

# Towards High Performance Multivariate Factorization

Michael Monagan

Center for Experimental and Constructive Mathematics  
Simon Fraser University  
British Columbia

This is joint work with Baris Tuncer.

To factor a multivariate polynomial in  $\mathbb{Z}[x_1, \dots, x_n]$  Hensel lifting must solve many polynomial diophantine equations in  $\mathbb{Z}_p[x_2, \dots, x_j][x_1]$  for  $2 \leq j \leq n$ . These are equations of the form

$$\sigma A + \tau B = C$$

where  $A, B, C$  are given,  $\gcd(A, B) = 1$ , and we must solve for  $\sigma, \tau \in \mathbb{Z}_p[x_2, \dots, x_j][x_1]$  s.t.  $\deg_{x_1} \sigma < \deg_{x_1} B$ . Usually this is the most costly part and Wang's algorithm can be exponential in  $n$ .

## Outline

- Some factorization timing data
- Factoring  $a \in \mathbb{Z}[x_1, \dots, x_n]$  using Hensel lifting
- Wang's solution
- Kaltofen's sparse solution v. our sparse solution
- Eliminating multi-precision arithmetic
- Eliminating the error computation

# Factoring the determinants of Cyclic and Toeplitz matrices

Let  $C_n$  denote the  $n \times n$  cyclic matrix and let  $T_n$  denote the  $n \times n$  symmetric Toeplitz matrix below.

$$C_n = \begin{pmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_n \\ x_n & x_1 & \cdots & x_{n-2} & x_{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_3 & x_4 & \cdots & x_1 & x_2 \\ x_2 & x_3 & \cdots & x_n & x_1 \end{pmatrix} \quad T_n = \begin{pmatrix} x_1 & x_2 & \cdots & x_{n-1} & x_n \\ x_2 & x_1 & \cdots & x_{n-2} & x_{n-1} \\ & \ddots & \ddots & \ddots & \\ x_{n-1} & x_{n-2} & \cdots & x_1 & x_2 \\ x_n & x_{n-1} & \cdots & x_2 & x_1 \end{pmatrix}$$

The determinants of  $C_n$  and  $T_n$  are homogeneous polynomials in  $n$  variables.

## Example

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

$$\det C_4 = (x_4 + x_3 + x_1 + x_2)(x_4 - x_3 - x_1 + x_2) \\ (x_1^2 - 2x_1x_3 + x_2^2 - 2x_2x_4 + x_3^2 + x_4^2)$$

$n$	$\#d_n$	$\#factors$	Maple	MDP	MTSHL	Magma	Singular
7	427	30,56	0.035	30%	0.046	0.01	0.02
8	1628	167,167	0.065	43%	0.073	0.04	0.05
9	6090	153,294	0.166	73%	0.122	0.10	0.28
10	23797	931,931	0.610	76%	0.418	0.89	1.77
11	90296	849,1730	2.570	74%	1.138	1.96	8.01
12	350726	5579,5579	19.45	80%	13.16	72.17	84.04
13	1338076	4983,10611	84.08	84%	21.77	181.0	607.99
14	5165957	34937,34937	637.8	77%	249.9	6039.0	20333.45
15	19732508	30458,66684	4153.2	84%	1651.7	12899.2	–

**Table:** Factorization timings (seconds) for  $\det T_n$  evaluated at  $x_n = 1$

Notes: Intel Core i5-4670 CPU @ 3.40GHz, 16 gigs RAM.

Maple 2017: kernelopts(numcpus=1), Magma 2.22–5, Singular 3–1–6,

$n$	$\#d_n$	$\#f_{\max}$	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.

# Multivariate polynomial factorization

**Input**  $a = fg \in \mathbb{Z}[x_1, \dots, x_n]$  primitive and square-free.

**Output**  $f, g$ .

- 1 Pick  $x_1$  and factor  $LC(a) = h_1^{e_1} h_2^{e_2} \dots$  in  $\mathbb{Z}[x_2, \dots, x_n]$ .
- 2 Pick  $\alpha = \alpha_2, \dots, \alpha_n \in \mathbb{Z}$  such that  $LC(a)(\alpha) \neq 0$  and  $\gcd(a(\alpha), \frac{\partial a}{\partial x_1}(\alpha)) = 1$  and  $h_i(\alpha) \in \mathbb{Z}$  have distinct prime divisors.
- 3 Factor  $a(x_1, \alpha) = f(x_1, \alpha) g(x_1, \alpha)$  over  $\mathbb{Z}$ .
- 4 Distribute the factors of  $LC(a)$  on  $f(x_1, \alpha)$  and  $g(x_1, \alpha)$ .
- 5 Pick a prime  $p$  and  $L \in \mathbb{N}$  s.t.  $p^L > 2\|f\|$  where  $f|a$  and  $\gcd(f(x_1, \alpha), g(x_1, \alpha)) = 1$  in  $\mathbb{Z}_p[x_1]$ .
- 6 Hensel lift  $x_2$  then  $x_3$  then ... then  $x_n$  doing arithmetic mod  $p^L$ .

P. Wang. An improved Multivariate Polynomial Factoring Algorithm.  
*Mathematics of Computation*, **32**:1215–1231. AMS (1978)

*Algorithms for Computer Algebra*, K.O. Geddes, S.R. Czapor, G. Labahn. Kluwer, 1992.

# Wang's Multivariate Hensel Lifting (MHL)

**Input**  $a \in \mathbb{Z}[x_1, \dots, x_n]$ ,  $\alpha = (\alpha_2, \dots, \alpha_n)$ ,  $(f_0, g_0) \in \mathbb{Z}[x_1]$  s.t.  
(i)  $a(x_1, \alpha) - f_0 g_0 = 0$  and (ii)  $\gcd(f_0, g_0) = 1$  in  $\mathbb{Z}_p[x_1]$ .

**Output**  $f, g$  satisfying  $a = fg$  or FAIL.

- 1 If  $n = 1$  output  $(f_0, g_0)$ .
- 2 Lift  $x_2, \dots, x_{n-1}$ :  $(f_0, g_0) := \text{MHL}( a(x_n = \alpha_n), \alpha, (f_0, g_0) )$

Idea:  $f = \sum_{k=0}^{df} f_k(x_n - \alpha_n)^k$  and  $g = \sum_{k=0}^{dg} g_k(x_n - \alpha_n)^k$ .

# Wang's Multivariate Hensel Lifting (MHL)

**Input**  $a \in \mathbb{Z}[x_1, \dots, x_n]$ ,  $\alpha = (\alpha_2, \dots, \alpha_n)$ ,  $(f_0, g_0) \in \mathbb{Z}[x_1]$  s.t.

(i)  $a(x_1, \alpha) - f_0 g_0 = 0$  and (ii)  $\gcd(f_0, g_0) = 1$  in  $\mathbb{Z}_p[x_1]$ .

**Output**  $f, g$  satisfying  $a = fg$  or FAIL.

1 If  $n = 1$  output  $(f_0, g_0)$ .

2 Lift  $x_2, \dots, x_{n-1}$ :  $(f_0, g_0) := \text{MHL}( a(x_n = \alpha_n), \alpha, (f_0, g_0) )$

Idea:  $f = \sum_{k=0}^{df} f_k(x_n - \alpha_n)^k$  and  $g = \sum_{k=0}^{dg} g_k(x_n - \alpha_n)^k$ .

3 Initialize:  $(f, g) := (f_0, g_0)$ ;  $\text{error} := a - f_0 g_0$ .

4 For  $k = 1, \dots, \deg_{x_1} a$  while  $\text{error} \neq 0$  do

$T_k := \text{Taylor coeff}(\text{error}, (x_n - \alpha_n)^k)$

If  $T_k \neq 0$  then

Solve  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Set  $f := f + f_k(x_n - \alpha_n)^k$  and  $g := g + g_k(x_n - \alpha_n)^k$

Set  $\text{error} := a - fg$

5 If  $\text{error} = 0$  output  $(f, g)$  else output FAIL.



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

Taylor series for  $f(z)$  about  $z = \alpha_3 = 0$ :

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

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

Taylor series for  $f(z)$  about  $z = \alpha_3 = 0$ :

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

Taylor series for  $f(z)$  about  $z = \alpha_3 = 2$ :

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

Note  $f_k = f_z^{(i)}(z = 2)/k!$ .

# Wang's Multivariate Diophantine Lifting (MDP)

**Input**  $A, B, C \in \mathbb{Z}_p[x_1, \dots, x_n], \alpha = \alpha_2, \dots, \alpha_n$

**Output**  $\sigma, \tau \in \mathbb{Z}_p[x_1, \dots, x_n]$  satisfying  $\sigma A + \tau B = C$

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

Let  $\sigma = \sum_{k=0}^{df} \sigma_k (x_n - \alpha_n)^k$  and  $\tau = \sum_{k=0}^{dg} \tau_k (x_n - \alpha_n)^k$ .

2  $(\sigma_0, \tau_0) := \text{MultiDioLift}( A(x_n = \alpha_n), B(x_n - \alpha_n), C(x_n = \alpha_n), \alpha )$

# Wang's Multivariate Diophantine Lifting (MDP)

**Input**  $A, B, C \in \mathbb{Z}_p[x_1, \dots, x_n]$ ,  $\alpha = \alpha_2, \dots, \alpha_n$

**Output**  $\sigma, \tau \in \mathbb{Z}_p[x_1, \dots, x_n]$  satisfying  $\sigma A + \tau B = C$

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

Let  $\sigma = \sum_{k=0}^{df} \sigma_k (x_n - \alpha_n)^k$  and  $\tau = \sum_{k=0}^{dg} \tau_k (x_n - \alpha_n)^k$ .

2  $(\sigma_0, \tau_0) := \text{MultiDioLift}(A(x_n = \alpha_n), B(x_n - \alpha_n), C(x_n = \alpha_n), \alpha)$

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

4 For  $k = 1, 2, \dots$  while  $error \neq 0$  do

$T_k := \text{Taylor coeff}(error, (x_n - \alpha_n)^k)$

    If  $T_k \neq 0$  then

$\sigma_k, \tau_k := \text{MultiDioLift}(\sigma_0, \tau_0, T_k, \alpha)$

$\sigma, \tau := \sigma + \sigma_k (x_n - \alpha_n)^k, \tau + \tau_k (x_n - \alpha_n)^k$

$error := error - \sigma_k (x_n - \alpha_n)^k A - \tau_k (x_n - \alpha_n)^k B$

5 output  $(\sigma, \tau)$ .

# Wang's Multivariate Diophantine Lifting (MDP)

**Input**  $A, B, C \in \mathbb{Z}_p[x_1, \dots, x_n]$ ,  $\alpha = \alpha_2, \dots, \alpha_n$

**Output**  $\sigma, \tau \in \mathbb{Z}_p[x_1, \dots, x_n]$  satisfying  $\sigma A + \tau B = C$

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

Let  $\sigma = \sum_{k=0}^{df} \sigma_k (x_n - \alpha_n)^k$  and  $\tau = \sum_{k=0}^{dg} \tau_k (x_n - \alpha_n)^k$ .

2  $(\sigma_0, \tau_0) := \text{MultiDioLift}(A(x_n = \alpha_n), B(x_n - \alpha_n), C(x_n = \alpha_n), \alpha)$

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

4 For  $k = 1, 2, \dots$  while  $error \neq 0$  do

$T_k := \text{Taylor coeff}(error, (x_n - \alpha_n)^k)$

**If  $T_k \neq 0$  then**

$\sigma_k, \tau_k := \text{MultiDioLift}(\sigma_0, \tau_0, T_k, \alpha)$

$\sigma, \tau := \sigma + \sigma_k (x_n - \alpha_n)^k, \tau + \tau_k (x_n - \alpha_n)^k$

$error := error - \sigma_k (x_n - \alpha_n)^k A - \tau_k (x_n - \alpha_n)^k B$

5 output  $(\sigma, \tau)$ .

Let  $M(n)$  count calls to the Euc. alg. and  $d \geq \max(\deg_{x_i} f, \deg_{x_i} g)$ .

Then  $M(1) = 1, M(n) \leq dM(n-1) \implies M(n) \leq d^{n-1}$ .

Solve the MDP  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Interpolate  $x_2, \dots, x_{n-1}$  from images in  $\mathbb{Z}_p[x_1]$  using sparse interpolation?

Solve the MDP  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Interpolate  $x_2, \dots, x_{n-1}$  from images in  $\mathbb{Z}_p[x_1]$  using sparse interpolation?

**Theorem [MT]** The Strong Sparse Hensel Assumption

If  $\alpha_n$  is random then, with high probability

$$\begin{aligned} \text{supp}(f_0) \supseteq \text{supp}(f_1) \supseteq \dots \supseteq \text{supp}(f_{df}) \text{ and} \\ \text{supp}(g_0) \supseteq \text{supp}(g_1) \supseteq \dots \supseteq \text{supp}(g_{dg}). \end{aligned}$$

Example ( $n = 8$ ,  $\#f = 10,000$ ,  $df = 16$ ,  $\alpha = 3$ )

$|f_k| = 9877, 7043, 4932, 3374, 2310, 1545, 1001, 654, 418, 245, 141, 81, 34, 13, 5, 2, 1.$

Solve the MDP  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Interpolate  $x_2, \dots, x_{n-1}$  from images in  $\mathbb{Z}_p[x_1]$  using sparse interpolation?

**Theorem [MT]** The Strong Sparse Hensel Assumption

If  $\alpha_n$  is random then, with high probability

$$\begin{aligned} \text{supp}(f_0) \supseteq \text{supp}(f_1) \supseteq \dots \supseteq \text{supp}(f_{df}) \text{ and} \\ \text{supp}(g_0) \supseteq \text{supp}(g_1) \supseteq \dots \supseteq \text{supp}(g_{dg}). \end{aligned}$$

Example ( $n = 8$ ,  $\#f = 10,000$ ,  $df = 16$ ,  $\alpha = 3$ )

$|f_k| = 9877, 7043, 4932, 3374, 2310, 1545, 1001, 654, 418, 245, 141, 81, 34, 13, 5, 2, 1$ .

**Kaltofen's Sparse Hensel Lifting.**

Let  $\text{supp}(f_0) = \{M_1, \dots, M_r\}$  and  $\text{supp}(g_0) = \{N_1, \dots, N_s\}$

Equate the coefficients  $c_i$  of

$$\left( \sum_{i=1}^r c_i M_i \right) g_0 + \left( \sum_{i=1}^s c_{r+i} N_i \right) f_0 = T_k \quad \text{in } \mathbb{Z}_p[x_1, \dots, x_{n-1}]$$

in monomials in  $(x_1, \dots, x_{n-1})$  and solve the  $(r+s) \times (r+s)$  linear system.



# MTSHL : Monagan and Tuncer's Sparse Hensel Lifting

Solve  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Let  $f_{k-1} = \sum_{i=1}^{df} a_i(x_2, \dots, x_{n-1})x_1^i$  and  $\text{supp}(a_i) = \{M_{i1}, \dots, M_{ir_i}\}$ ,  
and  $g_{k-1} = \sum_{i=1}^{dg} b_i(x_2, \dots, x_{n-1})x_1^i$  and  $\text{supp}(b_i) = \{N_{i1}, \dots, N_{is_i}\}$ .

**Assume**  $f_k = \sum_{i=0}^{df} \left( \sum_{j=1}^{r_i} a_{ij} M_{ij} \right) x_1^i$  and  $g_k = \sum_{i=0}^{dg} \left( \sum_{j=1}^{s_i} b_{ij} N_{ij} \right) x_1^i$ .

# MTSHL : Monagan and Tuncer's Sparse Hensel Lifting

Solve  $f_k g_0 + g_k f_0 = T_k$  for  $f_k, g_k \in \mathbb{Z}_p[x_1, \dots, x_{n-1}]$ .

Let  $f_{k-1} = \sum_{i=1}^{df} a_i(x_2, \dots, x_{n-1})x_1^i$  and  $\text{supp}(a_i) = \{M_{i1}, \dots, M_{i r_i}\}$ ,  
and  $g_{k-1} = \sum_{i=1}^{dg} b_i(x_2, \dots, x_{n-1})x_1^i$  and  $\text{supp}(b_i) = \{N_{i1}, \dots, N_{i s_i}\}$ .

**Assume**  $f_k = \sum_{i=0}^{df} \left( \sum_{j=1}^{r_i} a_{ij} M_{ij} \right) x_1^i$  and  $g_k = \sum_{i=0}^{dg} \left( \sum_{j=1}^{s_i} b_{ij} N_{ij} \right) x_1^i$ .

Let  $t = \max(\#a_i, \#b_i) \ll \#a + \#b$ . Let  $\beta = (\beta_2, \dots, \beta_{n-1}) \in \mathbb{Z}_p^{n-2}$ .

Solve  $\sigma_j g_0(\beta^j) + \tau_j f_0(\beta^j) = T_k(\beta^j)$  for  $\sigma_j, \tau_j \in \mathbb{Z}_p[x_1]$ . **for**  $1 \leq j \leq t$ .

Solve the Vandermonde systems

$$\begin{cases} \text{coeff}(f_k(\beta^j), x_n^i) = \text{coeff}(\sigma_j, x_n^i) \text{ for } 1 \leq j \leq \#a_i \} & \text{for } 0 \leq i < df, \\ \text{coeff}(g_k(\beta^j), x_n^i) = \text{coeff}(\tau_j, x_n^i) \text{ for } 1 \leq j \leq \#b_i \} & \text{for } 0 \leq i < dg. \end{cases}$$

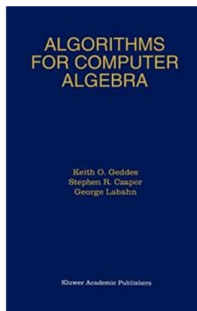
Cost of MTSHL: number of arith. ops in  $\mathbb{Z}_p$  where  $d = \max \deg_{x_i} a$

$$(n-1)d \ O\left( \underbrace{\#f \#g}_{\text{error}} + \underbrace{\#a}_{\text{Taylor}} + \underbrace{(\#a + \#f + \#g)t}_{\text{eval}} + \underbrace{d^2 t}_{\text{UniDio}} + \underbrace{dt^2}_{\text{Vandermonde}} \right).$$

$n/d/T$	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)

>  $M := \text{product}(x_i, i = 1..n)$ ;

>  $f := x_1^d + M \text{randpoly}([x_1, \dots, x_n], \text{terms} = T, \text{degree} = d)$ ;



## Chapter 6

### Newton's Iteration and the Hensel Construction

Run the entire Hensel lifting mod  $p^L$   
where  $p^L > 2 \max(\|a\|, \|f\|, \|g\|)$ .

Why?

To avoid the expression swell when solving  
 $\sigma g_0(\alpha) + \tau f_0(\alpha) = T_k(\alpha)$  in  $\mathbb{Z}[x_1]$  in MDP.

**Lemma [Gelfond]** Let  $a \in \mathbb{Z}[x_1, \dots, x_n]$  with  $d_i = \deg_{x_i} a$ .  
If  $f|a$  then

$$\|f\| \leq e^{d_1+d_2+\dots+d_n} \|a\| \quad \text{where } e = 2.718281828$$

Let  $p$  be a prime and let  $f = \sum_{k=0}^{hf} f_k p^k$  and  $g = \sum_{k=0}^{hg} g_k p^k$ .

**Example**

$$\begin{aligned} f &= 2x_1 + (5 + 0 \cdot p + 2p^2)x_2 + (7 + 3p)x_3 \\ &= \underbrace{(2x_1 + 5x_2 + 7x_3)}_{f_0} + \underbrace{3x_3}_{f_1} p + \underbrace{2x_2}_{f_2} p^2. \end{aligned}$$

$$\text{supp}(f_0) = \{x_1, x_2, x_3\} \supseteq \text{supp}(f_1) = \{x_3\} \not\supseteq \text{supp}(f_2) = \{x_2\}.$$

Let  $p$  be a prime and let  $f = \sum_{k=0}^{hf} f_k p^k$  and  $g = \sum_{k=0}^{hg} g_k p^k$ .

**Example**

$$\begin{aligned} f &= 2x_1 + (5 + 0 \cdot p + 2p^2)x_2 + (7 + 3p)x_3 \\ &= \underbrace{(2x_1 + 5x_2 + 7x_3)}_{f_0} + \underbrace{3x_3}_{f_1} p + \underbrace{2x_2}_{f_2} p^2. \end{aligned}$$

$$\text{supp}(f_0) = \{x_1, x_2, x_3\} \supseteq \text{supp}(f_1) = \{x_3\} \not\supseteq \text{supp}(f_2) = \{x_2\}.$$

**Theorem** If  $p$  is chosen at random from  $[2^{b-1}, 2^b]$  for  $b$  sufficiently large then

$$\begin{aligned} \text{supp}(f_0) \supseteq \text{supp}(f_1) \supseteq \text{supp}(f_2) \supseteq \dots \text{supp}(f_{hf}) \text{ and} \\ \text{supp}(g_0) \supseteq \text{supp}(g_1) \supseteq \text{supp}(g_2) \supseteq \dots \text{supp}(g_{hg}) \text{ whp.} \end{aligned}$$

**Idea:** Run the entire Hensel lifting modulo a **machine** prime  $p$  to obtain  $f_0, g_0 \in \mathbb{Z}_p[x_1, \dots, x_n]$  satisfying  $a - f_0 g_0 \pmod p = 0$ . Now do a  $p$ -adic lift on  $f_0, g_0$  to get  $f, g$ .

# Multivariate $p$ -adic Lifting

**Input**  $a \in \mathbb{Z}[x_1, \dots, x_n]$ ,  $(f_0, g_0) \in \mathbb{Z}_p[x_1, \dots, x_n]$  satisfying  $a - f_0 g_0 \pmod{p} = 0$   
prime  $p$  and a lifting bound  $L$  satisfying  $p^L > \|f\|$  where  $f|a$ .

**Output**  $f, g \in \mathbb{Z}[x_1, \dots, x_n]$  satisfying  $a = fg$  or FAIL.

1 Initialize:  $(f, g) := (f_0, g_0)$ ;  $error := a - f_0 g_0$ ;

2 For  $k = 1, \dots, L$  while  $error \neq 0$  do

$$c_k := \left(\frac{error}{p^k}\right) \pmod{p}.$$

If  $c_k \neq 0$  then

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

Set  $f_k := \text{mods}(\sigma, p)$  and  $g_k := \text{mods}(\tau, p)$ .

Set  $f := f + f_k p^k$ ;  $g := g + g_k p^k$ ;  $error := a - fg$ .

3 If  $error = 0$  output  $(f, g)$  else output FAIL.

The MDP is solved in  $\mathbb{Z}_p[x_1, \dots, x_n]$  where  $p$  is a machine prime!

$n/d/T$	$m$	$L$	MTSHL mod $p^L$	MTSHL mod $p$ + lift
5/10/300	2	5	5.86 (5.10)	0.197 + 0.241 = 0.438
5/10/300	4	9	6.92 (6.10)	0.514 + 0.553 = 1.067
5/10/300	8	17	8.63 (7.596)	0.477 + 2.076 = 2.553
5/10/1000	2	5	14.44 (12.82)	0.870 + 1.332 = 2.202
5/10/1000	4	9	16.94 (15.18)	0.921 + 2.632 = 3.553
5/10/1000	8	17	19.03 (17.22)	0.873 + 4.032 = 4.905

**Table:** CPU timings (seconds) for  $p$ -adic lifting for  $p = 2^{31} - 1$ .

$> f := x_1^d + \text{randpoly}([x_1, \dots, x_n], \text{terms} = T, \text{degree} = d, \text{coeffs} = \text{rand}(p^m));$



# Eliminating the error multiplication $a - fg$

Let  $f^{(k)} = \sum_{i=0}^{k-1} f_i(x_n - \alpha_n)^i$  and  $g^{(k)} = \sum_{i=0}^{k-1} g_i(x_n - \alpha_n)^i$ .

**Main Idea:**  $error(\beta^j) = a(\beta^j) - f^{(k)}(\beta^j) g^{(k)}(\beta^j)$  in  $\mathbb{Z}_p[x_1, x_n]$ .

for  $j = 1..t$  do evaluate  $a(\beta^j), f_0(\beta^j), g_0(\beta^j)$ .

for  $j = 1..t$  **in parallel** do

Hensel lift  $f_0(\beta^j)$  to  $f(\beta^j) = \sum_{k=0}^d f_k(\beta^j)(x_n - \alpha_n)^k$ .

And also  $g_0(\beta^j)$  to  $g(\beta^j)$ .

for  $k = 1..d$  in do solve for coefficients of  $f_k, g_k$ .

Set  $f = \sum_{k=0}^d f_k(x_n - \alpha_n)^k$  and  $g = \sum_{k=0}^d g_k(x_n - \alpha_n)^k$ .

$$(n-1)d O\left( \underbrace{\#f \#g}_{\text{error}} + \underbrace{\#a}_{\text{Taylor}} + \underbrace{d^2 t}_{\text{Taylor}} + \underbrace{(\#a + \#f + \#g) t/d}_{\text{eval}} + \underbrace{d^2 t}_{\text{UniDio}} + \underbrace{dt^2}_{\text{Vandermonde}} \right).$$

We eliminated the the error computation in  $\mathbb{Z}_p[x_1, \dots, x_{n-1}]$  !