A New Black Box Factorization Algorithm - the Non-monic Case

Tian Chen and Michael Monagan

Department of Mathematics,
Simon Fraser University,
Canada

The sparse and black box representation of a polynomial

The **black box representation** of $f \in \mathbb{Z}[x_1, \cdots, x_n]$ is a program that accepts a prime p and an evaluation point $\boldsymbol{\alpha} \in \mathbb{Z}_p^n$ and outputs $f(\boldsymbol{\alpha})$ mod p.

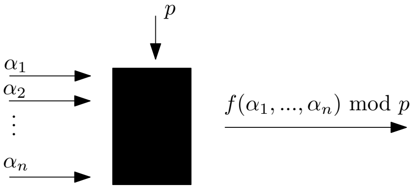


Figure: A modular black box for $f \in \mathbb{Z}[x_1, \cdots, x_n]$.

The sparse and black box representation of a polynomial

The **black box representation** of $f \in \mathbb{Z}[x_1, \cdots, x_n]$ is a program that accepts a prime p and an evaluation point $\boldsymbol{\alpha} \in \mathbb{Z}_p^n$ and outputs $f(\boldsymbol{\alpha}) \bmod p$.

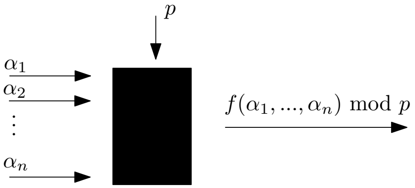


Figure: A modular black box for $f \in \mathbb{Z}[x_1, \cdots, x_n]$.

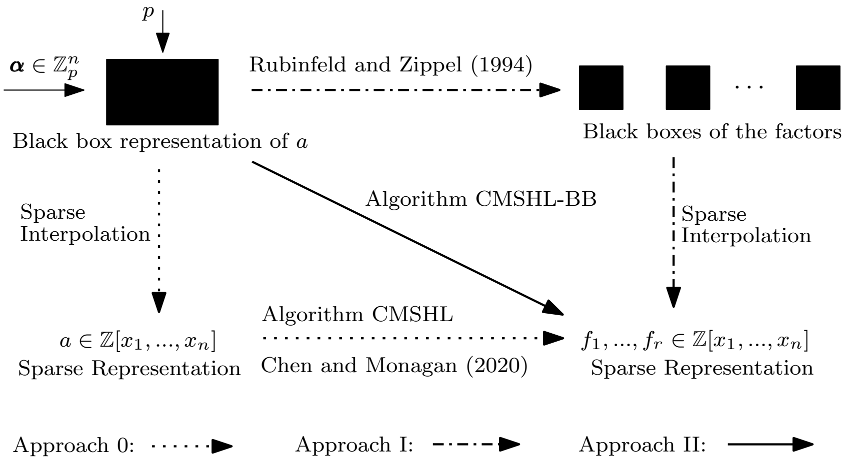
The **sparse representation** of $f \in \mathbb{Z}[x_1, \cdots, x_n]$ consists of a list of coefficients $c_k \in \mathbb{Z}, c_k \neq 0$ and exponents $(e_{k_1}, \cdots, e_{k_n}) \in \mathbb{N}^n$ such that

$$f = \sum_{k=1}^{t} c_k \cdot x_1^{e_{k_1}} \cdots x_n^{e_{k_n}},$$

where t is the number of non-zero terms of f.

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

Given a polynomial $a \in \mathbb{Z}[x_1, \cdots, x_n]$ represented by a black box, we aim to compute its factors in the sparse representation.



Previous work on multivariate polynomial factorization

Sparse Hensel lifting

- Yun (1974), Wang (1975), (1978): Multivariate Hensel lifting (MHL). Recovers the factors one variable at a time. Solves MDP $\sigma_i g_{j-1} + \tau_i f_{j-1} = c_i$ for $\sigma_i, \tau_i \in \mathbb{Z}_p[x_1, ..., x_{j-1}]$ one variable at a time (can be exponential in n).
- Zippel (1981), Kaltofen (1985): Sparse Hensel lifting (SHL).
- Monagan and Tuncer (2016): MTSHL. Solves MDP by sparse interpolation.
- Monagan and Tuncer (2018): Use bivariate Hensel lifts to compute σ_i, τ_i.
- Chen and Monagan (2020): CMSHL. No expression swell, highly parallelizable. Dominating cost is evaluating the input polynomial –> consider black box

Previous work on multivariate polynomial factorization

Sparse Hensel lifting

- Yun (1974), Wang (1975), (1978): Multivariate Hensel lifting (MHL). Recovers the factors one variable at a time. Solves MDP $\sigma_i g_{j-1} + \tau_i f_{j-1} = c_i$ for $\sigma_i, \tau_i \in \mathbb{Z}_p[x_1, ..., x_{j-1}]$ one variable at a time (can be exponential in n).
- Zippel (1981), Kaltofen (1985): Sparse Hensel lifting (SHL).
- Monagan and Tuncer (2016): MTSHL. Solves MDP by sparse interpolation.
- Monagan and Tuncer (2018): Use bivariate Hensel lifts to compute σ_i, τ_i.
- Chen and Monagan (2020): CMSHL. No expression swell, highly parallelizable. Dominating cost is evaluating the input polynomial –> consider black box

Black box factorization

- Kaltofen and Trager (1990): Outputs black boxes of the factors.
- Rubinfeld and Zippel (1994): For factoring $a \in \mathbb{Z}[x_1, \cdots, x_n]$.
- Chen and Monagan (2022): Monic and square-free case only.

Previous work on multivariate polynomial factorization

Sparse Hensel lifting

- Yun (1974), Wang (1975), (1978): Multivariate Hensel lifting (MHL). Recovers the factors one variable at a time. Solves MDP $\sigma_i g_{j-1} + \tau_i f_{j-1} = c_i$ for $\sigma_i, \tau_i \in \mathbb{Z}_p[x_1, ..., x_{j-1}]$ one variable at a time (can be exponential in n).
- Zippel (1981), Kaltofen (1985): Sparse Hensel lifting (SHL).
- Monagan and Tuncer (2016): MTSHL. Solves MDP by sparse interpolation.
- Monagan and Tuncer (2018): Use bivariate Hensel lifts to compute σ_i, τ_i.
- Chen and Monagan (2020): CMSHL. No expression swell, highly parallelizable. Dominating cost is evaluating the input polynomial –> consider black box

Black box factorization

- Kaltofen and Trager (1990): Outputs black boxes of the factors.
- Rubinfeld and Zippel (1994): For factoring $a \in \mathbb{Z}[x_1, \cdots, x_n]$.
- Chen and Monagan (2022): Monic and square-free case only.

Other work

- Huang and Gao (2023): Non-sparse Hensel lifting
- Lecerf (2007): Dense multivariate polynomial factorization

Our Contributions

A new black box factorization algorithm CMSHL-BB:

- Accepts all cases of input polynomials, i.e. non-monic, non-square-free and non-primitive cases.
- A Maple + C hybrid implementation with timing benchmarks.
- A worst case complexity analysis with failure probabilities (Monte Carlo).

Example 1: Computing the determinant of a Toeplitz matrix

Let T_n be an $n \times n$ symmetric 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	$\#\det(T_n)$	$\#f_i$	s
8	1628	167, 167	38
9	6090	294, 153	50
10	23797	931, 931	229
11	90296	1730, 849	337
12	350726	5579, 5579	1465
13	1338076	10611, 4983	2297
14	5165957	34937, 34937	9705
15	19732508	66684, 30458	34081
16	–	221854, 221854	127690

Table: Number of terms of $\det(T_n)$ and its factors. s is the maximum number of bivariate images [Chen and Monagan (2022)].

Algorithm CMSHL-BB (Approach II):

- Space efficient since $\#f_i \ll \#\det(T_n)$.
- Less probes to the black box than Rubinfeld and Zippel's algorithm since $s < \#f_{max}$.

Example 2: Non-monic case

$$B = \begin{bmatrix} uvw & v & uvw+v+w & \dots & uvw+v \\ v & uvw & uvw+2v & \dots & uvw+v \\ w & v & uvw+v+w & \dots & v+w \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w & v & uvw+v+w & \dots & 2vwx+2ux+3v+4w \end{bmatrix}.$$

$$\begin{aligned} a = \det(B) = &-(-v^2w^2x^2 + uvwx^2 + vw^2x - uwx + v^2 - 2vw + w^2) \\ &(v^2w^2x^2 + uvwx^2 + vw^2x + uwx - v^2 - 2vw - w^2) \\ &(u^2v^2w^2 + u^2vwx + uv^2w + uvx - v^2 - 2vw - w^2) \\ &(u^2v^2w^2 - u^2vwx - uv^2w + uvx - v^2 + 2vw - w^2). \end{aligned}$$

$\#\text{expand}(a) = 120.$

$$\begin{aligned} \text{lcoeff}(a,u) &= -v^6w^6x^4 + v^4w^4x^6 + v^4w^6x^2 - v^2w^4x^4, \\ \text{lcoeff}(a,v) &= u^4w^8x^4 - 2u^4w^6x^2 - 3u^2w^6x^4 + u^4w^4 + 6u^2w^4x^2 \\ &\quad + w^4x^4 - 3u^2w^2 - 2w^2x^2 + 1. \end{aligned}$$

Technicalities for our algorithm design

Algorithm CMSHL-BB:

1. Non-monic: Use **non-monic** bivariate Hensel lifts (BHL), modified from Monagan and Paluck (2022). Cubic cost: $O(d_1^2 d_j + d_1 d_j^2)$.
2. Non-square-free: Compute the **square-free part** of the bivariate images of $a(x_1, \beta^k, x_j)$ with dense interpolation, gcd computation and division. Cost of gcd: $O(d_1^2 d_j + d_1 d_j^2)$ [Brown (1971)].
3. Non-primitive: Compute the content recursively after recovering the primitive factors.

How does our algorithm work?

Prior Hensel lifting steps:

1. Choose a large prime p, e.g. $p = 2^{62} - 57$ and a positive integer $\tilde{N} < p$.
2. Choose $\boldsymbol{\alpha} = (\alpha_2, \cdots, \alpha_n) \in \mathbb{Z}^{n-1}$ from $[1, \tilde{N} - 1]^{n-1}$ randomly s.t.
   - $\boldsymbol{\alpha}$ is Hilbertian
   - $\boldsymbol{\alpha}$ satisfies the weak SHL assumption (Lemma 2.2)
3. Compute $a(x_1, \boldsymbol{\alpha})$ from the black box **B** with Chinese remaindering.
4. Factor $a(x_1, \boldsymbol{\alpha})$ over $\mathbb{Z}$ as follows.
   Let the factorization of a over $\mathbb{Z}$ be of the form

$$a = h f_1^{e_1} f_2^{e_2} \cdots f_r^{e_r} \in \mathbb{Z}[x_1, \cdots, x_n]$$

where $\deg(f_\rho, x_1) > 0$ $(1 \leq \rho \leq r)$, f_ρ is irreducible over $\mathbb{Z}$ and h is the content of a in x_1. Then, with high probability (w.h.p.),

$$a(x_1, \boldsymbol{\alpha}) = \hat{h} \hat{f}_1^{\,e_1} \hat{f}_2^{\,e_2} \cdots \hat{f}_r^{\,e_r} \in \mathbb{Z}[x_1],$$

where $\boxed{\hat{f}_\rho(x_1, \boldsymbol{\alpha}) = (1/\lambda_\rho) f_\rho(x_1, \boldsymbol{\alpha})}$ for some $\lambda_\rho \in \mathbb{Z}$ and $\hat{f}_\rho$ is irreducible in $\mathbb{Z}[x_1]$ $(1 \leq \rho \leq r)$.

How does our algorithm work?

Definition

The **square-free part** of a is defined as

$$\mathrm{sqf}(a) := f_1 f_2 \cdots f_r = \frac{a}{\gcd(a, \partial a/\partial x_1)}.$$

Let $\hat{f}_{\rho,1} := \hat{f}_\rho(x_1, \boldsymbol{\alpha}) \bmod p$ and **B** be the black box representation of a.

Algorithm CMSHL-BB (non-monic and non-square-free):

- Input: A prime p, the black box B, $\boldsymbol{\alpha} \in \mathbb{Z}^{n-1}$, $\deg(a, x_j)$ $(1 \le j \le n)$ (pre-computed), $\boxed{\hat{f}_{\rho,1} \in \mathbb{Z}_p[x_1]}$ $(1 \le \rho \le r)$ s.t.
  (i) $\gcd(\hat{f}_{k,1}, \hat{f}_{l,1}) = 1$ for $k \ne l$ in $\mathbb{Z}_p[x_1]$,
  (ii) $\mathrm{sqf}(a(x_1, \boldsymbol{\alpha})) = \prod_{\rho=1}^{r} \lambda_\rho \prod_{\rho=1}^{r} \hat{f}_{\rho,1} \bmod p$.
- Output: $\boxed{\hat{f}_{\rho,n} \in \mathbb{Z}_p[x_1, \cdots, x_n]}$ $(1 \le \rho \le r)$ s.t.
  $\boxed{\mathrm{sqf}(a(x_1, \cdots, x_n)) = \prod_{\rho=1}^{r} \lambda_\rho \prod_{\rho=1}^{r} \hat{f}_{\rho,n} \bmod p.}$ Or FAIL.

Finally, use rational number reconstruction to recover the integer coefficients to get $\boxed{f_\rho \in \mathbb{Z}[x_1, \cdots, x_n]}$ $(1 \le \rho \le r)$.

Algorithm CMSHL-BB: the j^{th} Hensel lifting step

Define $\hat{f}_{\rho,j} := \hat{f}_\rho(x_1, \cdots, x_j, \alpha_{j+1}, \cdots, \alpha_n) \bmod p$ for $2 \le j \le n$ (to be computed).

```
Let f̂_{ρ,j−1} = Σ_{i=0}^{df_ρ} σ_{ρ,i}(x_2, ..., x_{j−1}) x_1^i (for 1 ≤ ρ ≤ r)
where σ_{ρ,i} = Σ_{k=1}^{s_{ρ,i}} c_{ρ,ik} M_{ρ,ik} with M_{ρ,ik} the monomials in σ_{ρ,i} and df_ρ = deg(f̂_{ρ,j−1}, x_1).
Pick β = (β_2, ··· , β_{j−1}) ∈ Z_p^{j−2} at random.
Evaluate (for 1 ≤ ρ ≤ r): S_ρ = {S_{ρ,i} = {m_{ρ,ik} = M_{ρ,ik}(β), 1 ≤ k ≤ s_{ρ,i}}, 0 ≤ i ≤ df_ρ}.
if any |S_{ρ,i}| ≠ s_{ρ,i} then return FAIL end if
Let s be the maximum of s_{ρ,i}.
for k from 1 to s do
    Let Y_k = (x_2 = β_2^k, ··· , x_{j−1} = β_{j−1}^k).
    A_k ← a_j(x_1, Y_k, x_j) ∈ Z_p[x_1, x_j].  ................ O(s(d_1^2 d_j + d_1 d_j^2 + d_1 d_j C(probe B)))
    if deg(A_k, x_1) ≠ d_1 or deg(A_k, x_j) ≠ d_j then return FAIL end if
    g_k ← gcd(A_k, ∂A_k/∂x_1) mod p.  ................................ O(s(d_1^2 d_j + d_1 d_j^2))
    if deg(g_k, x_1) ≠ d_1 − Σ_{ρ=1}^{r} df_ρ then return FAIL end if
    A_sf ← quo(A_k, g_k) mod p.
    A_sfm ← A_sf/(lc(lc(A_sf, x_1), x_j)) mod p.
    F_{ρ,k} ← f̂_{ρ,j−1}(x_1, Y_k) ∈ Z_p[x_1] for 1 ≤ ρ ≤ r.
    if any deg(F_{ρ,k}) < df_ρ (for 1 ≤ ρ ≤ r) then return FAIL end if
    if gcd(F_{ρ,k}, F_{φ,k}) ≠ 1 for any ρ ≠ φ (1 ≤ ρ, φ ≤ r) then return FAIL end if
    f̂_{ρ,k} ← BivariateHenselLift(A_sfm(x_1, x_j), F_{ρ,k}(x_1), α_j, p).
    ...................................... O(s(d̃_1 d̃_j^2 + d̃_1^2 d̃_j)) ⊆ O(s(d_1 d_j^2 + d_1^2 d_j))
end for
```

Algorithm CMSHL: the j^{th} Hensel lifting step

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 $1 \le k \le s$
where $t_\rho = \#\hat{f}_{\rho,k}$ (for $1 \le \rho \le r$).

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 $\left\{ \sum_{k=1}^{s_{\rho,i}} m^t_{\rho,ik} c_{\rho,lk} = \alpha_{\rho,tl} \text{ for } 1 \le t \le s_{\rho,i} \right\}$ for $c_{\rho,lk}$.

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

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

end for

Pick $\boldsymbol{\beta} = (\beta_2, \cdots, \beta_j) \in \mathbb{Z}_p^{j-1}$ at random.

$A_{\boldsymbol{\beta}} \leftarrow \text{sqf}(a_j(x_1, \boldsymbol{\beta})) \bmod p$ // via probes to **B**, interpolation, and sqrfree compt.

if $\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta}) \mid A_{\boldsymbol{\beta}}$ **and** $\deg(\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta})) = df_\rho$ (for $1 \le \rho \le r$) **then**

  return $\hat{f}_{\rho,j}$ (for $1 \le \rho \le r$)

else return FAIL

end if

Non-monic bivariate Hensel lift (BHL)

Input: prime p, $\alpha \in \mathbb{Z}_p$, $a \in \mathbb{Z}_p[x, y]$, $\hat{f}_{\rho,0} \in \mathbb{Z}_p[x]$ for $1 \leq \rho \leq r$ s.t.

(i) a is primitive in x,

(ii) $a(y = \alpha) = \zeta \prod_{\rho=1}^{r} \hat{f}_{\rho,0}$, where $\zeta \in \mathbb{Z}_p$,

(iii) $\gcd(\hat{f}_{k,0}, \hat{f}_{l,0}) = 1$ for $k \neq l$.

Output: $\hat{f}_\rho \in \mathbb{Z}_p[x, y]$ for $1 \leq \rho \leq r$ s.t.

(i) $a = \zeta \prod_{\rho=1}^{r} \hat{f}_\rho$ and

(ii) $\hat{f}_\rho(y = \alpha) = \hat{f}_{\rho,0}$.

Otherwise, **FAIL**.

BHL is called in Step 17 in CMSHL:

Input: p, α_j, $\mathrm{sqf}(a_j(x_1, Y_k, x_j))$, $\hat{f}_{\rho,j-1}(x_1, Y_k)$ for $1 \leq \rho \leq r$ s.t.

$\mathrm{sqf}(a_j(x_1, x_j = \alpha_j)) = (\prod_{\rho=1}^{r} \lambda_\rho) \prod_{\rho=1}^{r} \hat{f}_{\rho,j-1}(x_1)$.

Output: $\hat{f}_{\rho,j}(x_1, x_j)$ for $1 \leq \rho \leq r$ s.t.

$\mathrm{sqf}(a_j(x_1, x_j)) = (\prod_{\rho=1}^{r} \lambda_\rho) \prod_{\rho=1}^{r} \hat{f}_{\rho,j}(x_1, x_j)$ and

$\hat{f}_{\rho,j}(x_1, x_j = \alpha_j) = \hat{f}_{\rho,j-1}(x_1)$.

Example

Example

Consider $a = f_1 f_2 \in \mathbb{Z}[x_1, \cdots, x_4]$ where

$$f_1 = (2x_2^2x_3^3 + 4)x_1^8 + (4x_2^2x_3^3 + 22x_2^2x_4^3 + 1452x_2^2x_4)x_1 + x_2^2x_3x_4 - 4x_3,$$
$$f_2 = (3x_2 + 39x_4 + 3x_3)x_1^8 + (5x_2x_3^2x_4 + 33x_2x_3x_4^2)x_1^2 - 363x_4^2 + 44.$$

In this case, $h = 1$ (a has no content in x_1, neither integer content) and $\mathrm{sqf}(a) = a$. Let $\boldsymbol{\alpha} = (2, 3, 9)$ and $p = 2^{31} - 1$,

$$\begin{aligned} a(x_1, \boldsymbol{\alpha}) &= 80520x_1^{16} + 3706560x_1^{10} + \cdots - 3430775304x_1 - 2818464 \\ &= \underbrace{4}_{\lambda_1}\underbrace{(55x_1^8 + 29214x_1 + 24)}_{\hat{f}_1}\underbrace{(366x_1^8 + 16848x_1^2 - 29359)}_{\hat{f}_2} \\ &= f_1(x_1, \boldsymbol{\alpha}) f_2(x_1, \boldsymbol{\alpha}). \end{aligned}$$

Example ctd..

After the 1st Hensel lifting step (a bivariate Hensel lift only),

$\hat{f}_{1,2} = (1073741837x_2^2 + 1)x_1^8 + 1073749127x_2^2x_1 + 1610612742x_2^2 + 2147483644,$

$\hat{f}_{2,2} = (3x_2 + 360)x_1^8 + 8424x_2x_1^2 + 2147454288.$

After the 2nd Hensel lifting step,

$$\hat{f}_{1,3} = (1073741824x_2^2x_3^3 + 1)x_1^8 + (x_2^2x_3^3 + 1073749100x_2^2)x_1 + 536870914x_2^2x_3 + 2147483646x_3,$$

$$\hat{f}_{2,3} = (3x_2 + 3x_3 + 351)x_1^8 + (45x_2x_3^2 + 2673x_2x_3)x_1^2 + 2147454288.$$

The last Hensel lifting step outputs $\hat{f}_{\rho,4}$ $(\rho = 1, 2)$ s.t.

$a_4 = \text{sqf}(a_4) = (\lambda_1\lambda_2)\hat{f}_{1,4}\hat{f}_{2,4} \bmod p$ with

$$\hat{f}_{1,4} = (1073741824x_2^2x_3^3 + 1)x_1^8 + (x_2^2x_3^3 + 1073741829x_2^2x_4^3 + 363x_2^2x_4)x_1 + 536870912x_2^2x_3x_4 + 2147483646x_3$$

$$\hat{f}_{2,4} = (3x_2 + 39x_4 + 3x_3)x_1^8 + (5x_2x_3^2x_4 + 33x_2x_3x_4^2)x_1^2 + 2147483284x_4^2 + 44.$$

Example ctd..

Rational number reconstruction in Maple:

```
> ff[1] := iratrecon(fhat[1,4], p);
```

$$\mathrm{ff}_1 := \frac{1}{2}x_2^2x_1^8x_3^3 + x_1^8 + x_1x_2^2x_3^3 + \frac{11}{2}x_2^2x_1x_4^3 + 363x_2^2x_1x_4 + \frac{1}{4}x_2^2x_3x_4 - x_3$$

λ_1 is the least common multiple of the denominators of coefficients of ff_1.
Multiply ff_1 by $\lambda_1 = 4$, we get the true factor $f_1 \in \mathbb{Z}[x_1, \cdots, x_n]$:

```
> f[1] := numer(ff[1]);
```

$$\mathrm{f}_1 := 2x_2^2x_1^8x_3^3 + 4x_1^8 + 4x_1x_2^2x_3^3 + 22x_2^2x_1x_4^3 + 1452x_2^2x_1x_4 + x_2^2x_3x_4 - 4x_3$$

A Hybrid Maple + C Implementation of Method II

CPU timings (in seconds) for our algorithm, compared with Maple and Magma's current best determinant and factorization algorithms.

n	4	5	6	7	8	9
$N = 2n$	8	10	12	14	16	18
$\# f_i$	7,7 7,7	12,7 12,7	32,32 32,32	56,30 56,30	167,167 167,167	153,294 153,294
$\#\det(A)$	120	701	5162	79740	1716810	7490224
CMSHL total	0.092	0.257	0.972	3.618	19.677	40.219
total probes	721	2112	6453	19584	85189	145065
Maple det	0.057	0.455	7.880	382.80	$>$ 64 gigs	-
Maple factor	0.140	0.109	0.326	1.270	-	-
Maple total	0.197	0.564	8.206	384.07	-	-
Magma det	0.140	1.680	6.290	594.60	$>$ 3h	-
Magma factor	0.800	0.120	0.480	33.140	-	-
Magma total	0.940	1.800	6.770	627.74	-	-

More Timings for Large Matrices

Table: Timings (in seconds) for computing determinants of large matrices.

	heron3d	heron4d	robotarms	heron5d
n	7	11	8	16
$N \times N$	13×13	63×63	20×20	399×399
r	6	4	3	8
$\#f_i$	3,23,3,3,1,3	22,1,6,131	2124,4,7	823,130,22,3,3,3,3,1
e_i	1,2,1,1,7,1	2,37,7,4	1,4,4	8,8,20,46,46,46,1831
$\#\det(A)$	525	37666243	178053	-
$\max \lambda_\rho$	1	1	169	1
CMSHL tot	1.096	81.376	1083.335	155054.324
probes tot	8560	339840	540834	36008392
Maple det	0.614	O/M	N/A	N/A
Maple fac	0.084	O/M	N/A	N/A
Maple tot	0.698	-	-	-

$$\det(A_{heron3d}) = 64as^7(as-bs+cs)(as-bs-cs)(as+bs+cs)(as+bs-cs)\underbrace{(as^4es^2+as^2bs^2cs^2-\cdots-cs^2es^2fs^2+144vo^2)^2}_{23\text{ terms}}$$

More Timings for Large Matrices

Table: Breakdown of timings (in seconds) at H.L. x_n.

	heron3d	heron4d	robotarms	heron5d
n	7	11	8	16
$N \times N$	13×13	63×63	20×20	399×399
H.L. x_n tot	0.229	16.612	441.593	10361.995
s	13	85	806	571
BB tot	0.046	12.801	421.366	9940.302
BB eval	0.028	5.428	415.676	4809.717
BB det	0.011	6.507	7.193	5087.231
Eval $\hat{f}_{\rho,j-1}$	0.011	0.132	0.374	0.467
BHL	0.005	0.023	0.298	0.196
VSolve	0.003	0.001	0.333	0.021

Complexity

Theorem

(Theorem 4.3) Let p *be a large prime and* $\tilde{N} < p$, $\tilde{N} \in \mathbb{Z}^+$. *Let* $a \in \mathbb{Z}[x_1, \cdots, x_n]$ *and* $\boldsymbol{\alpha} = (\alpha_2, \cdots, \alpha_n) \in \mathbb{Z}_p^{n-1}$ *be randomly chosen such that* $0 < \alpha_i < \tilde{N}$. *Suppose* $\boldsymbol{\alpha}$ *is Hilbertian and condition (i) of the input of CMSHL is satisfied. Then, with a high probability of success, the total number of arithmetic operations in* $\mathbb{Z}_p$ *in the worst case for lifting* $\hat{f}_{\rho,1}$ *to* $\hat{f}_{\rho,n}$ *using Algorithm CMSHL in* $n-1$ *steps is*

$$O(nd_1 d_{\max} s_{\max} C(probe\ \boldsymbol{B})) + O\left((n-2)s_{\max} d_{\max}\left(\sum_{\rho=1}^{r} \#\hat{f}_{\rho,j-1} + d_1^2 + d_1 d_{\max}\right)\right) \tag{1}$$

where $d_{\max} = \max_{3 \le i \le n}(\deg(sqf(a), x_i))$, $s_{\max} = \max(s_j)$ *where* s_j *is the number* s *defined at step 7 of Algorithm 1 and* $C(probe\ \boldsymbol{B})$ *is the cost of one probe to the black box* $\boldsymbol{B}$. *The total number of probes to the black box is* $O(nd_1 d_{\max} s_{\max})$.

Future Work

- Large integers
- Multi-point evaluations to speed up
- Parallelization

Thank you for attending!

References

W. S. Brown. On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *J. ACM* **18**:478–504, 1971.

T. Chen and M. Monagan. The complexity and parallel implementation of two sparse multivariate Hensel lifting algorithms for polynomial factorization. In Proceedings of CASC 2020, LNCS **12291**, 150–169. Springer (2020)

T. Chen and M. Monagan. Factoring multivariate polynomials represented by black boxes: A Maple + C Implementation. *Math. Comput. Sci.* **16**,18 (2022)

Q-L. Huang and X-S Gao. New sparse multivariate polynomial factorization algorithms over integers. In Proceedings of ISSAC 2023. ACM (2023)

E. Kaltofen. Sparse Hensel lifting. In Proceedings of EUROCAL '85, LNCS **204**, 4–17. Springer (1985)

E. Kaltofen and B. 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)

G. Lecerf. Improved dense multivariate polynomial factorization algorithms. *J. Symbolic Computation* **42**:477–494 (2007)

M. Monagan and B. Tuncer. Using Sparse Interpolation in Hensel Lifting. In Proceedings of CASC 2016, LNCS **9890**, 381–400. Springer (2016)

M. Monagan and B. Tuncer. Sparse multivariate Hensel lifting: a high-performance design and implementation. In Proceedings of ICMS 2018, LNCS **10931**, 359–368. Springer (2018)

M. Monagan and G. Paluck. Linear Hensel lifting for $\mathbb{Z}_p[x, y]$ for n factors with cubic cost. In Proceedings of ISSAC 2022, 169-166. ACM (2022)

R. Rubinfeld and R. E. Zippel. A new modular interpolation algorithm for factoring multivariate polynomials. In Proceedings of Algorithmic Number Theory, First International Symposium, ANTS-I (1994)

P. S. Wang and L. P. Rothschild. Factoring multivariate polynomials over the integers. *Math. Comp.* **29**, 935–950 (1975)

P. S. Wang. An improved multivariate polynomial factoring algorithm. *Math. Comp.* **32**, 1215–1231 (1978)

D. Y. Y. Yun. The Hensel Lemma in algebraic manipulation. Ph.D. Thesis (1974)

R. E. Zippel. Newton's iteration and the sparse Hensel algorithm. In Proceedings of the ACM Symposium on Symbolic Algebraic Computation, pp. 68–72 (1981)

Computing the content

After recovering $f_1, \cdots, f_r \in \mathbb{Z}[x_1, \cdots, x_n]$, we create another black box **C** for the content. Then the content is factored recursively.

- Let $F = f_1^{e_1} \cdots f_r^{e_r} \in \mathbb{Z}[x_1, \cdots, x_n]$.
- MakeCont := proc(B, F, X, p)

```
    local alpha1 := rand(p)();
    proc( Y, alpha, p ) nY := nops(Y);
        alphaF[1] := alpha1;
        for i to nY do alphaF[i+1] := alpha[i]; od;
        Feval := Eval(F, [X[1]=alpha1, seq(Y[i]=alpha[i], i=1..nY)])
mod p;
        if Feval = 0 then return FAIL; fi;
        c := B( [X[1],op(Y)], alphaF, p ) / Feval mod p;
    end;
end;
```

- C := MakeCont(B, F, X, p);