]$

1. Pick a lifting prime $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.
2. Factor $a(x_1, \alpha) = x_1^4 - 93x_1^2 + 420x_1 - 416$ over $\mathbb{Z}$.
   We get $a(x_1, \alpha) = (x_1^2 - 7x_1 + 8)(x_1^2 + 7x_1 - 52) = f_1 g_1$.
3. The first Hensel lifting step recovers x_2 in the factors by solving
   $$a(x_1, x_2, \alpha_3, \alpha_4) \equiv f_2(x_1, x_2) g_2(x_1, x_2) \pmod{p}$$
4. The second Hensel lifting step recovers x_3 by solving
   $$a(x_1, x_2, x_3, \alpha_4) \equiv f_3(x_1, x_2, x_3) g_3(x_1, x_2, x_3) \pmod{p}$$
5. The final Hensel lifting step recovers x_4 by solving
   $$a(x_1, x_2, x_3, x_4) \equiv f_4(x_1, x_2, x_3, x_4) g_3(x_1, x_2, x_3, x_4) \pmod{p}$$

Requires (i) α to be "Hilbertian", that is, $f(x_1, \alpha)$ has two factors, and (ii) p large enough.
So we must check that $\det(T_4) = f_4 g_4$ over $\mathbb{Z}$.

Suppose $f = x^3 + y^3 z - xyz^2 + (z^3 - 27)$ is a factor of a.
Define $\mathrm{supp}(f, \{x, y\}) = \{x^3, y^3, xy, 1\}$.

Suppose $\alpha = 2$. We want to recover z from $f(x, y, 2) = x^3 + 4y^3 - 4xy - 19$.
Consider $f = \sum_{i=0}^{d} \sigma_i(x, y)(z - \alpha)^i$ where $d = \deg(f, z)$.

α	taylor$(f, z = \alpha)$
0	$\underbrace{(x^3 - 27)}_{\sigma_0} + \underbrace{(y^3 - xy)}_{\sigma_2} z^2 + \underbrace{1}_{\sigma_3} z^3$
2	$\underbrace{(x^3 + 4y^3 - 4xy - 19)}_{\sigma_0} + \underbrace{(4y^3 - 4xy + 12)}_{\sigma_1}(z - 2)$ $+ \underbrace{(y^3 - xy + 6)}_{\sigma_2}(z - 2)^2 + \underbrace{1}_{\sigma_3}(z - 2)^3.$

Suppose $f = x^3 + y^3z - xyz^2 + (z^3 - 27)$ is a factor of a.
Define $\mathrm{supp}(f, \{x, y\}) = \{x^3, y^3, xy, 1\}$.

Suppose $\alpha = 2$. We want to recover z from $f(x, y, 2) = x^3 + 4y^3 - 4xy - 19$.
Consider $f = \sum_{i=0}^{d} \sigma_i(x, y)(z - \alpha)^i$ where $d = \deg(f, z)$.

α	$\mathrm{taylor}(f, z = \alpha)$
0	$\underbrace{(x^3 - 27)}_{\sigma_0} + \underbrace{(y^3 - xy)}_{\sigma_2} z^2 + \underbrace{1}_{\sigma_3} z^3$
2	$\underbrace{(x^3 + 4y^3 - 4xy - 19)}_{\sigma_0} + \underbrace{(4y^3 - 4xy + 12)}_{\sigma_1}(z - 2) + \underbrace{(y^3 - xy + 6)}_{\sigma_2}(z - 2)^2 + \underbrace{1}_{\sigma_3}(z - 2)^3.$

The Weak Sparse Hensel Lifting Assumption:
$\mathrm{supp}(\sigma_i) \subset \mathrm{supp}(\sigma_0)$ for $1 \le i \le d$

The Strong Sparse Hensel Lifting Assumption:
$\mathrm{supp}(\sigma_i) \subset \mathrm{supp}(\sigma_{i-1})$ for $1 \le i \le d$

Algorithm CMBBSHL: Hensel lifting x_j (non-monic, non-square-free)

Input: A modular black box $B : (\mathbb{Z}^n, p) \to \mathbb{Z}_p$ for $a \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$, $\boldsymbol{\alpha} \in \mathbb{Z}^{n-1}$, a lifting prime p, $d_i = \deg(a, x_i)$ for $1 \le i \le n$ (pre-computed), $j \in \mathbb{Z}$ and $\hat{f}_{\rho,j-1} \in \mathbb{Z}_p[x_1, \cdots, x_{j-1}]$ for $1 \le \rho \le r$, s.t. $\mathrm{sqf}(a(x_1, \ldots, x_{j-1}, \alpha_j, \ldots, \alpha_n)) = \prod_{\rho=1}^{r} \lambda_\rho \hat{f}_{\rho,j-1}$.

Output: $(\hat{f}_{\rho,j}, 1 \le \rho \le r) \in \mathbb{Z}_p[x_1, \cdots, x_j]^r$ s.t.
(i) $\mathrm{sqf}(a_j) = \prod_{\rho=1}^{r} \lambda_\rho \prod_{\rho=1}^{r} \hat{f}_{\rho,j}$, and
(ii) $\hat{f}_{\rho,j}(x_j = \alpha_j) = \hat{f}_{\rho,j-1}$ for all $1 \le \rho \le r$; otherwise, FAIL.

1. Let $\hat{f}_{\rho,j-1} = \sum_{i=0}^{df_\rho} \sigma_{\rho,i}(x_2, \ldots, x_{j-1}) x_1^i$ $(1 \le \rho \le r)$ where $\sigma_{\rho,i} = \sum_{k=1}^{s_{\rho,i}} c_{\rho,ik} M_{\rho,ik}$.
2. Pick $\boldsymbol{\beta} = (\beta_2, \cdots, \beta_{j-1}) \in (\mathbb{Z}_p \backslash \{0\})^{j-2}$ at random.
3. Evaluate (for $1 \le \rho \le r$): $\mathcal{S}_\rho = \{\mathcal{S}_{\rho,i} = \{m_{\rho,ik} = M_{\rho,ik}(\boldsymbol{\beta}), 1 \le k \le s_{\rho,i}\}, 0 \le i \le df_\rho\}$.
4. **if** any $|\mathcal{S}_{\rho,i}| \ne s_{\rho,i}$ **then return** FAIL **end if** // monomial evals must be distinct
5. Let s be the maximum of $s_{\rho,i}$. // Compute s images of the factors in $\mathbb{Z}_p[x_1, x_j]$:

6 **for** k from 1 to s **do**

6.1 Let $Y_k = (x_2 = \beta_2^k, \cdots, x_{j-1} = \beta_{j-1}^k)$.

6.2 $A_k \leftarrow a(x_1, Y_k, x_j, \alpha_{j+1}, \ldots, \alpha_n) \in \mathbb{Z}_p[x_1, x_j]$. .... $\mathcal{O}(sd_1 d_j)$ probes $+ \mathcal{O}(s(d_1^2 d_j + d_1 d_j^2))$

6.3 **if** $\deg(A_k, x_1) \neq d_1$ **or** $\deg(A_k, x_j) \neq d_j$ **then return** FAIL **end if**

6.4 $g_k \leftarrow \gcd(A_k, \frac{\partial A_k}{\partial x_1}) \bmod p \in \mathbb{Z}_p[x_1, x_j]$. .... $\mathcal{O}(s(d_1^2 d_j + d_1 d_j^2))$

6.5 **if** $\deg(g_k, x_1) \neq d_1 - \sum_{\rho=1}^{r} df_\rho$ **then return** FAIL **end if**

6.6 $A_{sf} \leftarrow$ quo (A_k, g_k) mod p. // $A_{sf} = \mathrm{sqf}(A_k)$ mod p, up to a constant in $\mathbb{Z}_p$.

6.7 $A_{sfm} \leftarrow A_{sf} / (\mathrm{LC}(\mathrm{LC}(A_{sf}, x_1), x_j))$ mod p. // make $\mathrm{LC}(A_{sf}, x_1)$ monic in x_j.

6.8 $F_{\rho,k} \leftarrow \hat{f}_{\rho,j-1}(x_1, Y_k) \in \mathbb{Z}_p[x_1]$ for $1 \leq \rho \leq r$. .... $\mathcal{O}(s(\sum_{\rho=1}^{r} \#\hat{f}_{\rho,j-1}))$

6.9 **if** any $\deg(F_{\rho,k}) < df_\rho$ (for $1 \leq \rho \leq r$) **then return** FAIL **end if**

6.10 **if** $\gcd(F_{\rho,k}, F_{\phi,k}) \neq 1$ for any $1 \leq \rho < \phi \leq r$ **then return** FAIL **end if**

6.11 $\hat{f}_{\rho,k} \leftarrow$ BivariateHenselLift$(A_{sfm}(x_1, x_j), F_{\rho,k}(x_1), \alpha_j, p)$. .... $\mathcal{O}(s(d_1 d_j^2 + d_1^2 d_j))$

We have $A_{sfm} = \prod_{i=1}^{\rho} \hat{f}_{\rho,k}$ in $\mathbb{Z}_p[x_1, x_j]$.

7 **end for**

Let $\hat{f}_{\rho,k} = \sum_{l=1}^{t_\rho} \alpha_{\rho,kl} \tilde{M}_{\rho,l}(x_1, x_j) \in \mathbb{Z}_p[x_1, x_j]$ for $1 \le k \le s$, for $1 \le \rho \le r$ $(t_\rho = \#\hat{f}_{\rho,k})$.

for ρ from 1 to r **do**

  for l from 1 to t_ρ **do**

    $i \leftarrow \deg(\tilde{M}_{\rho,l}, x_1)$.

    Solve the linear system for $c_{\rho,lk}$: $\left\{ \sum_{k=1}^{s_{\rho,i}} m_{\rho,ik}^t c_{\rho,lk} = \alpha_{\rho,tl} \text{ for } 1 \le t \le s_{\rho,i} \right\}$.

  end for .... $\mathcal{O}(s\tilde{d}_j(\sum_{\rho=1}^{r} \#\hat{f}_{\rho,j-1}))$

  Construct $\hat{f}_{\rho,j} \leftarrow \sum_{l=1}^{t_\rho} \left(\sum_{k=1}^{s_{\rho,i}} c_{\rho,lk} M_{\rho,ik}(x_2, ..., x_{j-1}) \right) \tilde{M}_{\rho,l}(x_1, x_j)$.

end for

Pick $\boldsymbol{\beta} = (\beta_2, \cdots, \beta_j) \in \mathbb{Z}_p^{j-1}$ at random until $\deg(\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta})) = df_\rho$ for all $1 \le \rho \le r$.

$A_{\boldsymbol{\beta}} \leftarrow a(x_1, \boldsymbol{\beta}, \alpha_{j+1}, \ldots, \alpha_n) \in \mathbb{Z}_p[x_1]$ .... $\mathcal{O}(d_1)$ probes

if $\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta}) \mid A_{\boldsymbol{\beta}}$ for all $1 \le \rho \le r$ **then return** $(\hat{f}_{\rho,j}, 1 \le \rho \le r)$ **else return** FAIL **end if**

Our very first benchmark

n	10	11	12	13	14	15	16
CMBBSHL	5.790	13.430	50.855	154.441	722.310	1967.725	17,212.991
# probes	109,139	267,465	894,358	2,180,399	6,981,462	17,175,949	53,416,615
Det minor	0.306	1.754	8.429	49.080	315.842	$>$ 72gb	N/A
Gentleman	0.67	3.52	10.41	57.99	339.77	2058.20	N/A
Maple fac	1.91	3.48	23.11	57.75	509.82	7334.50	N/A
Maple tot	2.22	5.23	31.54	106.83	825.66	9392.70	-
Magma det	1.89	5.10	36.12	327.79	2108.42	$>$ 72gb	N/A
Magma fac	1.21	7.58	158.97	583.39	13,640.79	$>$ 72gb	N/A
Magma tot	3.10	12.68	195.09	911.18	15,749.21	-	-

Table: CPU timings in seconds for factorin $\det(T_n)$. N/A: Not attempted.

Our very first benchmark

n	10	11	12	13	14	15	16
H.L. x_n total	1.045	1.819	9.256	20.785	143.883	266.496	4182.20
s (H.L. x_n)	522	814	3174	5223	19,960	34,081	127,690
BB	0.137	0.240	1.304	3.043	11.363	20.350	109.59
Interp2var	0.046	0.081	0.307	0.631	2.172	3.469	17.19
Eval $f_{\rho,j-1}$	0.153	0.262	1.327	2.931	21.158	41.056	683.224
BHL	0.106	0.180	0.754	1.678	5.200	8.238	51.35
VSolve	0.058	0.101	1.937	4.219	72.887	143.183	2903.87

Table: Breakdown of CPU timings in seconds for Hensel lifting the last variable x_n.

We used Richard Zippel's $O(s^2)$ time $O(s)$ space Vandermonde solver from

Interpolating Polynomials from their Values. *J. Symb. Cmpt.* **9**:375–403, 1990.

Fast Vandermonde Solver

We implemented Erich Kaltofen and Lakshamn Yagati's $O(M(s) \log s)$ time $O(s \log s)$ space Vandermonde solver from
Improved sparse multivariate polynomial interpolation algorithms.
Proc ISSAC '88, LNCS **358**:467–474, Springer, 1989.

n	10	11	12	13	14	15	16
H.L. x_n total	1.309	2.162	7.129	12.663	64.635	126.665	1041.96
t_n	522	814	3174	5223	19,960	34,081	127,69
BB	0.195	0.394	1.031	2.046	9.152	18.496	80.85
Interp2var	0.024	0.033	0.149	0.254	0.981	1.764	10.05
Eval $f_{i,j-1}$	0.061	0.099	0.634	1.269	14.709	32.935	508.66
BHL	0.578	0.992	3.455	6.234	24.352	45.136	240.10
VSolve	0.330	0.453	1.243	1.773	10.594	19.547	165.37

Table: Breakdown of CPU timings in seconds for Hensel lifting the last variable x_n.

Fast Vandermonde Solver

n	10	11	12	13	14	15	16
CMBBSHL	6.299	14.679	43.927	106.838	403.089	1020.001	4876.827
# probes	109,139	267,465	894,358	2,180,399	6,981,462	17,175,949	53,416,615
Maple det	0.306	1.754	8.429	49.080	315.842	$>$ 72gb	N/A
Maple fac	1.91	3.48	23.11	57.75	509.82	7334.50	N/A
Maple tot	2.22	5.23	31.54	106.83	825.66	-	-
Magma det	1.89	5.10	36.12	327.79	2108.42	$>$ 72gb	N/A
Magma fac	1.21	7.58	158.97	583.39	13,640.79	$>$ 72gb	N/A
Magma tot	3.10	12.68	195.09	911.18	15,749.21	-	-

Table: CPU timings in seconds for computing $\det(T_n)$ using the fast Vandermonde solver. N/A: Not attempted.

Our new benchmark: $N = 17, 18, 19$

n	15	16	17	18	19
CMBBSHL	623.538 (10.39min)	3321.708 (0.89h)	8940.541 (2.48h)	68962.758 (19.15h)	347216.840 (96.45h)
# probes	17,178,578	53,419,850	131,362,184	399,884,433	-
CMBBSHL (old)	1020.001	4876.827	-	-	-
# probes (old)	17,175,949	53,416,615	-	-	-
Maple det	> 72gb	N/A	-	-	-
Maple fac	7334.50	N/A	-	-	-
Maple tot	-	-	-	-	-

Table: CPU timings in seconds for computing $\det(T_n)$ using the fast Vandermonde solver.

Our new benchmark: $N = 17, 18, 19$

n	15	16	17	18	19
H.L. x_n total	150.434	767.778	2332.31	22981.94	93792.43
t_n	34081	127,690	222842	821851	1457184
BB	61.493	33.768	66.888	293.16	557.32
Interp2var	0.421	1.494	6.558	13.275	25.09
Eval $f_{i,j-1}$	14.656	227.538	552.584	9396.91	23078.06
BHL	9.444	71.401	106.447	437.056	1923.67
VSolve	23.86	214.490	649.404	8156.58	50105.98

Table: Breakdown of CPU timings in seconds for Hensel lifting the last variable x_n.

Computing the content recursively

Consider

$$a = (y+1)(x+y)^2(x-y+2)$$

We compute $g = \gcd(a, \partial a/\partial x) = (y+1)(x+y)$ and $s = a/g = (x+y)(x-y+2)$.
Since $\mathrm{cont}(a,x) = y+1$ vanishes from s, our algorithm computes the factors $x+y$ and $x-y+2$ only.

```
> F := proc(alpha::list,p::prime)
>     Eval( (x+y)^2*(x-y+2), {x=alpha[1],y=alpha[2]} ) mod p
> end:
> MakeCont := proc( A::procedure, F::procedure, p::prime )
>     local gamma := rand(p)();
>     proc( alpha::list, p::prime )
>         alphaNew := subsop(1=gamma,alpha);
>         f := A( alphaNew, p );
>         g := F( alphaNew, p );
>         if g = 0 then return FAIL; fi;
>         f/g mod p;
>     end;
> end;
```

Factoring Vandermonde Matrices

The n by n Vandermonde matrix Vn is defined by $V_{nij} = x_i^{j-1}$ for $1 \le i \le n$, $1 \le j \le n$. The factorization of $\det(V_n)$ is $\prod_{1 \le i < j \le n}(x_j - x_i)$. For example

$$V_3 = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \qquad \begin{aligned} \det(V_3) &= -x_1^2x_2 + x_1^2x_3 + x_1x_2^2 - x_1x_3^2 - x_2^2x_3 + x_2x_3^2 \\ &= (x_3 - x_2)(x_3 - x_1)(x_2 - x_1). \end{aligned}$$

How far can we go in the sparse representation.

n	t	det	factor
6	720	0.015s	0.003s
7	5040	0.007s	0.017s
8	40320	0.020s	0.052s
9	362880	0.161s	0.615s
10	3628800	15.08s	17.91s
11	39916800	4.41m	10.00m
12	479001600		

Large Vandermonde matrices

Table: CPU timings (in seconds) for computing the factors of $\det(V_n)$ for larger n.

$n = N$	15	20	25	30	35	40
$r = \binom{n}{2}$	105	190	300	435	595	780
CMBBSHL tot	18.625	109.996	440.17	1376.793	3560.706	9057.977
probes tot	27311	85622	207912	429752	793809	1350786
$\mathrm{pp}(a)$ fac	0.791	2.246	5.891	13.968	29.597	57.745
H.L. x_n	0.055	0.117	0.256	0.467	0.800	1.487
probes x_n	465	820	1275	1830	2485	3240
s	1	1	1	1	1	1

Large Vandermonde matrices

Table: CPU timings (in seconds) for computing the factors of $\det(V_n)$ for larger n.

$n = N$	15	20	25	30	35	40
$r = \binom{n}{2}$	105	190	300	435	595	780
CMBBSHL tot	18.625	109.996	440.17	1376.793	3560.706	9057.977
probes tot	27311	85622	207912	429752	793809	1350786
$\mathrm{pp}(a)$ fac	0.791	2.246	5.891	13.968	29.597	57.745
H.L. x_n	0.055	0.117	0.256	0.467	0.800	1.487
probes x_n	465	820	1275	1830	2485	3240
s	1	1	1	1	1	1

Thank you!!