Linear Hensel Lifting for $\mathbb{F}_p[x, y]$ and $\mathbb{Z}[x]$ with Cubic Cost

Michael Monagan

Department of Mathematics
Simon Fraser University
British Columbia

Hensel lifting in $\mathbb{Z}[x]$ – two factor case

Input: $a, f_0, g_0 \in \mathbb{Z}[x]$, prime p, $m \in \mathbb{N}$ s.t.
(i) $a \equiv f_0 g_0 \pmod{p}$ and (ii) $\gcd(f_0, g_0) = 1$ in $\mathbb{F}_p[x]$.

Output: $f, g \in \mathbb{Z}[x]$ satisfying
(i) $a \equiv fg \pmod{p^m}$ and (ii) $f \equiv f_0 \pmod{p}$ and $g \equiv g_0 \pmod{p}$.

Hensel lifting in $\mathbb{Z}[x]$ – two factor case

Input: $a, f_0, g_0 \in \mathbb{Z}[x]$, prime p, $m \in \mathbb{N}$ s.t.
(i) $a \equiv f_0 g_0 \pmod{p}$ and (ii) $\gcd(f_0, g_0) = 1$ in $\mathbb{F}_p[x]$.

Output: $f, g \in \mathbb{Z}[x]$ satisfying
(i) $a \equiv fg \pmod{p^m}$ and (ii) $f \equiv f_0 \pmod{p}$ and $g \equiv g_0 \pmod{p}$.

Hensel lifting in $\mathbb{F}_p[x, y]$ – two factor case

Input: $a \in \mathbb{F}_p[x, y]$, $f_0, g_0 \in \mathbb{F}_p[x]$, $\alpha \in \mathbb{F}_p$, $m \in \mathbb{N}$ s.t.
(i) $a(x, \alpha) = f_0 g_0$ and (ii) $\gcd(f_0, g_0) = 1$.

Output: $f, g \in \mathbb{F}_p[x, y]$ satisfying
(i) $a \equiv fg \pmod{(y - \alpha)^m}$ and (ii) $f(x, \alpha) = f_0$ and $g(x, \alpha) = g_0$.

Note: in Hensel lifting we want to stop lifting if $a - fg = 0$.

Main Result: Let $a = fg$ in $\mathbb{F}_p[x, y]$ and $\alpha \in \mathbb{F}_p$.
Let $d_x = \deg(a, x)$ and $d_y = \deg(a, y)$.
$\texttt{HenselLift}(a, f_0 := f(x, \alpha), g_0 := g(x, \alpha), \alpha, m := d_y)$.

	Classical	Fast Arithmetic
Multiply $f \times g$	$O(d_x^2 d_y^2)$	$M(d_x d_y)$
Linear lifting	$O(d_x^2 d_y^2)$	$O(M(d_x) d_y^2)$
Quadratic lifting	$O(d_x^2 d_y^2)$	$28M(d_x d_y)$

Main Result: Let $a = fg$ in $\mathbb{F}_p[x, y]$ and $\alpha \in \mathbb{F}_p$.
Let $d_x = \deg(a, x)$ and $d_y = \deg(a, y)$.
$\texttt{HenselLift}(a, f_0 := f(x, \alpha), g_0 := g(x, \alpha), \alpha, m := d_y)$.

	Classical	Fast Arithmetic
Multiply $f \times g$	$O(d_x^2 d_y^2)$	$M(d_x d_y)$
Linear lifting	$O(d_x^2 d_y^2)$	$O(M(d_x) d_y^2)$
Quadratic lifting	$O(d_x^2 d_y^2)$	$28M(d_x d_y)$
Our cubic result	$O(d_x d_y^2 + d_x^2 d_y)$	$O(d_x d_y^2 + d_y M(d_x))$

- How does it work?
- Is it faster in practice?
- Does it work for $\mathbb{Z}[x]$?
- What can we use it for?
- More than 2 factors?

Linear Hensel lifting for $a \in \mathbb{F}_p[x, y]$ monic in x computes

Initialize $f \leftarrow f_0; g \leftarrow g_0; k \leftarrow 1; e \leftarrow a - fg$
while $k < m$ and $e \neq 0$ **do**
  $c_k \leftarrow \operatorname{coeff}(e, (y-\alpha)^k)$ ..... $O(d_x d_y)$
  Solve $f_k g_0 + g_k f_0 = c_k$ for f_k, g_k in $\mathbb{F}_p[x]$ ..... $O(d_x^2)$
  $f \leftarrow f + f_k (y-\alpha)^k$ ..... $O(k d_x)$
  $g \leftarrow g + g_k (y-\alpha)^k$ ..... $O(k d_x)$
  $k \leftarrow k+1; e \leftarrow a - fg$ ..... $O(k^2 d_x^2)$
end while

Linear Hensel lifting for $a \in \mathbb{F}_p[x, y]$ monic in x computes

Initialize $f \leftarrow f_0; g \leftarrow g_0; k \leftarrow 1; e \leftarrow a - fg$
while $k < m$ and $e \neq 0$ **do**
  $c_k \leftarrow \text{coeff}(e, (y-\alpha)^k)$ ..... $O(d_x d_y)$
  Solve $f_k g_0 + g_k f_0 = c_k$ for f_k, g_k in $\mathbb{F}_p[x]$ ..... $O(d_x^2)$
  $f \leftarrow f + f_k (y-\alpha)^k$ ..... $O(k d_x)$
  $g \leftarrow g + g_k (y-\alpha)^k$ ..... $O(k d_x)$
  $k \leftarrow k+1; e \leftarrow a - fg$ ..... $O(k^2 d_x^2)$
end while

Bernardin (Issac 1998) computes c_k without computing e as follows:

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

$$\begin{aligned} c_k &= coeff(e, (y-\alpha)^k) \\ &= coeff(a - f^{(k)} g^{(k)}, (y-\alpha)^k) \\ &= coeff(a, (y-\alpha)^k) - coeff(f^{(k)} g^{(k)}, (y-\alpha)^k) \\ &= a_k(x) - \sum_{i=1}^{k-1} f_i(x) g_{k-i}(x). \end{aligned}$$

How can we compute $\Delta = \sum_{i=1}^{k-1} f_i(x) g_{k-i}(x)$ where $\deg(f_i\, g_{k-i}, x) < d_x$?

Cost $\leq \sum_{k=1}^{d_y-1} (k-1) O(d_x^2) = O(d_y^2 d_x^2)$.

Bernardin [ISSAC 98] computes $f_i g_{k-i}$ in parallel using Karatsuba.

How can we compute $\Delta = \sum_{i=1}^{k-1} f_i(x) g_{k-i}(x)$ where $\deg(f_i\, g_{k-i}, x) < d_x$?

Cost $\leq \sum_{k=1}^{d_y-1} (k-1) O(d_x^2) = O(d_y^2 d_x^2)$.

Bernardin [ISSAC 98] computes $f_i g_{k-i}$ in parallel using Karatsuba.

We compute Δ by evaluation and interpolation in x.
Since $\deg(\Delta, x) < d_x$ we use d_x points.

For $0 \leq \beta < d_x$ do $\Delta_j \leftarrow \sum_{i=1}^{k-1} f_i(\beta) g_{k-i}(\beta)$ ..... $O(k d_x)$
Interpolate $\Delta(x)$ and set $c_k \leftarrow a_k - \Delta$ ..... $O(d_x^2)$
Solve $f_k g_0 + g_k f_0 = c_k$ for f_k and g_k ..... $O(d_x^2)$
Compute $f_k(\beta)$ and $g_k(\beta)$ for $0 \leq \beta < d_x$ ..... $O(d_x^2)$

Cost $\sum_{k=1}^{d_y-1} O(d_x^2) + \sum_{k=1}^{d_y-1} O(k d_x) = O(d_y d_x^2 + d_x d_y^2)$.
Need to remember $f_k(\beta)$ and $g_k(\beta)$ for $0 \leq \beta < d_x$ and $1 \leq k < d_y$.
Requires $p \geq d_x$. If $p < d_x$ then ...

How good is it?

For $a = fg$ in $\mathbb{F}_p[x, y]$ where $p = 2^{31} - 1$ and
$d = \deg(f, x) = \deg(g, x) = \deg(f, y) = \deg(g, y)$

d	LHL $O(d^4)$	New $O(d^3)$ Linear Lift				Fast QHL in Magma	
		New1 (Δ)		New2 (Δ)			
10	0.14ms	0.22ms	(0.14)	0.17ms	(0.07)	10.9ms	
15	0.35ms	0.57ms	(0.37)	0.34ms	(0.11)	35.7ms	
20	0.75ms	1.23ms	(0.85)	0.63ms	(0.25)	48.9ms	
40	6.58ms	8.57ms	(5.49)	3.22ms	(0.98)	244ms	
60	26.7ms	27.7ms	(22.2)	8.70ms	(2.56)	464ms	
100	166ms	126ms	(103)	34.2ms	(10.2)	1.62s	
200	2.15s	992ms	(834)	230ms	(65)	8.70s	
400	29.5s	7.91s	(6.71)	1.63s	(0.49s)	43.9s	
600	140s	26.5s	(22.6)	6.41s	(1.50s)	167.9s	(0.78gb)
800	425s	63.4s	(53.8)	13.9s	(3.87s)	241.7s	(1.04gb)
1000	1017s	125s	(109)	26.7s	(7.41s)	349.8s	(1.48gb)
2000	NA	1000s		207s	(0.69gb)	2054.8s	(5.12gb)
4000	NA	NA		1704s	(4.35gb)	9255.7s	(22.8gb)
6000	NA	NA		5438s	(6.20gb)	36500.s	(68.2gb)
8000	NA	NA		13035s	(11.0gb)	NA	

Optimization 1: Evaluate and Interpolate at $x = \pm 1, \pm 2, \pm 3, \ldots$

Optimization 2: Use an accumulator to eliminate multiplications modulo p.

To compute $S = \sum_{i=1}^{k-1} a_i b_{k-i}$ in $\mathbb{F}_p$ for $p < 2^{63}$ use

$$S = \left(\sum_{i=1}^{k-1} (a_i b_i \bmod m) \right) \bmod p \quad \text{where} \quad m = 2^{64} \times p$$

This is 3.5 to 4 times faster than Möller and Grandlund 2011.

The paper shows how to re-organize evaluation and Lagrange interpolation so we can use both optimizations simultaneously.

What happens for Hensel lifting $\mathbb{Z}[x]$?

1 Compute $\Delta = \sum_{i=1}^{k-1} f_i(x) g_{k-i}(x)$ in $\mathbb{Z}[x]$ where

$\|f_i\| < p$ and $\|g_i\| < p$ hence $\|\Delta\| < \deg(a, x)p^2$.

Use Chinese remaindering: For $p < 2^{63}$ use the same evaluation/interpolation approach mod three 50 bit primes q_1, q_2, q_3.
This increases the cost of computing Δ by a factor of 3.

2 Must also reorganize the computation of

$$e_{k+1} = \frac{a - f^{(k)} g^{(k)}}{p^k} \quad \text{in} \quad \mathbb{Z}[x]$$

3 The paper treats the non-monic case.
Can be done in $O(d^2 m + m^2 d)$ where $d = \deg(a, x)$ and $10^m > \max(\|a\|, \|f\|, \|g\|)$.

Timings in CPU seconds for linear Hensel Lifting in $\mathbb{Z}[x]$ using $p = 2^{50} - 27$.
For $a = fg$ in $\mathbb{Z}[x]$ where $d = \deg(f, x) = \deg(g, x)$ and $\|f\| < p^m, \|g\| < p^m$

	Linear HL coded in C				Quadratic HL in Magma		
d / m	quartic	(Δ)	cubic	(Δ)	$f \times g$	QHL1	QHL2
25 / 25	.0012	(.0005)	0.0025	(.0005)	0.0002	0.009	0.015
50 / 50	.0101	(.0072)	0.0106	(.0028)	0.0012	0.048	0.062
100 / 100	0.124	(0.108)	0.051	(0.015)	0.0068	0.301	0.294
200 / 200	1.774	(1.685)	0.263	(0.080)	0.032	1.67	1.638
400 / 400	26.66	(26.13)	1.500	(0.650)	0.180	9.82	10.70
600 / 600	138.2	(136.1)	6.310	(3.900)	0.643	50.17	50.44
800 / 800	429.4	(424.7)	10.93	(6.350)	0.971	59.17	65.98
1000/ 1000	1052.	(1042.)	17.16	(9.070)	1.39	69.90	74.92
2000/ 2000	NA		119.2	(69.46)	6.51	395.9	425.1
4000/ 4000	NA		880.6	(531.3)	29.69	2,238	2,345
8000/ 8000	NA		7,055	(4480.)	129.33	11,073	12,287
10000 / 10000	NA		19,086	(14008)	281.29	41,265	46,646
12000 / 12000	NA		29,462	(20814)	434.52	45,528	54,341
16000 / 16000	NA		59,295	(39408)	562.17	>64gb	>64gb

Application 1: GCD Computation in $\mathbb{Z}[x]$.

Let $a, b \in \mathbb{Z}[x]$ and $g = \gcd(a, b)$.
Let $d \geq \max(\deg a, \deg b)$ and $10^m > \max(\|a\|, \|b\|, \|g\|)$.

Maple and Magma use the $O(dm^2 + md^2)$ modular GCD algorithm.
Why? Because linear Hensel lifting (Miola and Yun 1974) is $O(d^2m^2)$
But our LHL is $O(dm^2 + md^2)$!!

Application 2: Polynomial Factorization in $\mathbb{Z}[x_1, x_2, \ldots, x_n]$.

Let $a = fg$ in $\mathbb{Z}[x_1, x_2, \ldots, x_n]$. To factor a Monagan and Tuncer 2018 reduce to many Hensel lifts in $\mathbb{F}_p[x, y]$ for a machine prime p. Here the degrees are usually under 100 and rarely over 1000.

What if $a = f_1 f_2 \ldots f_n$ and $n > 2$?

What if $a = f_1 f_2 \ldots f_n$ and $n > 2$?

See the ISSAC 2019 Poster: G. Paluck and M. Monagan
New Bivariate Hensel lifting algorithm for n factors.

Thank You.