Polyomial factorization in Maple 2019.

Michael Monagan

Centre for Experimental and Constructive Mathematics
Simon Fraser University
British Columbia

This is joint work with Baris Tuncer.

Supported by NSERC of Canada and Maplesoft

Problem: Factor a in $\mathbb{Z}[x_1, x_2, \ldots, x_n]$ over $\mathbb{Q}$.

```
> a := 20*x^2*y^6*z+5*x^4*y^3*z+30*x*y^4*z^3+12*x*y^4*z^2
>     +4*x^3*y^3+3*x^3*y*z^2 +18*y^2*z^4+x^5+6*x^2*y*z^2;
```

$$20\,x^2y^6z+5\,x^4y^3z+30\,xy^4z^3+12\,xy^4z^2+4\,x^3y^3+3\,x^3yz^2+18\,y^2z^4+x^5+6\,x^2yz^2$$

```
> factor(a);
```

$$\left(5\,zxy^3+3\,yz^2+x^2\right)\left(4\,xy^3+x^3+6\,yz^2\right)$$

Problem: Factor a in $\mathbb{Z}[x_1, x_2, \ldots, x_n]$ over $\mathbb{Q}$.

```
> a := 20*x^2*y^6*z+5*x^4*y^3*z+30*x*y^4*z^3+12*x*y^4*z^2
>    +4*x^3*y^3+3*x^3*y*z^2 +18*y^2*z^4+x^5+6*x^2*y*z^2;
```

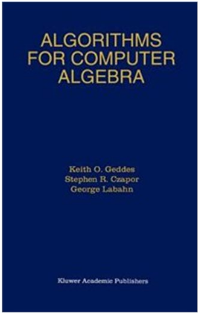
$$20\,x^2y^6z+5\,x^4y^3z+30\,xy^4z^3+12\,xy^4z^2+4\,x^3y^3+3\,x^3yz^2+18\,y^2z^4+x^5+6\,x^2yz^2$$

```
> factor(a);
```

$$\left(5\,zxy^3+3\,yz^2+x^2\right)\left(4\,xy^3+x^3+6\,yz^2\right)$$

Main Tool: Multivariate Hensel Lifting (MHL)

Developed in the 1970's.
Coded in Maple by Keith Geddes in the 1980's.
Keith implemented Paul Wang's MHL which recovers the variables one at a time.
See Ch 6 of *Algorithms for Computer Algebra*, 1992.
Wang's MHL is also in Magma, Reduce, Mathematica and Singular $\Longrightarrow$ Sage.



Talk Outline

1. Wang's MHL
2. 1: Solving **multi-term** multivariate diophantine equations.
3. 2: Idea for a random polynomial time algorithm.
4. Benchmark 1
5. 3: Factors with large integer coefficients.
6. Benchmarks 2 & 3
7. Concluding Remarks

Factoring polynomials using Wang's Hensel lifting

```
> e1 := (a = f*g);
```

$$e1 := x^5 + 20\,x^2y^6z + 5\,x^4y^3z + 30\,xy^4z^3 + 12\,xy^4z^2 + 4\,x^3y^3 + 3\,x^3yz^2 + 18\,y^2z^4 + 6\,x^2yz^2 = \left(x^2 + 5\,xy^3z + 3\,yz^2\right)\left(x^3 + 4\,xy^3 + 6\,yz^2\right)$$

```
> e2 := eval(e1,z=3);
```

$$e2 := x^5 + 60\,x^2y^6 + 15\,x^4y^3 + 4\,x^3y^3 + 918\,xy^4 + 27\,x^3y + 54\,x^2y + 1458\,y^2 = \left(x^2 + 15\,xy^3 + 27\,y\right)\left(x^3 + 4\,xy^3 + 54\,y\right)$$

```
> e3 := eval(e2,y=-5);
```

$$e3 := x^5 - 1875\,x^4 - 635\,x^3 + 937230\,x^2 + 573750\,x + 36450 = \left(x^2 - 1875\,x - 135\right)\left(x^3 - 500\,x - 270\right)$$

Notes: Let h be any factor of a and let $B > \max(||h||_\infty, ||a||_\infty)$.
Multivariate Hensel Lifting (MHL) is done modulo $m = p^L > 2B$.

Wang's Multivariate Hensel Lifting (MHL) : j'th step

Input $a \in \mathbb{Z}_p[x_1, ..., x_j]$, $\alpha = (\alpha_2, \ldots \alpha_j)$, $f_{10}, \ldots, f_{r0} \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$ s.t.
(i) $a(x_1, ..., x_{j-1}, \alpha_j) = \prod_{i=1}^{r} f_{i0}$ and
(ii) $\forall i \neq \gcd(f_{i0}(x_1, \alpha), f_{j0}(x_1, \alpha)) = 1$ in $\mathbb{Z}_p[x_1]$.

Idea: $f_i = f_{i0} + \sigma_{i1}(x_j - \alpha_j) + \sigma_{i2}(x_j - \alpha_j)^2 + \ldots$

Wang's Multivariate Hensel Lifting (MHL) : j'th step

Input $a \in \mathbb{Z}_p[x_1, ..., x_j]$, $\alpha = (\alpha_2, \ldots \alpha_j)$, $f_{10}, \ldots, f_{r0} \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$ s.t.
(i) $a(x_1, ..., x_{j-1}, \alpha_j) = \prod_{i=1}^{r} f_{i0}$ and
(ii) $\forall i \neq \gcd(f_{i0}(x_1, \alpha), f_{j0}(x_1, \alpha)) = 1$ in $\mathbb{Z}_p[x_1]$.

Idea: $f_i = f_{i0} + \sigma_{i1}(x_j - \alpha_j) + \sigma_{i2}(x_j - \alpha_j)^2 + \ldots$

Initialize: $f_i \leftarrow f_{i0}$ for $1 \leq i \leq r$ and $error := a - f_1 f_2 \cdots f_r$

For $k = 1, 2, \ldots,$ while $\deg(f_1 f_2 \cdots f_r, x_j) < \deg(a, x_j)$ do

$c_k :=$ coeff($error$, $(x_j - \alpha_j)^k$)
If $c_k \neq 0$ then

Solve the MDP $\sum_{i=1}^{r} \sigma_{ik} \prod_{j \neq i} f_{j0} = c_k$ for $\sigma_{ik} \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$.

Set $f_i \leftarrow f_i + \sigma_{ik}(x_j - \alpha_j)^k$ for $1 \leq i \leq r$.
Set $error := a - f_1 f_2 \cdots f_r$

If $error = 0$ output $(f_1, f_2, \ldots, f_r)$ else output FAIL.

Wang's MDP algorithm is $O(d^n)$ if the α_i's are non-zero.

1: Multi-term multivariate diophantine equations.

At the k'th iteration of the j'th step we must solve the multi-term MDP

$$\sum_{i=1}^{r} \sigma_{ik} \prod_{j \neq i} f_{j0} = c_k \text{ in } \mathbb{Z}_p[x_1, \ldots, x_{j-1}] \text{ s.t. } \deg(\sigma_{ik}, x_1) < \deg(\prod_{j \neq i} f_{j0}, x_1).$$

The iterative method (see [**GCL Ch. 6**]) for $r = 4$ factors

$$\begin{aligned} \sigma_1 f_2 f_3 f_4 + \sigma_2 f_1 f_3 f_4 + \sigma_3 f_1 f_2 f_4 + \sigma_4 f_1 f_2 f_3 &= c_k \\ \sigma_1 \underbrace{f_2 f_3 f_4}_{b_1} + f_1 \underbrace{\left(\sigma_2 f_3 f_4 + f_2 \underbrace{\left(\sigma_3 f_4 + \sigma_4 f_3\right)}_{\tau_2}\right)}_{\tau_1} &= c_k \end{aligned}$$

This iterative method reduces to solving $r - 1$ two term MDPs.

Idea: interpolate $\sigma_1, \sigma_2, ..., \sigma_r$ simultaneously from values. For sparse interpolation we first need to find $\mathrm{supp}(\sigma_i)$, the set of monomials $x_1^{\blacksquare} x_2^{\blacksquare} \cdots x_{j-1}^{\blacksquare}$ in σ_i .

The Taylor Coefficients : MT CASC 2016

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

Idea: $f = f_0 + \sigma_1(z-\alpha_3) + \sigma_2(z-\alpha_3)^2 + \sigma_3(z-\alpha_3)^3 + \sigma_4(z-\alpha_3)^4.$

If $\alpha_3 = 0$ then $f(z) = \underbrace{(x^3-2)}_{f_0} + \underbrace{(y^3-xy)}_{\sigma_2} z^2 + \underbrace{1}_{\sigma_4} z^4.$

If $\alpha_3 = 2$ then

$$\begin{aligned} f(z) &= \underbrace{(x^3+4\,y^3-4\,xy+14)}_{f_0} + \underbrace{(4\,y^3-4\,xy+32)}_{\sigma_1}(z-2) + \\ &\quad \underbrace{(y^3-xy+24)}_{\sigma_2}(z-2)^2 + \underbrace{8}_{\sigma_3}(z-2)^3 + \underbrace{1}_{\sigma_4}(z-2)^4 \end{aligned}$$

The Taylor Coefficients : MT CASC 2016

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

Idea: $f = f_0 + \sigma_1(z-\alpha_3) + \sigma_2(z-\alpha_3)^2 + \sigma_3(z-\alpha_3)^3 + \sigma_4(z-\alpha_3)^4.$

If $\alpha_3 = 0$ then $f(z) = \underbrace{(x^3 - 2)}_{f_0} + \underbrace{(y^3 - xy)}_{\sigma_2} z^2 + \underbrace{1}_{\sigma_4} z^4.$

If $\alpha_3 = 2$ then

$$\begin{aligned} f(z) &= \underbrace{(x^3 + 4y^3 - 4xy + 14)}_{f_0} + \underbrace{(4y^3 - 4xy + 32)}_{\sigma_1}(z-2) + \\ &\quad \underbrace{(y^3 - xy + 24)}_{\sigma_2}(z-2)^2 + \underbrace{8}_{\sigma_3}(z-2)^3 + \underbrace{1}_{\sigma_4}(z-2)^4 \end{aligned}$$

Lemma 1 [MT 2016] If α_j is chosen at random from a sufficiently large set then

$$\mathrm{Prob}[\,\mathrm{supp}(f_0) \supseteq \mathrm{supp}(f_1) \supseteq \cdots \supseteq \mathrm{supp}(f_k)\,] \text{ is high}$$

Solving $\sum_{i=1}^{r} \sigma_{ik} \prod_{j \neq i} f_{j0} = c_k$ for $\sigma_{ik} \in \mathbb{Z}_p[x_1, \ldots, x_{j-1}]$.

Let $B_i = \prod_{j \neq i} f_{j0}$ and $\sigma_{i0} = f_{i0}$.
At step k

1. Let $\sigma_{ik-1} = \sum_{j=0} a_{ij} x_1^j$ and $X_{ij} = \mathrm{supp}(a_{ij})$.

   Assume $\sigma_{ik} = \sum_{j=0} b_{ij} x_1^j$ where $b_{ij} = \sum_{k=1}^{|X_{ij}|} a_{ijk} X_{ijk}$ for unknowns a_{ijk}.

2. Pick $\beta = (\beta_2, \ldots, \beta_{j-1})$ at random from $\mathbb{Z}_p$ Needs $|X_{ij}(\beta)| = |X_{ij}|$.
3. Solve the univariate multi-term diophantine equations

$$\sum_{i=1}^{r} \sigma_{is} B_i(x_1, \beta^s) = c_k(x_1, \beta^s) \;\text{ for } 1 \leq s \leq \max |X_{ij}| \text{ for } \sigma_{is} \in \mathbb{Z}_p[x_1].$$

   Needs $\gcd(f_{i0}(x_1, \beta^s), f_{j0}(x_1, \beta^s)) = 1$ in $\mathbb{Z}_p[x_1]$ for $i \neq j$.

4. Equate coefficients and solve the shifted Vandermonde systems for a_{ijk}
   for $1 \leq i \leq r$ **for** $1 \leq j < \deg(f_i, x_1)$ solve

$$\{ b_{ij}(\beta^s) = \mathrm{coeff}(\sigma_{is}, x_1^j) \;\text{ for } 1 \leq s \leq |X_{ij}| \}.$$

Benchmark for r factors

$r/n/d/\#f_i$	Wang(MDP)	old MTSHL(MDP)	new MTSHL(MDP)
3/9/10/30	18.94(16.00)	2.26(0.60)	1.36(0.30)
4/9/15/30	OOM	104.72(23.23)	90.04(6.55)
3/9/10/50	251.20(240.77)	8.87(2.28)	4.99(0.71)
3/9/15/100	2302.7(2235.2)	122.36(**28.58**)	99.28(**8.17**)
3/11/15/100	OOM	272.78(42.74)	208.35(11.51)
3/11/10/100	515.98(424.76)	189.07(23.90)	146.80(6.25)
3/11/20/100	OOM	316.12(66.7)	256.79(19.22)

Timings in seconds for Wang's method, MTSHL from 2016 and new MTSHL.
Legend: $r = \#$factors, $n =$ number of variables, $d = \deg(f_i)$.

2: Factors with large integer coefficients

Wang's method [see **GCL** Ch. 6]

1 Factor $a(x_1, \alpha_2, \ldots, \alpha_j) = \prod_{i=1}^{r} f_i(x_1)$.
2 Pick a prime p and $L \in \mathbb{N}$ such that $p^L > 2\max(||f||_\infty, ||a||_\infty)$ where $f|a$.

Gelfond: $||f||_\infty \leq e^{d_1+d_2+\cdots+d_n}||a||_\infty$ where $d_i = \deg(a, x_i)$

3 **MHL**: Lift x_2 then x_3 etc. doing coefficient arithmetic mod p^L.
This usually means lots of multi-precision integer arithmetic.

2: Factors with large integer coefficients

Wang's method [see **GCL** Ch. 6]

1 Factor $a(x_1, \alpha_2, \ldots, \alpha_j) = \prod_{i=1}^{r} f_i(x_1)$.
2 Pick a prime p and $L \in \mathbb{N}$ such that $p^L > 2\max(||f||_\infty, ||a||_\infty)$ where $f|a$.

Gelfond: $||f||_\infty \leq e^{d_1+d_2+\cdots+d_n}||a||_\infty$ where $d_i = \deg(a, x_i)$

3 **MHL**: Lift x_2 then x_3 etc. doing coefficient arithmetic mod p^L.
This usually means lots of multi-precision integer arithmetic.

We propose instead

1 same
2 same except use a machine prime p
3 **MHL**: Lift x_2 then x_3 etc. doing coefficient arithmetic mod p.

Then lift the factors mod p^2, p^3, ... to p^L stopping if error is 0.

Must solve MDPs in $\mathbb{Z}_p[x_1, \ldots, x_n]$ – in all n variables!

Consider a factor f of a with large integer coefficients.

Let p be a prime and let $f = \sum_{k=0}^{df} f_k p^k$ be a factor of $a(x_1, \ldots, x_n)$.

After MHL mod p we have computed f_0.

Example

$$\begin{aligned} f &= 2x_1 + (5 + 0 \cdot p + 2p^2)x_2 + (7 + 3p)x_3 \\ &= \underbrace{(2x_1 + 5x_2 + 7x_3)1}_{f_0} + \underbrace{3x_3}_{f_1} p + \underbrace{2x_2}_{f_2} p^2. \end{aligned}$$

$$\mathrm{supp}(f_0) \supseteq \mathrm{supp}(f_1) \not\supseteq \mathrm{supp}(f_2).$$

Consider a factor f of a with large integer coefficients.

Let p be a prime and let $f = \sum_{k=0}^{df} f_k p^k$ be a factor of $a(x_1, \ldots, x_n)$.

After MHL mod p we have computed f_0.

Example

$$\begin{aligned} f &= 2x_1 + (5 + 0 \cdot p + 2p^2)x_2 + (7 + 3p)x_3 \\ &= \underbrace{(2x_1 + 5x_2 + 7x_3)1}_{f_0} + \underbrace{3x_3}_{f_1}\, p + \underbrace{2x_2}_{f_2}\, p^2. \end{aligned}$$

$$\mathrm{supp}(f_0) \supseteq \mathrm{supp}(f_1) \not\supseteq \mathrm{supp}(f_2).$$

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

$$\text{Prob}[\ \mathrm{supp}(f_0) \supseteq \mathrm{supp}(f_1) \supseteq \mathrm{supp}(f_2) \supseteq \ldots \mathrm{supp}(f_{df})\ ] \text{ is high.}$$

Idea: Pick a **63 bit** prime p and assume it's true!

Algorithm p-adic lift for $r = 2$: monic case

Input : $a \in \mathbb{Z}[x_1, \ldots, x_n]$, $f_0, g_0 \in \mathbb{Z}_p[x_1, \ldots, x_n]$ such that $a = f_0 g_0$ in $\mathbb{Z}_p[x_1, \ldots, x_n]$. Also an integer lifting bound $L > 0$.

Output : $f, g \in \mathbb{Z}[x_1, \ldots, x_n]$ such that $a = fg \in \mathbb{Z}[x_1, \ldots, x_n]$ or FAIL

$(f, g) \leftarrow (f_0, g_0)$.
$error \leftarrow a - fg$.
for k from 1 to L **while** $error \neq 0$ **do**
    $c_k \leftarrow \frac{error}{p^k} \mod p$
    Solve the MDP $f_k\, g_0 + g_k\, f_0 = c_k$ for f_k and g_k in $\mathbb{Z}_p[x_1, \ldots, x_n]$
    assuming $\mathrm{supp}(f_k) \subseteq \mathrm{supp}(f_{k-1})$ and $\mathrm{supp}(g_k) \subseteq \mathrm{supp}(g_{k-1})$
    if (σ, τ)=FAIL **then return** FAIL **end if**
    $(f, g) \leftarrow (f + f_k p^k,\ g + g_k p^k)$.
    $error \leftarrow a - fg$
end for
if $error \neq 0$ **then return** FAIL **else return** (f, g) **end if**

Benchmark for p-adic lift

For $p = 2^{31} - 1$, coefficients from $(-p^l, p^l)$, L is the lifting bound.

$n/d/\#f_i$	l	L	MTSHL (MDP)	MTSHL-d (MDP) (Lift)		
5/10/300	2	5	5.866 (5.101)	0.438	(0.132)	(0.241)
5/10/500	2	5	9.265 (7.937)	1.194	(0.186)	(0.480)
5/10/1000	2	5	14.448 (12.826)	2.202	(0.264)	(1.332)
5/10/300	4	9	6.923 (6.104)	1.067	(0.156)	(0.553)
5/10/500	4	9	10.971 (9.737)	1.854	(0.219)	(1.231)
5/10/1000	4	9	16.943 (15.183)	3.552	(0.350)	(2.632)
5/10/300	4	17	8.638 (7.596)	2.553	(0.201)	(2.076)
5/10/500	4	17	13.118 (11.686)	3.101	(0.280)	(2.396)
5/10/1000	4	17	19.031 (17.225)	4.905	(0.459)	(4.032)

Timings in CPU seconds for two factors with large integer coefficients.

Compute and factor det C_n where C_n is the $n \times n$ cylic matrix. For example

$$C_4 = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_4 & x_1 & x_2 & x_3 \\ x_3 & x_4 & x_1 & x_2 \\ x_2 & x_3 & x_4 & x_1 \end{bmatrix}$$

$\det(C_4) =$
$x_1^4 - 4\,x_1^2 x_2 x_4 - 2\,x_1^2 x_3^2 + 4\,x_1 x_2^2 x_3 + 4\,x_1 x_3 x_4^2 - x_2^4 + 2\,x_2^2 x_4^2 - 4\,x_2 x_3^2 x_4 + x_3^4 - x_4^4$
$= (x_1 + x_2 + x_3 + x_4)(x_1 - x_2 + x_3 - x_4)(x_1^2 - 2\,x_1\,x_3 + x_2^2 - 2\,x_2\,x_4 + x_3^2 + x_4^2)$

n	#det	deg(f_i)	max #f_i	MTSHL	Wang	(MDP)	Magma
8	810	1,1,2,4	86	0.140s	0.096s	(52%)	0.12s
9	2704	1,2,6	1005	0.465s	0.253s	(76%)	1.02s
10	7492	1,1,4,4	715	3.03s	1.020s	(49%)	10.97s
11	32066	1,10	184756	1.33s	12.43s	(88%)	142.85s
12	86500	1,1,2,2,2,4	621	4.97s	20.51s	(65%)	7575.14s
13	400024	1,12	2704156	10.24s	212.40s	(88%)	30,871.9s
14	1366500	1,1,6,6	27132	666.0s	1364.4s	(68%)	$> 10^6$s

Table: Timings (CPU time seconds) for factoring $\det(C_n(x_n = 1))$

Concluding Remarks

Baris installed the new MTSHL code into Maple for Maple 2019.
This was done under a MITACS internship with Maplesoft.

- Michael Monagan and Baris Tuncer:
  Using Sparse Interpolation in Hensel Lifting.
  Proceedings of CASC 2016, LNCS **9890**, pp. 381–400, 2016.
- Michael Monagan and Baris Tuncer:
  Factoring multivariate polynomials with many factors and huge coefficients.
  Proceedings of CASC 2018, LNCS **11077**, pp. 319–400, 2018.
- Michael Monagan and Baris Tuncer:
  The complexity of sparse Hensel lifting and sparse polynomial factorization To appear in *J. Symbolic Computation*, 2019.

Thank you for attending!