Record-Setting Toeplitz Determinant Computation via Black Box Factorization

Tian Chen

Department of Mathematics,
Simon Fraser University, Canada

Multivariate Polynomial Factorization

Problem $\mathcal{P}$: 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., $6x^2 - 6y^2 = 6(x + y)(x - y)$.

A Brief History of Multivariate polynomial factorization

Hensel lifting

- 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): Sparse Hensel lifting (SHL).
- Monagan and Tuncer (2016), (2018): MTSHL.
- Chen and Monagan (2020): CMSHL. No expression swell, no multivariate polynomial arithmetic, highly parallelizable.
  Dominating cost is evaluating the input polynomial
  –> black box representation

A Brief History of Multivariate 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.
- 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 significantly fewer probes to the black box than Rubinfeld and Zippel's algorithm.

Sparse polynomials

Definition (Monagan [2])

Let $a \in \mathbb{Z}[x_1, \cdots, x_n]$ and let $d = \deg(a)$ be the total degree of a. Let $T = \binom{n+d}{d}$, which is the maximum possible number of terms in a. Let $\#a$ denote the number of nonzero terms of a. We say a is **sparse** if $\#a \leq \sqrt{T}$; otherwise, a is **dense**.

Example

$a = 21x_1^5 + 4x_1^3x_2x_4 - 3x_1^3x_4 + 7x_1x_2^2x_3 + 20x_2^3x_3x_4$ is sparse. In this case, $n = 4$, $d = 5$, $T = \binom{n+d}{d} = 126$. $\sqrt{T} \approx 11.225$ and $\#a = 5 < \sqrt{T}$.

Definition

Let $a \in \mathbb{Z}[x_1, \ldots, x_n]$ be nonzero with total degree $d = \deg(a)$, and let $T = \binom{n+d}{d}$. We define the **sparsity ratio** of a as

$$\mathrm{sp}(a) := \frac{\#a}{T} \in (0, 1].$$

For a fixed $\theta \in (0, 1]$, we call the polynomial a θ-**sparse** if $\mathrm{sp}(a) \leq \theta$.

Sparse representation of a polynomial

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

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

Example.

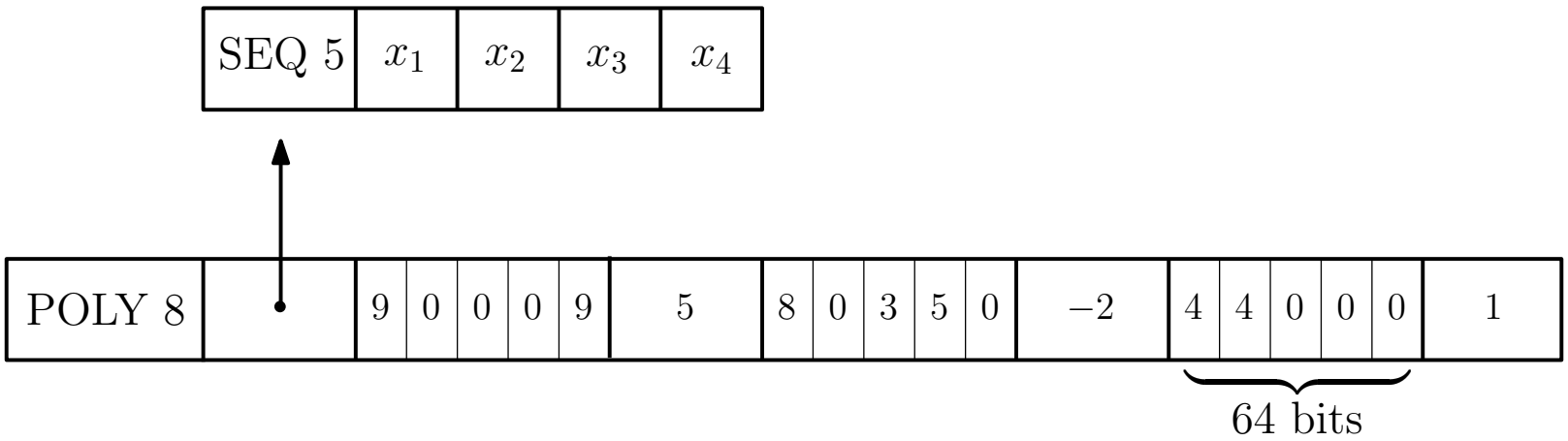


Figure: Maple's POLY data structure for $a = 5x_4^9 - 2x_2^3x_3^5 + x_1^4$.

Black box representation of a polynomial

Definition

A *modular* **black box representation** of $a \in \mathbb{Z}[x_1, \cdots, x_n]$ is a computer program $\mathsf{B} : \mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$ that on input $\boldsymbol{\alpha} \in \mathbb{Z}^n$ and a prime p outputs $\mathsf{B}(\boldsymbol{\alpha}, p) = a(\boldsymbol{\alpha}) \bmod p$ (see Figure 2).

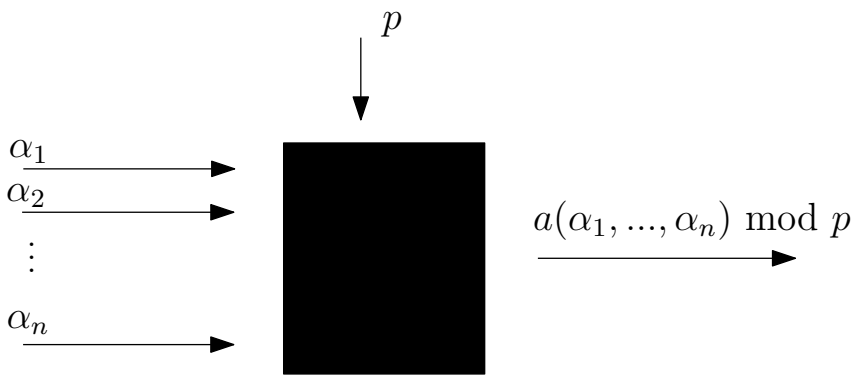


Figure: A modular black box representation of $a \in \mathbb{Z}[x_1, ..., x_n]$.

Example: Black box for the determinant of a Toeplitz matrix

Let T_n be an $n \times n$ symmetric Toeplitz matrix. The modular black box representation of $\det(T_n)$ can be coded in Maple as a procedure:

```
B := proc( alpha::{Array,list}, p::prime )
    local n := numlems(alpha), i,j,Tn;
    Tn := Matrix(n,n);
    for i to n do
        for j to n do
            Tn[i,j] := alpha[abs(i-j)+1];
        od;
    od;
    return Det(Tn) mod p;
end:
> B ( [1,2], 7 );
```

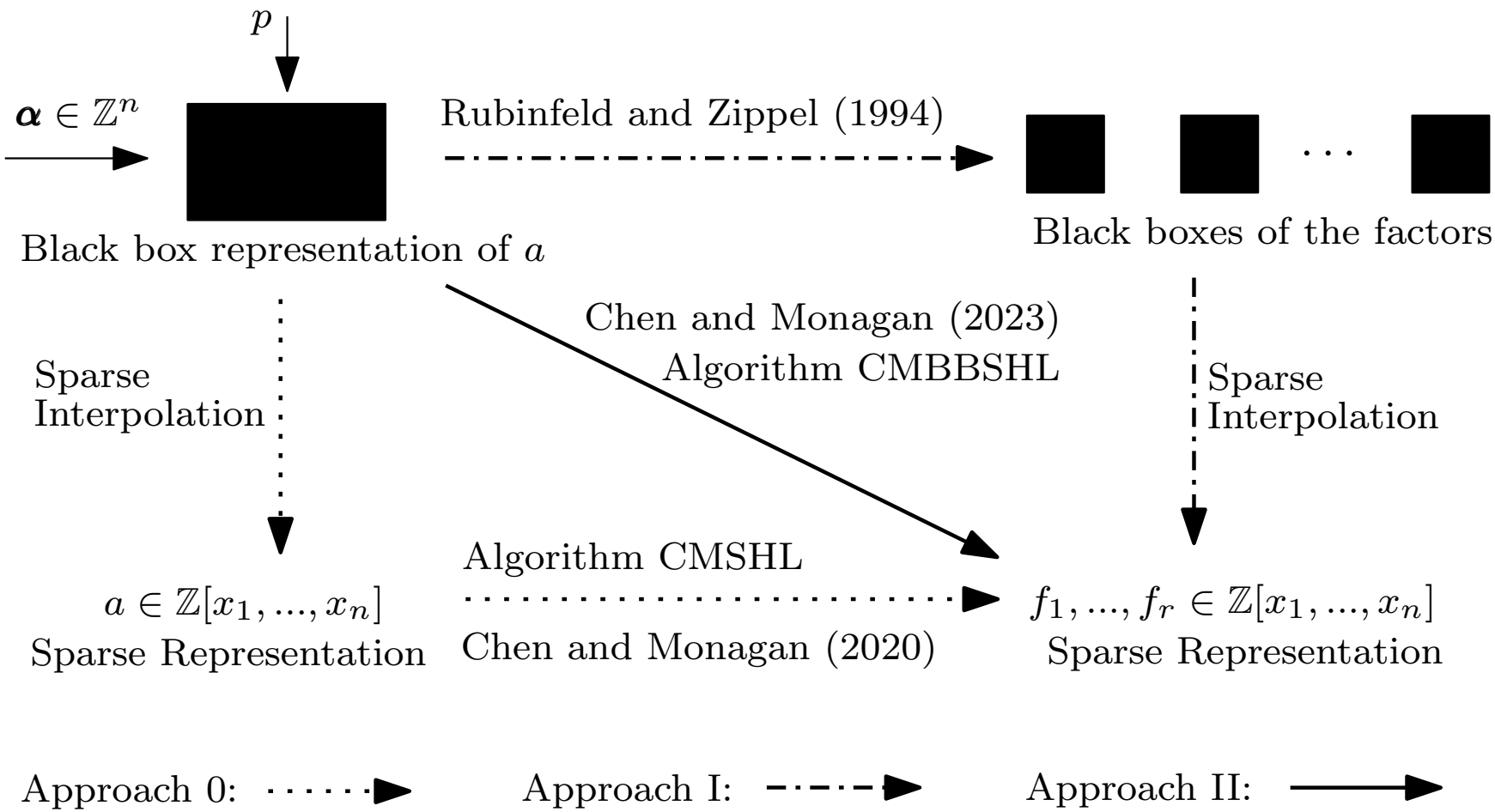
4

Multivariate Polynomial Factorization

Problem $\mathcal{P}2$: Given a polynomial $a \in \mathbb{Z}[x_1, \cdots, x_n]$ represented by a *black box* $\mathsf{BB} : \mathbb{Z}^n \to \mathbb{Z}$ or a *modular black box* $\mathsf{B} : \mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$, compute its irreducible factors in $\mathbb{Z}[x_1, \cdots, x_n]$ in their *sparse representation*, but do not factor the integer content.

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.



Past Contributions on Black Box Factorization

1. T. Chen and M. Monagan. Factoring multivariate polynomials represented by black boxes: A Maple + C Implementation. *Math. Comput. Sci.* **16**,18 (2022) (Input polynomial is monic and square-free; Toeplitz N=16.)
2. T. Chen and M. Monagan. A new black box factorization algorithm - the non-monic case. In Proceedings of ISSAC 2023, pp. 173–181. ACM (2023)
3. T. Chen. Sparse Hensel lifting algorithms for multivariate polynomial factorization. PhD Thesis. Simon Fraser University (2024)
4. T. Chen and M. Monagan. A Maple program to factor multivariate polynomials given by black boxes. ACM Communications in Computer Algebra 58(3):77-80. ACM (2025)

Computing 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	–	221854, 221854
17	–	191164, 424292
18	–	1419659, 1419659
19	–	1209612, 2714726

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

Algorithm CMBBSHL (Approach II):

- Space efficient since $\#f_i \ll \#\det(T_n)$.
- Fewer probes to the black box than Rubinfeld and Zippel's algorithm.

Example (Computing the factors of $\det(T_4)$)

- Let $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.

Example (Computing the factors of $\det(T_4)$)

- Let $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.
- $a(x_1, \boldsymbol{\alpha}) = x_1^4 - 93x_1^2 + 420x_1 - 416 = (x_1^2 - 7x_1 + 8)(x_1^2 + 7x_1 - 52) \in \mathbb{Z}[x_1]$.

Example (Computing the factors of $\det(T_4)$)

- Let $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.
- $a(x_1, \boldsymbol{\alpha}) = x_1^4 - 93x_1^2 + 420x_1 - 416 = (x_1^2 - 7x_1 + 8)(x_1^2 + 7x_1 - 52) \in \mathbb{Z}[x_1]$.
- The first Hensel lifting step recovers x_2 and we get

$$f_2 = x_1^2 - x_1x_2 - x_2^2 - 4x_1 + 14x_2 - 25,$$
$$g_2 = x_1^2 + x_1x_2 - x_2^2 + 4x_1 - 6x_2 - 25.$$

Example (Computing the factors of $\det(T_4)$)

- Let $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.
- $a(x_1, \boldsymbol{\alpha}) = x_1^4 - 93x_1^2 + 420x_1 - 416 = (x_1^2 - 7x_1 + 8)(x_1^2 + 7x_1 - 52) \in \mathbb{Z}[x_1]$.
- The first Hensel lifting step recovers x_2 and we get

$$f_2 = x_1^2 - x_1x_2 - x_2^2 - 4x_1 + 14x_2 - 25,$$
$$g_2 = x_1^2 + x_1x_2 - x_2^2 + 4x_1 - 6x_2 - 25.$$

- The second Hensel lifting step recovers x_3 and

$$f_3 = x_1^2 - x_1x_2 - x_2^2 + 2x_2x_3 - x_3^2 - 4x_1 + 4x_2,$$
$$g_3 = x_1^2 + x_1x_2 - x_2^2 - 2x_2x_3 - x_3^2 + 4x_1 + 4x_2.$$

Example (Computing the factors of $\det(T_4)$)

- Let $p = 101$ and choose $\boldsymbol{\alpha} = (3, 5, 4)$.
- $a(x_1, \boldsymbol{\alpha}) = x_1^4 - 93x_1^2 + 420x_1 - 416 = (x_1^2 - 7x_1 + 8)(x_1^2 + 7x_1 - 52) \in \mathbb{Z}[x_1]$.
- The first Hensel lifting step recovers x_2 and we get

$$f_2 = x_1^2 - x_1x_2 - x_2^2 - 4x_1 + 14x_2 - 25,$$
$$g_2 = x_1^2 + x_1x_2 - x_2^2 + 4x_1 - 6x_2 - 25.$$

- The second Hensel lifting step recovers x_3 and

$$f_3 = x_1^2 - x_1x_2 - x_2^2 + 2x_2x_3 - x_3^2 - 4x_1 + 4x_2,$$
$$g_3 = x_1^2 + x_1x_2 - x_2^2 - 2x_2x_3 - x_3^2 + 4x_1 + 4x_2.$$

- At the final step, we recover x_4 and obtain the true factors

$$f = x_1^2 - x_1x_2 - x_1x_4 - x_2^2 + 2x_2x_3 + x_2x_4 - x_3^2,$$
$$g = x_1^2 + x_1x_2 + x_1x_4 - x_2^2 - 2x_2x_3 + x_2x_4 - x_3^2.$$

Complexity

Theorem (Theorem 6.4.4 in [2])

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 a randomly chosen evaluation point from $[1, \tilde{N}]^{n-1}$. Suppose $\boldsymbol{\alpha}$ is Hilbertian. Then, if algorithm CMBBSHL returns an answer that is not FAIL, the total number of arithmetic operations in $\mathbb{Z}_p$ in the worst case for lifting $\hat{f}_{\rho,1}$ to $\hat{f}_{\rho,n}$ is

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

where $d_1 = \deg(a, x_1)$, $d_{\max} = \max_{j=2}^{n}(\deg(a, x_j))$, and $C(\text{probe } \boldsymbol{B})$ is the number of arithmetic operations in $\mathbb{Z}_p$ for one probe to the black box $\boldsymbol{B}$. The total number of probes to the black box is $O(nd_1 d_{\max} s_{\max})$.

The number of probes to the black box

	Approach I	Kaltofen & Trager	Rubinfeld & Zippel
Zippel's S.I.	# probes # univariate fac.	$\mathcal{O}(n\delta_{\max} d^2 \# f_{\max})$ $\mathcal{O}(n\delta_{\max} \# f_{\max})$	$\mathcal{O}(rn^2\delta_{\max}^2 d_1 T_{\max})$ $\mathcal{O}(rn^2\delta_{\max}^2 T_{\max})$
Ben-Or/ Tiwari	# probes # univariate fac.	$\mathcal{O}(d^2 \# f_{\max})$ $\mathcal{O}(\# f_{\max})$	$\mathcal{O}(rn\delta_{\max} d_1 T_{\max})$ $\mathcal{O}(rn\delta_{\max} T_{\max})$

Approach II	CMBBSHL
# probes	$\mathcal{O}(nd_1 d_{\max} s_{\max})$
# univariate fac.	1

Algorithm CMBBSHL requires the least number of probes since $T_{\max} \geq s_{\max}$ and $r\delta_{\max} \geq d_{\max}$.

Our very first benchmark [4]

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 computing $\det(T_n)$ using Zippel's quadratic Vandermonde solver. N/A: Not attempted.

Our very first benchmark [4]

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	4182.199
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	109.592
Interp2var	0.046	0.081	0.307	0.631	2.172	3.469	17.191
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	51.347
VSolve	0.058	0.101	1.937	4.219	72.887	143.183	2903.867

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

Benchmark: Computing the factors of $\det(T_n)$

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.

Fast Vandermonde Solver [2]

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	1041.959
t_n	522	814	3174	5223	19,960	34,081	127,690
BB	0.195	0.394	1.031	2.046	9.152	18.496	80.853
Interp2var	0.024	0.033	0.149	0.254	0.981	1.764	10.052
Eval $f_{i,j-1}$	0.061	0.099	0.634	1.269	14.709	32.935	508.658
BHL	0.578	0.992	3.455	6.234	24.352	45.136	240.095
VSolve	0.330	0.453	1.243	1.773	10.594	19.547	165.371

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

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.

New Benchmark: $N = 17, 18, 19$

n	15	16	17	18	19
H.L. x_n total	150.434	767.778	2332.311	22981.937	93792.430
t_n	34081	127,690	222842	821851	1457184
BB	61.493	33.768	66.888	293.162	557.319
Interp2var	0.421	1.494	6.558	13.275	25.094
Eval $f_{i,j-1}$	14.656	227.538	552.584	9396.908	23078.060
BHL	9.444	71.401	106.447	437.056	1923.675
VSolve	23.860	214.490	649.404	8156.583	50105.981

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

Thank you!!

References

- M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In Proceedings of STOC '88, pp. 301–309. ACM (1988)
- Chen, T.: Sparse Hensel lifting algorithms for multivariate polynomial factorization. PhD Thesis. Simon Fraser University (2024)
- Chen, T., Monagan, M.: 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)
- Chen, T., Monagan, M.: Factoring multivariate polynomials represented by black boxes: A Maple + C Implementation. *Math. Comput. Sci.* **16**,18 (2022)
- Chen, T., Monagan, M.: A new black box factorization algorithm - the non-monic case. In Proceedings of ISSAC 2023, pp. 173–181. ACM (2023)
- A. Diaz and E. Kaltofen. FOXBOX: a system for manipulating symbolic objects in black box representation. Proceedings of ISSAC 1998. ACM (1998)
- Von zur Gathen, J., Gerhard, J.: *Modern Computer Algebra*. Cambridge University Press (2013)
- 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. Speeding up polynomial GCD, a crucial operation in Maple. *Maple Transactions.* **2**:1 Article 14457, 2022


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


Monagan, M., Tuncer, B.: The complexity of sparse Hensel lifting and sparse polynomial factorization. *J. Symb. Cmpt.* **99**, 189–230. Elsevier (2020)


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)


Schwartz, J.: Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, **27**(4), 701–717 (1980)


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


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


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


H. Zassenhaus. On Hensel factorization I. *J. Number Theory*, **1**(3), 291–311 (1969)


Zippel, R.E.: Probabilistic algorithms for sparse polynomials. LNCS **72**, 216–226 (1979)


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


Zippel, R.E.: Interpolating polynomials from their values. *J. Symb. Cmpt.* **9**(3), 375–403 (1990)

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

[Algorithm 13, Chen (2024)]

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

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

1: Let $\hat{f}_{\rho,j-1} = \sum_{i=0}^{df_\rho} \sigma_{\rho,i}(x_2, ..., x_{j-1}) x_1^i$ $(1 \le \rho \le r)$ 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**
7: Let $Y_k = (x_2 = \beta_2^k, \cdots, x_{j-1} = \beta_{j-1}^k)$.
8: $A_k \leftarrow a_j(x_1, Y_k, x_j) \in \mathbb{Z}_p[x_1, x_j]$. ..... $\mathcal{O}(sd_1d_j\mathrm{C}(\text{probe B})) + \mathcal{O}(s(d_1^2 d_j + d_1 d_j^2))$
9: **if** $\deg(A_k, x_1) \ne d_1$ **or** $\deg(A_k, x_j) \ne d_j$ **then return** FAIL **end if**
10: $g_k \leftarrow \gcd(A_k, \frac{\partial A_k}{\partial x_1}) \bmod p \in \mathbb{Z}_p[x_1, x_j]$. ..... $\mathcal{O}(s(d_1^2 d_j + d_1 d_j^2))$
11: **if** $\deg(g_k, x_1) \ne d_1 - \sum_{\rho=1}^r df_\rho$ **then return** FAIL **end if**
12: $A_{sf} \leftarrow \mathrm{quo}\,(A_k, g_k) \bmod p$. // $A_{sf} = \mathrm{sqf}(A_k) \bmod p$, up to a constant in $\mathbb{Z}_p$.
13: $A_{sfm} \leftarrow A_{sf}/(\mathrm{LC}(\mathrm{LC}(A_{sf}, x_1), x_j)) \bmod p$. // make $\mathrm{LC}(A_{sf}, x_1)$ monic in x_j.
14: $F_{\rho,k} \leftarrow \hat{f}_{\rho,j-1}(x_1, Y_k) \in \mathbb{Z}_p[x_1]$ for $1 \le \rho \le r$. ..... $\mathcal{O}(s(\sum_{\rho=1}^r \#\hat{f}_{\rho,j-1}))$
15: **if** any $\deg(F_{\rho,k}) < df_\rho$ $(\text{for } 1 \le \rho \le r)$ **then return** FAIL **end if**

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

17: $\hat{f}_{\rho,k} \leftarrow \text{BivariateHenselLift}(A_{sfm}(x_1, x_j), F_{\rho,k}(x_1), \alpha_j, p)$. $\ldots\ldots\ldots\ldots \mathcal{O}(s(\tilde{d}_1 \tilde{d}_j^2 + \tilde{d}_1^2 \tilde{d}_j))$

18: **end for**

19: 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 $1 \leq k \leq s$, for $1 \leq \rho \leq r$ ($t_\rho = \#\hat{f}_{\rho,k}$).

20: **for** ρ from 1 to r **do**

21: **for** l from 1 to t_ρ **do**

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

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

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

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

26: **end for**

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

28: $A_\beta \leftarrow a_j(x_1, \beta) \bmod p$ via probes to **B** and Lagrange interpolation.

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

Failure Probability

Proposition (Proposition 6.4.3)

Let p be a 63-bit prime, i.e. $p \in (2^{62}, 2^{63})$. Let $\mathbb{P}_{63} = \{\text{all 63-bit primes}\}$. Let $f_\rho = \sum_{i=1}^{\#f_\rho} c_{\rho,i} \cdot x_1^{e_{i_1}} \cdots x_n^{e_{i_n}}$ for $1 \leq \rho \leq r$, where $c_{\rho,i} \neq 0$, $c_{\rho,i} \in \mathbb{Z}$, and $(e_{i_1}, \cdots, e_{i_n}) \in \mathbb{N}^n$. Let $\chi_\rho = \{i \in \mathbb{Z} \mid |c_{\rho,i}| \geq p\}$ and let $\#f_{\rho,p} = |\chi_\rho|$ for $1 \leq \rho \leq r$. Let $h_\rho = \|f_\rho\|_\infty$ for $1 \leq \rho \leq r$. Let $h_{\max} = \max_{\rho=1}^{r} h_\rho$. Then,

$$\Pr[p \mid \text{at least one } c_{\rho,i} \text{ in any } f_\rho] \leq \frac{1}{|\mathbb{P}_{63}|} \lfloor \frac{\log_2(h_{\max})}{62} \rfloor \sum_{\rho=1}^{r} \#f_{\rho,p}. \tag{2}$$

Factoring the content recursively

Let $\mathcal{C}_n = a \in \mathbb{Z}[x_1, \cdots, x_n]$. We have $\mathcal{C}_n = \mathrm{cont}(\mathcal{C}_n) \cdot \mathrm{pp}(\mathcal{C}_n)$.

$\mathrm{pp}(\mathcal{C}_n) = \prod_{\rho=1}^{r} f_\rho^{e_\rho}$. f_ρ's are irreducible over $\mathbb{Z}$ and primitive in x_1.

Let $\mathcal{C}_{n-1} = \mathrm{cont}(\mathcal{C}_n) \in \mathbb{Z}[x_2, \cdots, x_n]$.

- After factoring $\mathrm{pp}(\mathcal{C}_n)$, we create a black box $\mathsf{F}_n : \mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$ s.t. $\mathsf{F}_n(\boldsymbol{\alpha}, p) = \mathrm{pp}(\mathcal{C}_n)(\boldsymbol{\alpha}) \bmod p$ by computing

$$\mathrm{pp}(\mathcal{C}_n)(\boldsymbol{\alpha}) \bmod p = f_1(\boldsymbol{\alpha})^{e_1} \cdots f_r(\boldsymbol{\alpha})^{e_r} \bmod p.$$

- Next, we create another black box $\mathsf{C}_{n-1} : \mathbb{Z}^{n-1} \times \{p\} \to \mathbb{Z}_p$ for the content $\mathcal{C}_{n-1} = \mathrm{cont}(\mathcal{C}_n)$ s.t.

$$\mathsf{C}_{n-1}(\boldsymbol{\beta}, p) = \frac{\mathsf{C}_n([\gamma, \boldsymbol{\beta}], p)}{\mathsf{F}_n([\gamma, \boldsymbol{\beta}], p)} \bmod p$$

  for a fixed $\gamma \in \mathbb{Z}$. If $\mathsf{F}_n([\gamma, \boldsymbol{\beta}], p) = 0$ then FAIL is returned.
- Then the content is factored recursively. $\mathcal{C}_0 \in \mathbb{Z}$.

Computing the content recursively

```
MakeCont := proc( C::procedure, F::procedure, gamma::integer,
p::prime )
    proc( alpha::Array, p::prime )
        local na := numelems(alpha), alphaNew, g;
        alphaNew := Array(1..na+1);
        alphaNew[1] := gamma;
        for i to na do alphaNew[i+1] := alpha[i]; od;
        g := F( alphaNew, p );
        if g = 0 then return FAIL; fi;
        C( alphaNew, p )/g mod p;
    end;
end;
gamma0 := rand(p)();
BBC := MakeCont( Cn, Fn, gamma0, p );
```

CMBBSHL: non-monic, non-square-free

[Algorithm 12]

Input: The modular black box B : $\mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$ s.t. $\mathrm{B}(\boldsymbol{\beta}, p) = a(\boldsymbol{\beta}) \bmod p$, $(\hat{f}_{\rho,1}, 1 \le \rho \le r) \in \mathbb{Z}_p[x_1]^r$, $\boldsymbol{\alpha} \in \mathbb{Z}^{n-1}$, a prime p, $d_i = \deg(a, x_i)$ for $1 \le i \le n$ (pre-computed), $X = [x_1, \cdots, x_n]$, $n \in \mathbb{Z}$ (the recursive variable) s.t. conditions (i)-(iii) of the input are satisfied.

Output: $(\hat{f}_{\rho,n}, 1 \le \rho \le r) \in \mathbb{Z}_p[x_1, \cdots, x_n]^r$ s.t. conditions (i)-(iii) of the output are satisfied. Otherwise, **return** FAIL.

1: **if** $n = 2$ **then**
2: $A_k \leftarrow a_2(x_1, x_2) \in \mathbb{Z}_p[x_1, x_2]$. ..... $\mathcal{O}(d_1 d_2 \mathrm{C}(\text{probe B})) + \mathcal{O}(d_1^2 d_2 + d_1 d_2^2)$
3: **if** $\deg(A_k, x_1) \neq d_1$ **or** $\deg(A_k, x_2) \neq d_2$ **then return** FAIL **end if**
4: $g_k \leftarrow \gcd(A_k, \frac{\partial A_k}{\partial x_1}) \bmod p \in \mathbb{Z}_p[x_1, x_2]$. ..... $\mathcal{O}(d_1^2 d_2 + d_1 d_2^2)$
5: **if** $\deg(g_k, x_1) \neq d_1 - \sum_{\rho=1}^{r} \deg(\hat{f}_{\rho,1}, x_1)$ **then return** FAIL **end if**
6: $A_{sf} \leftarrow \mathrm{quo}(A_k, g_k) \bmod p$. // $A_{sf} = \mathrm{sqf}(A_k) \bmod p$, up to a constant in $\mathbb{Z}_p$.
7: $A_{sfm} \leftarrow A_{sf}/(\mathrm{LC}(\mathrm{LC}(A_{sf}, x_1), x_2)) \bmod p$. // make $\mathrm{LC}(A_{sf}, x_1)$ monic in x_2.
8: **return** BivariateHenselLift$(A_{sfm}, (\hat{f}_{\rho,1}, 1 \le \rho \le r), \alpha_2, p)$ ..... $\mathcal{O}(d_1^2 d_2 + d_1 d_2^2)$
9: **end if**
10: $(\hat{f}_{\rho,n-1}, 1 \le \rho \le r) \leftarrow$ CMBBSHL$(\mathrm{B}, (\hat{f}_{\rho,1}, 1 \le \rho \le r), \boldsymbol{\alpha}, p, d_i, X, n-1)$.
11: **return** CMBBSHLstepj$(\mathrm{B}, (\hat{f}_{\rho,n-1}, 1 \le \rho \le r), \boldsymbol{\alpha}, p, d_i, X, n)$

Benchmark 3: Dixon matrices

Table: Timings (in seconds) for computing the determinant of Dixon matrices.

	heron3d	heron4d	robotarms (b_1)	robotarms (t_1)	heron5d
n	7	11	8	8	16
$N \times N$	13×13	63×63	16×16	16×16	399×399
$d_j = \deg(a, x_j)$ $(1 \leq j \leq n)$	19,12,12, 8,8,8,4	89,26,26, 12,12,12 8,8,8,8,8	16,64,128,44 36,16,16,16	16,64,32,56, 32,128,16,16	2159,328,328,144, 144,144,64,64, 64,64,32,32, 32,32,32,16
r	6	4	8	7	8
$\#f_i$ $(1 \leq i \leq r)$	3,23,3, 3,1,3	22,1, 6,131	1,1,2,39, 2,1,4,7	2,1,30,6, 2,7,7	823,130,22,3 3,3,3,1
e_i $(1 \leq i \leq r)$	1,2,1, 1,7,1	2,37, 7,4	8,24,48,8 24,4,4,4	48,16,8,8, 16,4,4	8,8,20,46 46,46,1831
$\#\det(A)$	525	37666243	O/M*	O/M*	N/A
$\max \lambda_\rho$	1	1	1	2	1
CMBBSHL tot	0.685	43.809	169.851	350.809	165208.747
probes tot	5701	201183	99652	131250	36008392
$\mathrm{pp}(a)$ fac	0.683	43.804	18.972	43.178	165106.278
probes $\mathrm{pp}(a)$	5699	201181	11448	16626	-
Maple det	0.614	O/M	N/A	N/A	N/A
Maple minor	0.006	0.383	O/M	O/M	N/A
Maple fac	0.084	O/M	N/A	N/A	N/A
Maple tot	0.620	-	-	-	-

N/A: Not attempted. O/M: Out of memory.

O/M*: Out of memory when expanding the factors in Maple.

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 (H.L. x_n)	1	1	1	1	1	1
BB tot	0.024	0.070	0.156	0.353	0.650	1.224
BB eval	0.015	0.039	0.090	0.208	0.368	0.709
BB det	0.009	0.031	0.066	0.145	0.282	0.515
Interp2var	0.001	0.002	0.004	0.009	0.015	0.027
Eval $\hat{f}_{\rho_j - 1}$	0.002	0.002	0.003	0.004	0.004	0.005
BHL	0.004	0.005	0.008	0.009	0.010	0.011
VSolve	0.004	0.003	0.006	0.009	0.007	0.012

Breakdown of timings for Dixon matrices

Table: Breakdown of timings (in seconds) for computing the determinant of Dixon Matrices.

	heron3d	heron4d	robotarms (b_1)	robotarms (t_1)	heron5d
n	7	11	8	8	16
$N \times N$	13×13	63×63	16×16	16×16	399×399
H.L. x_n tot	0.146	10.431	1.451	2.958	14347.878
probes x_n	930	41112	901	1173	-
s	13	85	3	3	571
BB tot	0.039	9.303	1.398	2.910	13771.116
BB eval	0.022	4.764	1.369	2.888	7254.237
BB det	0.008	3.720	0.029	0.022	6414.483
Interp2var	0.000	0.061	0.003	0.003	17.574
Eval $\hat{f}_{\rho,j-\mathbf{1}}$	0.029	0.029	0.000	0.000	0.576
BHL	0.029	0.160	0.000	0.000	450.033
VSolve	0.001	0.002	0.000	0.002	0.031