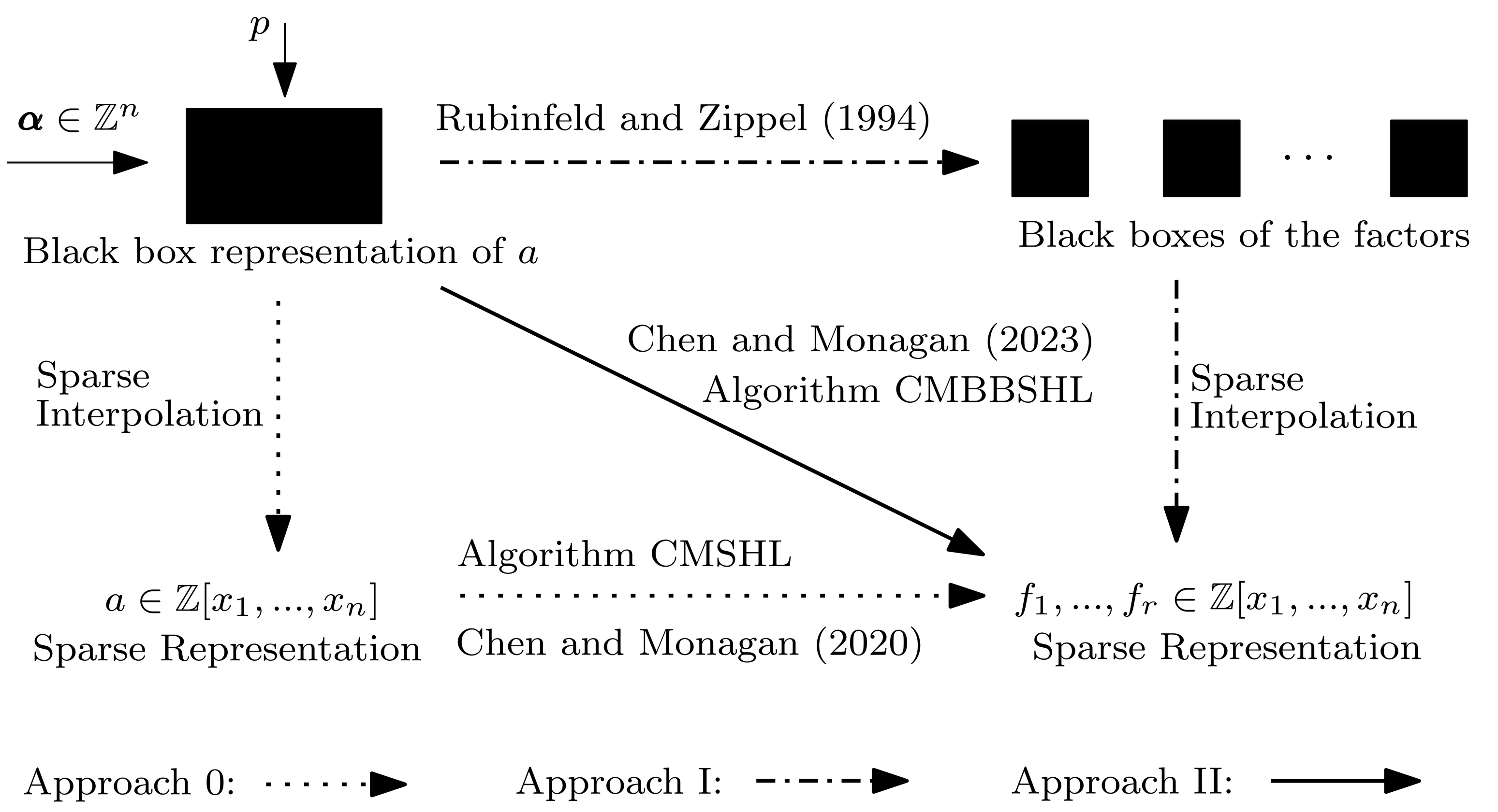
A Maple program to factor multivariate polynomials given by black boxes

Tian Chen and Michael Monagan
Department of Mathematics, Simon Fraser University, Canada

Black box factorization

Problem $\mathcal{P}$: Given a polynomial $a \in \mathbb{Z}[x_1, \cdots, x_n]$ represented by a black box **BB** : $\mathbb{Z}^n \to \mathbb{Z}$ or a **modular black box 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.



The figure above shows three approaches for solving Problem $\mathcal{P}$. Approach 0 first interpolates the sparse representation of a and then factors it using a sparse Hensel lifting algorithm, e.g. algorithm CMSHL [2]. Approach I first constructs black boxes of the factors and then applies sparse polynomial interpolation (e.g. [1, 6]) to obtain the sparse representation of the factors, e.g, Kaltofen and Trager's algorithm [4] and Rubinfeld and Zippel's algorithm [5]. Approach II (Algorithm CMBBSHL) computes the factors in the sparse representation directly [3]. **Approach II is a modular algorithm and it is the most efficient of the three.**

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 $r\delta_{\max} \geq d_{\max}$.

Algorithm CMBBSHL: Hensel lifting x_j $(j > 2)$.

Input: The modular black box $\mathbf{B} : \mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$ s.t. $\mathbf{B}(\boldsymbol{\beta}, p) = a(\boldsymbol{\beta}) \bmod p$, $(\hat{f}_{\rho,j-1}, 1 \leq \rho \leq r) \in \mathbb{Z}_p[x_1, \cdots, x_{j-1}]^r$, $\boldsymbol{\alpha} \in \mathbb{Z}^{n-1}$, a prime p, $d_i = \deg(a, x_i)$ for $1 \leq i \leq n$ (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: $(\hat{f}_{\rho,j}, 1 \leq \rho \leq r) \in \mathbb{Z}_p[x_1, \cdots, x_j]^r$ 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 $1 \leq \rho \leq r$; Otherwise, **return** FAIL.

Let $\hat{f}_{\rho,j-1} = \sum_{i=0}^{df_\rho} \sigma_{\rho,i}(x_2, ..., x_{j-1})x_1^i$ $(1 \leq \rho \leq r)$ where $\sigma_{\rho,i} = \sum_{k=1}^{s_{\rho,i}} c_{\rho,ik}M_{\rho,ik}$ with $M_{\rho,ik}$ the monomials in $\sigma_{\rho,i}$ and $df_\rho = \deg(\hat{f}_{\rho,j-1}, x_1)$.
Pick $\boldsymbol{\beta} = (\beta_2, \cdots, \beta_{j-1}) \in (\mathbb{Z}_p \backslash \{0\})^{j-2}$ at random.
Evaluate (for $1 \leq \rho \leq r$): $\mathcal{S}_\rho = \{\mathcal{S}_{\rho,i} = \{m_{\rho,ik} = M_{\rho,ik}(\boldsymbol{\beta}), 1 \leq k \leq s_{\rho,i}\}, 0 \leq i \leq df_\rho\}$.
if any $|\mathcal{S}_{\rho,i}| \neq s_{\rho,i}$ **then return** FAIL **end if** // monomial evals must be distinct
Let s be the maximum of $s_{\rho,i}$. // Compute s images of the factors in $\mathbb{Z}_p[x_1, x_j]$:
for k from 1 to s **do**
  Let $Y_k = (x_2 = \beta_2^k, \cdots, x_{j-1} = \beta_{j-1}^k)$.
  $A_k \leftarrow a_j(x_1, Y_k, x_j) \in \mathbb{Z}_p[x_1, x_j]$. // via probes to **B** and bivariate dense interpolation .... $\mathcal{O}(sd_1d_j\mathrm{C}(\text{probe }\mathbf{B})) + \mathcal{O}(s(d_1^2d_j + d_1d_j^2))$
  if $\deg(A_k, x_1) \neq d_1$ **or** $\deg(A_k, x_j) \neq d_j$ **then return** FAIL **end if**
  $g_k \leftarrow \gcd(A_k, \frac{\partial A_k}{\partial x_1}) \bmod p \in \mathbb{Z}_p[x_1, x_j]$. .... $\mathcal{O}(s(d_1^2d_j + d_1d_j^2))$
  if $\deg(g_k, x_1) \neq d_1 - \sum_{\rho=1}^r df_\rho$ **then return** FAIL **end if**
  $A_{sf} \leftarrow$ quo $(A_k, g_k) \bmod p$. // $A_{sf} = \mathrm{sqf}(A_k) \bmod p$, up to a constant in $\mathbb{Z}_p$.
  $A_{sfm} \leftarrow A_{sf}/(\mathrm{LC}(\mathrm{LC}(A_{sf}, x_1), x_j)) \bmod p$. // make $\mathrm{LC}(A_{sf}, x_1)$ monic in x_j.
  $F_{\rho,k} \leftarrow \hat{f}_{\rho,j-1}(x_1, Y_k) \in \mathbb{Z}_p[x_1]$ for $1 \leq \rho \leq r$. .... $\mathcal{O}(s(\sum_{\rho=1}^r \#\hat{f}_{\rho,j-1}))$
  if any $\deg(F_{\rho,k}) < df_\rho$ (for $1 \leq \rho \leq r$) **then return** FAIL **end if**
  if $\gcd(F_{\rho,k}, F_{\phi,k}) \neq 1$ for any $1 \leq \rho < \phi \leq r$ **then return** FAIL **end if**
  $\hat{f}_{\rho,k} \leftarrow$ BivariateHenselLift$(A_{sfm}(x_1, x_j), F_{\rho,k}(x_1), \alpha_j, p)$. .... $\mathcal{O}(s(\tilde{d}_1\tilde{d}_j^2 + \tilde{d}_1^2\tilde{d}_j))$
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 $1 \leq k \leq s$, for $1 \leq \rho \leq r$ $(t_\rho = \#\hat{f}_{\rho,k})$.
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 $\left\{\sum_{k=1}^{s_{\rho,i}} m_{\rho,ik}^t c_{\rho,lk} = \alpha_{\rho,tl} \text{ for } 1 \leq t \leq s_{\rho,i}\right\}$ 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 until $\deg(\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta})) = df_\rho$ for all $1 \leq \rho \leq r$.
$A_{\boldsymbol{\beta}} \leftarrow a_j(x_1, \boldsymbol{\beta}) \bmod p$ via probes to **B** and Lagrange interpolation.
if $\hat{f}_{\rho,j}(x_1, \boldsymbol{\beta}) \mid A_{\boldsymbol{\beta}}$ for all $1 \leq \rho \leq r$ **then return** $(\hat{f}_{\rho,j}, 1 \leq \rho \leq r)$ **else return** FAIL **end if**

Complexity

Theorem. Let $a \in \mathbb{Z}[x_1, \cdots, x_n]$. Let $(f_\rho, 1 \leq \rho \leq r)$ be the irreducible factors of a. Let $f_\rho = \sum_i \sigma_{i,\rho}(x_2, \cdots, x_n)x_1^i$. Define $s_{\max} = \max_\rho \max_i \#\sigma_{i,\rho}$. Let p be a large prime and $\tilde{N} < p$, $\tilde{N} \in \mathbb{Z}^+$. Let $\boldsymbol{\alpha} = (\alpha_2, \cdots, \alpha_n) \in \mathbb{Z}_p^{n-1}$ be randomly chosen from $[0, \tilde{N}]^{n-1}$. 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 Hensel lifting $\hat{f}_{\rho,1}$ to $\hat{f}_{\rho,n}$ using Algorithm CMBBSHL step j $n-1$ times is

$$O\left((n-2)s_{\max}d_{\max}\left(\sum_{\rho=1}^r \#\hat{f}_{\rho,j-1} + d_1^2 + d_1d_{\max} + d_1\mathrm{C}(\text{probe }\mathbf{B})\right)\right). \qquad (1)$$

where $d_1 = \deg(a, x_1)$, $d_{\max} = \max_{j=2}^n(\deg(a, x_j))$ and C(probe **B**) is the number of arithmetic operations in $\mathbb{Z}_p$ for one probe to the black box **B**. The total number of probes to the black box is $O(nd_1d_{\max}s_{\max})$.

Demonstration of the software

Let us factor $a = x_1x_2 + x_1x_3 + x_2^2 + 2x_2x_3 + x_3^2 + x_1 + 2x_2 + 2x_3 + 1$ over $\mathbb{Z}$. The factorization is $a = (x_2 + x_3 + 1)(x_1 + x_2 + x_3 + 1)$. The input modular black box $\mathbf{B} : \mathbb{Z}^n \times \{p\} \to \mathbb{Z}_p$ is coded in Maple as a Maple procedure.

The following options are specified.

```
X := [x2, x1, x3]: # Variables with a chosen ordering
alpha := Array(1..3, [2908,3830,2798]): # Random evaluation point
degA := [2, 1, 2]: # Pre-computed individual degrees
p := prevprime(2^62-1): # A chosen large prime number
Var_Perm := 1: # Yes, X is permuted
Cont_Flag := 0: # Don't compute and factor the content
MapleCode := [Maple,C,C,C]: # Maple or C code for each subroutine
LI := 0: # Not for large coefficients
```

Running `CMBBSHLcont` produces the following output:

```
> CNT := 0: # for counting the number of black box probes
> ff := CMBBSHLcont(BBInput, X, alpha, degA, p, Var_Perm,
Cont_Flag, 0, MapleCode, LI);
CMBBSHLcont: N = 3
1 prime(s) used to interpolate a(x2) = a(alpha1,x2,alpha3)
a(x2) = x2^2+9428*x2+18554571
N = 3, factors of a(x2) = [[x2+6629, 1], [x2+2799, 1]]
CMBBSHL step 3:
fN = [x1+x2+x3+1, x2+x3+1]

             ff := (x1 + x2 + x3 + 1) (x2 + x3 + 1)

> CNT; # number of black box probes
                                44
```

Our code can be downloaded from
`www.cecm.sfu.ca/~mmonagan/code/CMBBSHL`

Benchmark

We have implemented Algorithm CMBBSHL in Maple with major subroutines coded in C. The table shows the CPU timings (in seconds) for computing the factors of the determinant of T_n, the symmetric Toeplitz matrices.

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
# det(T_n)	23797	90296	350726	1338076	5165957	19732508	O/M*
# f_ρ	931, 931	1730, 849	5579, 5579	10611, 4983	34937, 34937	66684, 30458	221854, 221854
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	-	-

References

[1] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In Proceedings of STOC '88, pp. 301–309. ACM (1988)

[2] 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)

[3] Chen, T., Monagan, M.: A new black box factorization algorithm - the non-monic case. In Proceedings of ISSAC 2023. ACM (2023)

[4] 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.

[5] 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)

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