Factoring multivariate sparse polynomials

Joris van der Hoeven

Joint work with Alexander Demin and Grégoire Lecerf

CNRS, École polytechnique, France

Funded by the European Union

European Research Council
Established by the European Commission

ODELIX thematic program, Palaiseau

November 14, 2025

Sparse polynomials — example 1

Symbolic expressions $\longrightarrow$ sparse polynomials or rational functions

Sparse polynomials — example 1

Symbolic expressions ⟶ sparse polynomials or rational functions

```
Caas] derive (x^x^x, x, x, x, x)
```

```
Caas]
```

Sparse polynomials — example 1

Symbolic expressions $\longrightarrow$ sparse polynomials or rational functions

```
Caas] derive (x^x^x, x, x, x, x)
```

$$\Bigg(6\,\frac{(\log(x)+1)^2 x^x \log(x)}{x} - 4\,\frac{(\log(x)+1)\,x^x \log(x)}{x^2} - 6\,\frac{(\log(x)+1)^2 x^x}{x^2} + (\log(x)+$$

$$1)^4\,x^x\log(x) + 4\,\frac{(\log(x)+1)^3 x^x}{x} + 8\,\frac{(\log(x)+1)\,x^x}{x^3} + 12\,\frac{(\log(x)+1)\,x^x}{x^2} + 2\,\frac{x^x\log(x)}{x^3} +$$

$$3\,\frac{x^x\log(x)}{x^2} - 6\,\frac{x^x}{x^4} - 10\,\frac{x^x}{x^3}\Bigg)\,x^{x^x} + 4\,\Bigg((\log(x)+1)^3\,x^x\log(x) + 3\,\frac{(\log(x)+1)\,x^x\log(x)}{x} +$$

$$3\,\frac{(\log(x)+1)^2 x^x}{x} - 3\,\frac{(\log(x)+1)\,x^x}{x^2} - \frac{x^x\log(x)}{x^2} + 2\,\frac{x^x}{x^3} + 3\,\frac{x^x}{x^2}\Bigg)\Bigg((\log(x)+1)\,x^x\log(x) +$$

$$\frac{x^x}{x}\Bigg)\,x^{x^x} + 6\,\Bigg((\log(x)+1)^2 x^x\log(x) + 2\,\frac{(\log(x)+1)\,x^x}{x} + \frac{x^x\log(x)}{x} - \frac{x^x}{x^2}\Bigg)\Bigg((\log(x)+$$

$$1)\,x^x\log(x) + \frac{x^x}{x}\Bigg)^2 x^{x^x} + 3\,\Bigg((\log(x)+1)^2 x^x\log(x) + 2\,\frac{(\log(x)+1)\,x^x}{x} + \frac{x^x\log(x)}{x} - \frac{x^x}{x^2}\Bigg)^2 x^{x^x} +$$

$$\Bigg((\log(x)+1)\,x^x\log(x) + \frac{x^x}{x}\Bigg)^4 x^{x^x}$$

Caas]

Sparse polynomials — example 1

Symbolic expressions ⟶ sparse polynomials or rational functions

```
[aas] derive (x^x^x, x, x, x)
```

Sparse polynomials — example 1

Symbolic expressions $\longrightarrow$ sparse polynomials or rational functions

```
[aas] derive (x^x^x, x, x, x)
```

$$\left((\log(x)+1)^3 x^x \log(x) + 3\,\frac{(\log(x)+1)\,x^x \log(x)}{x} + 3\,\frac{(\log(x)+1)^2 x^x}{x} - 3\,\frac{(\log(x)+1)\,x^x}{x^2} - \frac{x^x \log(x)}{x^2} + 2\,\frac{x^x}{x^3} + 3\,\frac{x^x}{x^2}\right) x^{x^x} + 3\left((\log(x)+1)^2 x^x \log(x) + 2\,\frac{(\log(x)+1)\,x^x}{x} + \frac{x^x \log(x)}{x} - \frac{x^x}{x^2}\right)\left((\log(x)+1)\,x^x \log(x) + \frac{x^x}{x}\right) x^{x^x} + \left((\log(x)+1)\,x^x \log(x) + \frac{x^x}{x}\right)^3 x^{x^x}$$

Sparse polynomials — example 1

Symbolic expressions $\longrightarrow$ sparse polynomials or rational functions

```
[aas] derive (x^x^x, x, x, x)
```

$$\left((\log(x)+1)^3 x^x \log(x) + 3\frac{(\log(x)+1)\,x^x \log(x)}{x} + 3\frac{(\log(x)+1)^2 x^x}{x} - 3\frac{(\log(x)+1)\,x^x}{x^2} - \frac{x^x \log(x)}{x^2} + 2\frac{x^x}{x^3} + 3\frac{x^x}{x^2}\right) x^{x^x} + 3\left((\log(x)+1)^2 x^x \log(x) + 2\frac{(\log(x)+1)\,x^x}{x} + \frac{x^x \log(x)}{x} - \frac{x^x}{x^2}\right)\left((\log(x)+1)\,x^x \log(x) + \frac{x^x}{x}\right) x^{x^x} + \left((\log(x)+1)\,x^x \log(x) + \frac{x^x}{x}\right)^3 x^{x^x}$$

$$f = x^{x^x}$$

$$f', f'', f''', \ldots \in \mathbb{Q}\left[\log x, \tfrac{1}{x}, x^x, x^{x^x}\right]$$

Sparse polynomials — example 1

Symbolic expressions $\longrightarrow$ sparse polynomials or rational functions

```
[aas] derive (x^x^x, x, x, x)
```

$$\left((\log(x)+1)^3 x^x \log(x) + 3\frac{(\log(x)+1)\,x^x\log(x)}{x} + 3\frac{(\log(x)+1)^2 x^x}{x} - 3\frac{(\log(x)+1)\,x^x}{x^2} - \frac{x^x\log(x)}{x^2} + 2\frac{x^x}{x^3} + 3\frac{x^x}{x^2}\right) x^{x^x} + 3\left((\log(x)+1)^2 x^x\log(x) + 2\frac{(\log(x)+1)\,x^x}{x} + \frac{x^x\log(x)}{x} - \frac{x^x}{x^2}\right)\left((\log(x)+1)\,x^x\log(x) + \frac{x^x}{x}\right) x^{x^x} + \left((\log(x)+1)\,x^x\log(x) + \frac{x^x}{x}\right)^3 x^{x^x}$$

$$\begin{aligned} f &= x^{x^x} \\ f', f'', f''', \ldots &\in \mathbb{Q}\left[\log x, \tfrac{1}{x}, x^x, x^{x^x}\right] \\ &\subseteq \mathbb{Q}\left(\log x, x, x^x, x^{x^x}\right) \end{aligned}$$

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

```
Caas] det M
```

```
Caas] simplify invert M
```

```
Caas] simplify transpose invert transpose invert M
```

```
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
```

```
Caas] det M
```

$$(cg+fh)(-a-e)+(ac+bf)g+(cd+ef)h+(ae-bd)i$$

```
Caas] simplify invert M
```

```
Caas] simplify transpose invert transpose invert M
```

```
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
```

$$\left[\begin{matrix} \frac{ei-fh}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{ch-bi}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{bf-c}{(dh-eg)c+(ei-f} \\ \frac{fg-di}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{ai-cg}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{cd-a}{(dh-eg)c+(ei-f} \\ \frac{dh-eg}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{bg-ah}{(dh-eg)c+(ei-fh)a+(fg-di)b} & \frac{ae-b}{(dh-eg)c+(ei-f} \end{matrix}\right.$$

```
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
```

```
Caas] det M
```

```
Caas] simplify invert M
```

```
Caas] simplify transpose invert transpose invert M
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

```
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Sparse polynomials — example 2


```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Computations with parameters ⟶ expression swell

Sparse polynomials — example 2

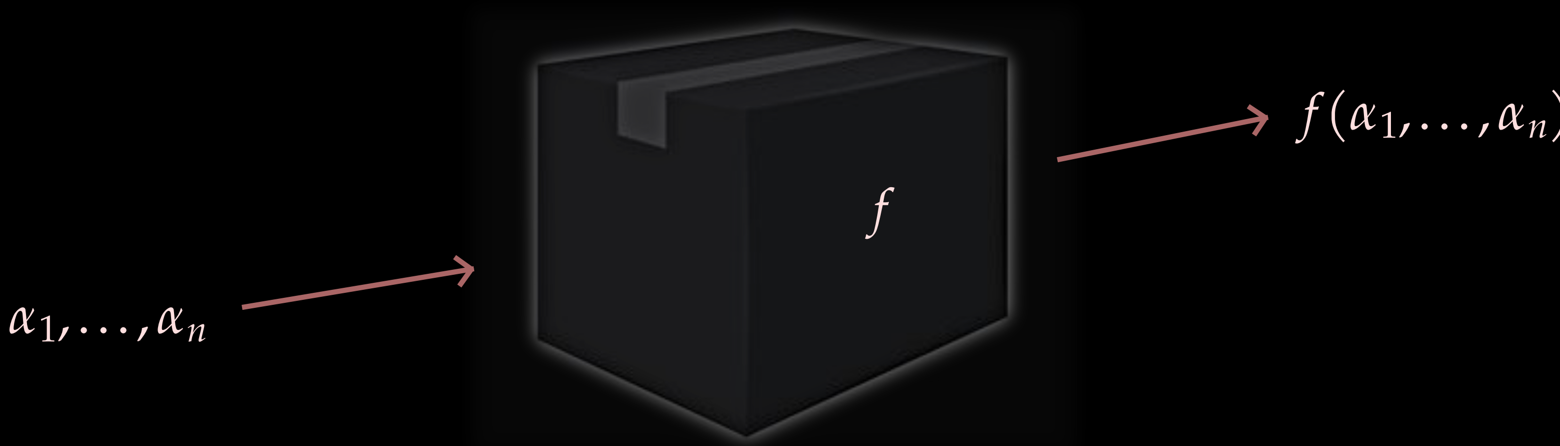

```
Caas] M == [ a, b, c; d, e, f; g, h, i ]
Caas] det M
Caas] simplify invert M
Caas] simplify transpose invert transpose invert M
Caas]
```

Computations with parameters ⟶ expression swell

How to make running times depend only on the input & output size ?

Sparse interpolation

Input



Output

$$f(x_1,\ldots,x_n) \;=\; c_1 x_1^{e_{1,1}} \cdots x_n^{e_{1,n}} + \cdots + c_t x_1^{e_{t,1}} \cdots x_n^{e_{t,n}}$$

Complexity of sparse interpolation (see previous talk by Grégoire)

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L

OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) = O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) = \tilde{O}(t)$$

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L

OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) \;=\; O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) \;=\; \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) \;=\; O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) \;=\; \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) = O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) = \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

- Greatest common divisors

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) = O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) = \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

- Greatest common divisors
- Sparse interpolation of rational functions

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) = O(Lt) + \mathsf{S}(t), \qquad \mathsf{S}(t) = \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

- Greatest common divisors
- Sparse interpolation of rational functions
- Factorization

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) = O(L t) + \mathsf{S}(t), \qquad \mathsf{S}(t) = \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

- Greatest common divisors
- Sparse interpolation of rational functions
- Factorization

Long history ⟶ paper with Alexander Demin

Sparse interpolation

Complexity of sparse interpolation (see previous talk by Grégoire)

INPUT: an SLP f of size L
OUTPUT: sparse interpolation $f = c_1 \boldsymbol{x}^{e_1} + \cdots + c_t \boldsymbol{x}^{e_t}$

$$\mathsf{T}(t) \;=\; O(Lt) + \mathsf{S}(t), \qquad\qquad \mathsf{S}(t) \;=\; \tilde{O}(t)$$

CONSTRAINT: evaluation points → roots of unity or geometric progression

Other operations ?

- Greatest common divisors
- Sparse interpolation of rational functions
- Factorization

Long history ⟶ paper with Alexander Demin

We will focus on the case where the total degree d is “modest”

Sparse gcd computations

$$G(x_1,\ldots,x_n) \;=\; \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned}$$

U, V coprime

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned}$$

U, V coprime

$$\begin{aligned} A(\alpha_1 t,\ldots,\alpha_n t) &= U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) \\ B(\alpha_1 t,\ldots,\alpha_n t) &= V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) \end{aligned}$$

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned}$$

U, V coprime

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

Sparse gcd computations

$$G(x_1,\dots,x_n) = \gcd\big(A(x_1,\dots,x_n), B(x_1,\dots,x_n)\big)$$

$$\begin{aligned} A(x_1,\dots,x_n) &= U(x_1,\dots,x_n)\, G(x_1,\dots,x_n) \\ B(x_1,\dots,x_n) &= V(x_1,\dots,x_n)\, G(x_1,\dots,x_n) \end{aligned} \qquad U,V \text{ coprime}$$

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\dots,\alpha_n t) = U(\alpha_1 t,\dots,\alpha_n t)\, G(\alpha_1 t,\dots,\alpha_n t) = U_\alpha(t)\, G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\dots,\alpha_n t) = V(\alpha_1 t,\dots,\alpha_n t)\, G(\alpha_1 t,\dots,\alpha_n t) = V_\alpha(t)\, G_\alpha(t) \end{aligned}$$

With high probability

$$G_\alpha(t) \overset{\text{HP}}{=\!=\!=} \gcd\big(A_\alpha(t), B_\alpha(t)\big)$$

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned} \qquad U, V \text{ coprime}$$

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

With high probability

$$G_\alpha(t) \overset{\text{HP}}{=\!=\!=} \gcd(A_\alpha(t), B_\alpha(t))$$

Why?

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned} \qquad U,V \text{ coprime}$$

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

With high probability (over Laurent polynomials)

$$G_\alpha(t) \overset{\text{HP}}{=\!=} \gcd(A_\alpha(t), B_\alpha(t))$$

Indeed,

$$U_\alpha, V_\alpha \text{ not coprime} \iff \mathrm{Res}_t(U_\alpha(t), V_\alpha(t)) = 0$$

Sparse gcd computations

$$G(x_1,\ldots,x_n) = \gcd\left(A(x_1,\ldots,x_n), B(x_1,\ldots,x_n)\right)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned}$$

U, V coprime

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

With high probability

$$G_\alpha(t) \overset{\text{HP}}{=\!=} \gcd(A_\alpha(t), B_\alpha(t))$$

Sparse gcd computations

$$G(x_1,\ldots,x_n) \;=\; \gcd\big(A(x_1,\ldots,x_n),B(x_1,\ldots,x_n)\big)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned} \qquad U,V \text{ coprime}$$

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

With high probability

$$G_\alpha(t) \overset{\text{HP}}{=\!=\!=} \gcd(A_\alpha(t),B_\alpha(t))$$

Normalization problem

$G_\alpha(t)$ defined up to a “constant” c_α

Sparse gcd computations

$$G(x_1,\ldots,x_n) \;=\; \gcd\big(A(x_1,\ldots,x_n),B(x_1,\ldots,x_n)\big)$$

$$\begin{aligned} A(x_1,\ldots,x_n) &= U(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \\ B(x_1,\ldots,x_n) &= V(x_1,\ldots,x_n)\,G(x_1,\ldots,x_n) \end{aligned} \qquad U,V \text{ coprime}$$

$$\begin{aligned} A_\alpha(t) &= A(\alpha_1 t,\ldots,\alpha_n t) = U(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = U_\alpha(t)\,G_\alpha(t) \\ B_\alpha(t) &= B(\alpha_1 t,\ldots,\alpha_n t) = V(\alpha_1 t,\ldots,\alpha_n t)\,G(\alpha_1 t,\ldots,\alpha_n t) = V_\alpha(t)\,G_\alpha(t) \end{aligned}$$

With high probability

$$G_\alpha(t) \overset{\mathrm{HP}}{=\!=\!=} \gcd(A_\alpha(t),B_\alpha(t))$$

Normalization problem

$$G_\alpha(t) \quad \text{defined up to a “constant”} \quad c_\alpha$$

⟶ no completely specified SLP to compute the coefficients of $G_\alpha(t)$

A lucky example when normalization is easy

$$A = -2025\,xyz - 14\,x^4y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1$$
$$B = 14\,x^3z + 11\,y^2z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1$$

A lucky example when normalization is easy

$$A = -2025\,xyz - 14\,x^4y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1$$
$$B = 14\,x^3z + 11\,y^2z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1$$

A lucky example when normalization is easy

$$A = -2025xyz - 14x^4y - 11xy^3 - xy + 14x^3 + 11y^2 + 2025z + 1$$
$$B = 14x^3z + 11y^2z + 14x^3 + 11y^2 + 2025z^2 + 2026z + 1$$

We can normalize as follows:

$$A = UG \qquad U = \cdots + 1$$
$$B = VG \qquad V = \cdots + 1$$
$$G = \cdots + 1$$

A lucky example when normalization is easy

$$A = -2025\,xyz - 14\,x^4y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1$$
$$B = 14\,x^3z + 11\,y^2z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1$$

We can normalize as follows:

$$A = UG \qquad U = \cdots + 1 \qquad G = \cdots + 1$$
$$B = VG \qquad V = \cdots + 1$$

$$G_\alpha(t) = G_{\alpha,\deg G}\,t^{\deg G} + \cdots + G_{\alpha,1}\,t + 1$$

A lucky example when normalization is easy

$$A = -2025\,xyz - 14\,x^4 y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1$$
$$B = 14\,x^3 z + 11\,y^2 z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1$$

We can normalize as follows:

$$A = UG \qquad U = \cdots + 1$$
$$B = VG \qquad V = \cdots + 1 \qquad G = \cdots + 1$$

$$G_\alpha(t) = G_{\alpha,\deg G}\, t^{\deg G} + \cdots + G_{\alpha,1}\, t + 1$$

- There is only one gcd of $A_\alpha(t)$ and $B_\alpha(t)$ of this form

A lucky example when normalization is easy

$$\begin{aligned} A &= -2025\,xyz - 14\,x^4y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1 \\ B &= 14\,x^3z + 11\,y^2z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1 \end{aligned}$$

We can normalize as follows:

$$\begin{aligned} A &= UG \\ B &= VG \end{aligned} \qquad \begin{aligned} U &= \cdots + 1 \\ V &= \cdots + 1 \end{aligned} \qquad G = \cdots + 1$$

$$G_\alpha(t) = G_{\alpha,\deg G}\,t^{\deg G} + \cdots + G_{\alpha,1}\,t + 1$$

- There is only one gcd of $A_\alpha(t)$ and $B_\alpha(t)$ of this form
- There is$_{\text{HP}}$ an efficient SLP to compute it

A lucky example when normalization is easy

$$A = -2025\,xyz - 14\,x^4 y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1$$
$$B = 14\,x^3 z + 11\,y^2 z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1$$

We can normalize as follows:

$$A = UG \qquad U = \cdots + 1$$
$$B = VG \qquad V = \cdots + 1 \qquad G = \cdots + 1$$

$$G_\alpha(t) = G_{\alpha,\deg G}\, t^{\deg G} + \cdots + G_{\alpha,1}\, t + 1$$

- There is only one gcd of $A_\alpha(t)$ and $B_\alpha(t)$ of this form
- There is$_{\text{HP}}$ an efficient SLP to compute it, of length $L = \tilde{O}(\deg G) = \tilde{O}(d)$

A lucky example when normalization is easy

$$\begin{aligned} A &= -2025\,xyz - 14\,x^4y - 11\,xy^3 - xy + 14\,x^3 + 11\,y^2 + 2025\,z + 1 \\ B &= 14\,x^3z + 11\,y^2z + 14\,x^3 + 11\,y^2 + 2025\,z^2 + 2026\,z + 1 \end{aligned}$$

We can normalize as follows:

$$\begin{aligned} A &= UG \\ B &= VG \end{aligned} \qquad \begin{aligned} U &= \cdots + 1 \\ V &= \cdots + 1 \end{aligned} \qquad G = \cdots + 1$$

$$G_\alpha(t) = G_{\alpha,\deg G}\, t^{\deg G} + \cdots + G_{\alpha,1}\, t + 1$$

- There is only one gcd of $A_\alpha(t)$ and $B_\alpha(t)$ of this form
- There is$_{\text{HP}}$ an efficient SLP to compute it, of length $L = \tilde{O}(\deg G) = \tilde{O}(d)$
- $\longrightarrow$  $\longrightarrow$ $G = 14\,x^3 + 11\,y^2 + 2025\,z + 1$

Regularizing weights

Standard projection

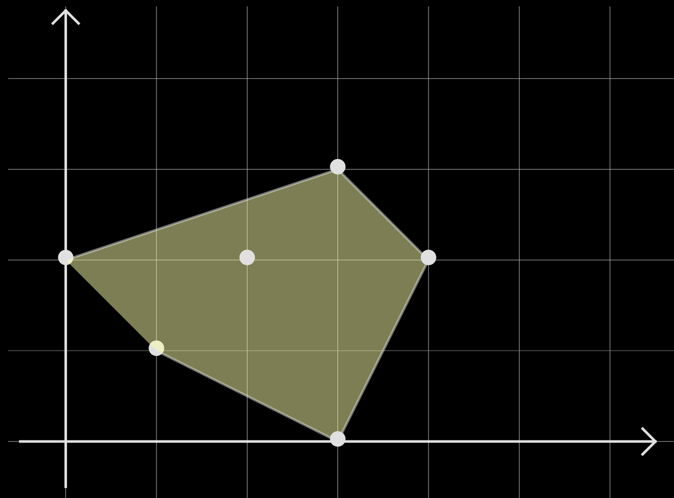
$$A(x_1,\ldots,x_n) \longrightarrow A(\alpha_1 t,\ldots,\alpha_n t)$$
$$B(x_1,\ldots,x_n) \longrightarrow B(\alpha_1 t,\ldots,\alpha_n t)$$

Regularizing weights

Standard projection

$$A(x_1,\ldots,x_n) \longrightarrow A(\alpha_1 t,\ldots,\alpha_n t)$$
$$B(x_1,\ldots,x_n) \longrightarrow B(\alpha_1 t,\ldots,\alpha_n t)$$

Newton polytopes



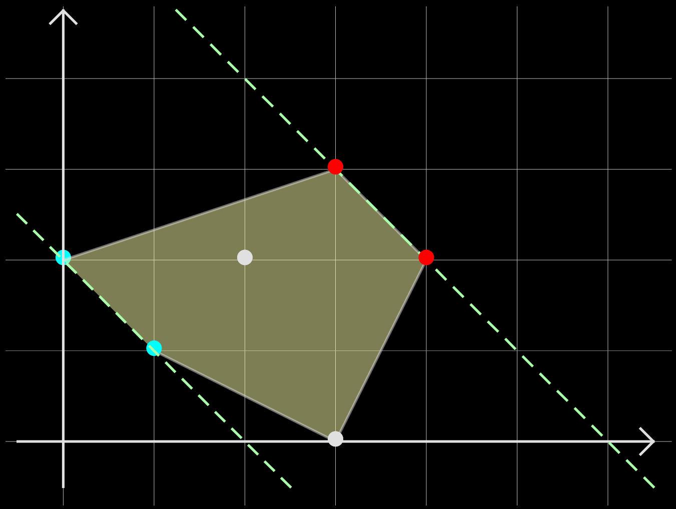
$$A(x,y) = x^4y^2 + x^3y^3 - x^2y^2 - x^3 + xy - 2y^2$$

Regularizing weights

Standard projection

$$A(x_1,\dots,x_n) \longrightarrow A(\alpha_1 t,\dots,\alpha_n t)$$
$$B(x_1,\dots,x_n) \longrightarrow B(\alpha_1 t,\dots,\alpha_n t)$$

Newton polytopes



$$A(x,y) = x^4y^2 + x^3y^3 - x^2y^2 - x^3 + xy - 2y^2$$

$$A_\alpha(t) = \square t^6 + \square t^4 + \square t^3 + \square t^2$$

Weighted projection with integer weights

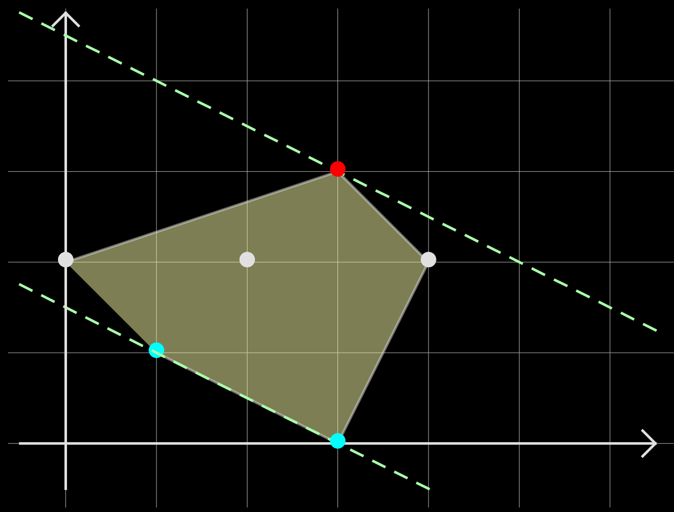
$$A(x_1,\ldots,x_n) \longrightarrow A(\alpha_1 t^{w_1},\ldots,\alpha_n t^{w_n})$$
$$B(x_1,\ldots,x_n) \longrightarrow B(\alpha_1 t^{w_1},\ldots,\alpha_n t^{w_n})$$

Regularizing weights

Weighted projection with integer weights

$$A(x_1,\dots,x_n) \longrightarrow A(\alpha_1 t^{w_1},\dots,\alpha_n t^{w_n})$$
$$B(x_1,\dots,x_n) \longrightarrow B(\alpha_1 t^{w_1},\dots,\alpha_n t^{w_n})$$

Newton polytopes



Head regularizing weight $w = (1,2)$

$$A(x,y) = x^4y^2 + x^3y^3 - x^2y^2 - x^3 + xy - 2y^2$$

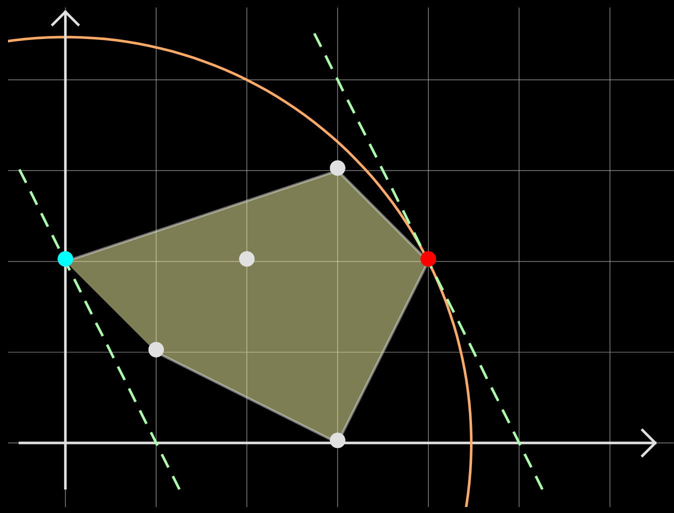
$$A_\alpha(t) = \square t^9 + \square t^8 + \square t^6 + \square t^4 + \square t^3$$

Regularizing weights

Weighted projection with integer weights

$$A(x_1,\dots,x_n) \longrightarrow A(\alpha_1 t^{w_1},\dots,\alpha_n t^{w_n})$$
$$B(x_1,\dots,x_n) \longrightarrow B(\alpha_1 t^{w_1},\dots,\alpha_n t^{w_n})$$

Existence of regularizing weight of small norm $|w| \leqslant d$



Regularizing weight $w = (4,2)$

$$A(x,y) = x^4y^2 + x^3y^3 - x^2y^2 - x^3 + xy - 2y^2$$

$$A_\alpha(t) = \square t^{20} + \square t^{18} + \square t^{12} + \square t^6 + \square t^4$$

Complexity of sparse gcd computations

Notation

s_P: number of terms of $P \in \mathbb{K}[x_1, \ldots, x_n]$
d_P: total degree of P
$\mathrm{ec}_w P = \deg_w P - \mathrm{val}_w P$: weighted écart of P

$\mathsf{S}(s)$ complexity of sparse interpolation (for s terms)
$\mathsf{M}(d)$ complexity of dense polynomial multiplication (for degree d)

Theorem

Let $A, B \in \mathbb{K}[x_1, \ldots, x_n]$, $G = \gcd(A, B)$, and w regularizing for A or B.
Let $s := s_G$, $\bar{s} := s_A + s_B + s_G$, $d := \max(d_A, d_B)$, and $e := \max(\mathrm{ec}_w A, \mathrm{ec}_w B) \leqslant d^2$.
Then there is an algorithm to compute G with high probability, in time

$$O((\bar{s}/s + e)\,\mathsf{S}(s) + s\,\mathsf{M}(e)\log e).$$

An iterative alternative

Goal

$$G(x_1, \ldots, x_{n-1}, x_n) \;=\; \gcd\left(A(x_1, \ldots, x_{n-1}, x_n), B(x_1, \ldots, x_{n-1}, x_n)\right)$$

An iterative alternative

Goal

$$G(x_1,\ldots,x_{n-1},x_n) \;=\; \gcd\left(A(x_1,\ldots,x_{n-1},x_n),B(x_1,\ldots,x_{n-1},x_n)\right)$$

Fibers for sparse interpolation

$$\underbrace{G(\alpha_1,\ldots,\alpha_{n-1},t)}_{G_\alpha(t)} \;=\; \gcd\left(\underbrace{A(\alpha_1,\ldots,\alpha_{n-1},t)}_{A_\alpha(t)},\underbrace{B(\alpha_1,\ldots,\alpha_{n-1},t)}_{B_\alpha(t)}\right)$$

An iterative alternative

Goal

$$G(x_1,\dots,x_{n-1},x_n) \;=\; \gcd\left(A(x_1,\dots,x_{n-1},x_n),B(x_1,\dots,x_{n-1},x_n)\right)$$

Fibers for sparse interpolation

$$\underbrace{G(\alpha_1,\dots,\alpha_{n-1},t)}_{G_\alpha(t)} \;=\; \gcd\left(\underbrace{A(\alpha_1,\dots,\alpha_{n-1},t)}_{A_\alpha(t)},\underbrace{B(\alpha_1,\dots,\alpha_{n-1},t)}_{B_\alpha(t)}\right)$$

Recursively compute

$$G_c(x_1,\dots,x_{n-1}) \;=\; \gcd\left(A(x_1,\dots,x_{n-1},c),B(x_1,\dots,x_{n-1},c)\right)$$

An iterative alternative

Goal

$$G(x_1,\ldots,x_{n-1},x_n) = \gcd\left(A(x_1,\ldots,x_{n-1},x_n), B(x_1,\ldots,x_{n-1},x_n)\right)$$

Fibers for sparse interpolation

$$\underbrace{G(\alpha_1,\ldots,\alpha_{n-1},t)}_{G_\alpha(t)} = \gcd\left(\underbrace{A(\alpha_1,\ldots,\alpha_{n-1},t)}_{A_\alpha(t)}, \underbrace{B(\alpha_1,\ldots,\alpha_{n-1},t)}_{B_\alpha(t)}\right)$$

Recursively compute

$$G_c(x_1,\ldots,x_{n-1}) = \gcd\left(A(x_1,\ldots,x_{n-1},c), B(x_1,\ldots,x_{n-1},c)\right)$$

Iterative normalization

$$G_\alpha(t) = \gcd\left(A_\alpha(t), B_\alpha(t)\right)$$

$$G_\alpha(c) = G_c(x_1,\ldots,x_{n-1})$$

Complexity of the iterative algorithm

Notation

$\delta_P = \max(\deg_{x_1} P, \ldots, \deg_{x_n} P)$: maximum of partial degrees of P

Theorem

Let $A, B \in \mathbb{K}[x_1, \ldots, x_n]$ and $G = \gcd(A, B)$.

Let $s := s_G$, $\bar{s} := s_P + s_Q + s_G$, $d := \max(d_P, d_Q)$, and $\delta := \max(\delta_P, \delta_Q)$.

Then there is an algorithm to compute G with high probability, in time

$$O(n((\bar{s}/s + \delta)\,\mathsf{S}(s) + s\,\mathsf{M}(\delta)\log\delta)).$$

Related operations

Content factorization

$$\mathrm{cont}_z(xz - yz + x^2 - y^2 + x - y) \;=\; \gcd(x - y, x^2 - y^2 + x - y) \;=\; x - y$$

$$xz - yz + x^2 - y^2 + x - y \;=\; (x - y)(1 + x + y + z)$$

Related operations

Content factorization

$$\mathrm{cont}_z(xz - yz + x^2 - y^2 + x - y) = \gcd(x - y, x^2 - y^2 + x - y) = x - y$$

$$xz - yz + x^2 - y^2 + x - y = (x - y)(1 + x + y + z)$$

Root extraction

$$6xy^2z^2 - y^3z^3 - 12x^2yz + 8x^3 = (2x - yz)^3$$

Related operations

Content factorization

$$\mathrm{cont}_z(xz - yz + x^2 - y^2 + x - y) = \gcd(x - y, x^2 - y^2 + x - y) = x - y$$
$$xz - yz + x^2 - y^2 + x - y = (x - y)(1 + x + y + z)$$

Root extraction

$$6xy^2z^2 - y^3z^3 - 12x^2yz + 8x^3 = (2x - yz)^3$$

Square-free factorization

$$F = A_1 A_2^2 \cdots A_k^k, \qquad A_1, \ldots, A_k \text{ pairwise coprime}$$

E.g. repeat $\gcd(F, F')$ and root extraction.

Related operations

Content factorization

$$\mathrm{cont}_z(xz-yz+x^2-y^2+x-y) = \gcd(x-y, x^2-y^2+x-y) = x-y$$
$$xz-yz+x^2-y^2+x-y = (x-y)(1+x+y+z)$$

Root extraction

$$6xy^2z^2-y^3z^3-12x^2yz+8x^3 = (2x-yz)^3$$

Square-free factorization

$$F = A_1 A_2^2 \cdots A_k^k, \qquad A_1, \ldots, A_k \text{ pairwise coprime}$$

E.g. repeat $\gcd(F, F')$ and root extraction.

Sparse interpolation of rational function $f = \frac{A}{B}$

- Guess regularizing weight for A and B (fast to check), next as for gcd.
- Iterative approach (straightforward to adapt).

Lifting factorizations

$$G(\boldsymbol{x},t) = \gcd\left(A(\boldsymbol{x},t),B(\boldsymbol{x},t)\right)$$

$$\Big\Updownarrow \text{HP}$$

$$G(\alpha,t) = \gcd\left(A(\alpha,t),B(\alpha,t)\right)$$

Lifting factorizations

$$F(\boldsymbol{x},t) \quad = \quad A_1(\boldsymbol{x},t)\cdots A_\ell(\boldsymbol{x},t) \quad \text{irreducible factorization}$$

$$\not\Updownarrow \text{HP}$$

$$F(\boldsymbol{\alpha},t) \quad = \quad A_1(\boldsymbol{\alpha},t)\cdots A_\ell(\boldsymbol{\alpha},t) \quad \text{irreducible factorization}$$

Lifting factorizations

$$F(\boldsymbol{x},t) \quad = \quad A_1(\boldsymbol{x},t)\cdots A_\ell(\boldsymbol{x},t) \quad \text{irreducible factorization}$$

$$\not\Updownarrow \text{HP}$$

$$F(\boldsymbol{\alpha},t) \quad = \quad A_1(\boldsymbol{\alpha},t)\cdots A_\ell(\boldsymbol{\alpha},t) \quad \text{irreducible factorization}$$

Example. Over $\mathbb{K} = \mathbb{C}$, any univariate polynomial splits

$$F(\boldsymbol{x},t) \;=\; A_1(\boldsymbol{x},t)\cdots A_\ell(\boldsymbol{x},t) \quad \text{irreducible factorization}$$

$$\not\Updownarrow \text{HP}$$

$$F(\boldsymbol{\alpha},t) \;=\; A_1(\boldsymbol{\alpha},t)\cdots A_\ell(\boldsymbol{\alpha},t) \quad \text{irreducible factorization}$$

Example. Over $\mathbb{K}=\mathbb{C}$, any univariate polynomial splits

Example. Over $\mathbb{K}=\mathbb{F}_p$ with p odd consider

$$F(x,y,z) \;=\; \Phi(x,y)^2-z$$

$$\Phi(x,y) \;=\; x+y \qquad \text{(or } \Phi \text{ irreducible)}$$

A random $\alpha\in\mathbb{F}_p^*$ is square $\alpha=\beta^2$ with probability $\frac{1}{2}$.

In that case $F(x,y,\alpha)=(\Phi+\beta)\,(\Phi-\beta)$.

Lifting factorizations

$$F(\boldsymbol{x},t) = A_1(\boldsymbol{x},t)\cdots A_\ell(\boldsymbol{x},t) \quad \text{irreducible factorization}$$

$$\not\Updownarrow \text{HP}$$

$$F(\boldsymbol{\alpha},t) = A_1(\boldsymbol{\alpha},t)\cdots A_\ell(\boldsymbol{\alpha},t) \quad \text{irreducible factorization}$$

Example. Over $\mathbb{K}=\mathbb{C}$, any univariate polynomial splits

Example. Over $\mathbb{K}=\mathbb{F}_p$ with p odd consider

$$F(x,y,z) = \Phi(x,y)^2 - z$$
$$\Phi(x,y) = x+y \qquad \text{(or } \Phi \text{ irreducible)}$$

A random $\alpha \in \mathbb{F}_p^*$ is square $\alpha=\beta^2$ with probability $\frac{1}{2}$.

In that case $F(x,y,\alpha)=(\Phi+\beta)\,(\Phi-\beta)$.

Problem: how to recombine projected factors when lifting back?

Hilbert-Bertini irreducibility theorem

Hilbert-Bertini irreducibility Theorem

Assume $F \in \mathbb{K}[x_1, \ldots, x_n] \setminus \mathbb{K}$ *irreducible. Let* U *be the set of points* $(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n) \in \mathbb{K}^{3n}$ *for which*

$$F(\alpha_1 t + \beta_1 u + \gamma_1, \ldots, \alpha_n t + \beta_n u + \gamma_n)$$

is irreducible in $\mathbb{K}[t,u]$. *Then* U *is a Zariski open subset of* $\mathbb{K}^{3n}$, *which is dense over the algebraic closure of* $\mathbb{K}$.

Hilbert-Bertini irreducibility theorem

Hilbert-Bertini irreducibility Theorem

Assume $F \in \mathbb{K}[x_1, \ldots, x_n] \setminus \mathbb{K}$ *irreducible. Let* U *be the set of points* $(\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n) \in \mathbb{K}^{3n}$ *for which*

$$F(\alpha_1 t + \beta_1 u + \gamma_1, \ldots, \alpha_n t + \beta_n u + \gamma_n)$$

is irreducible in $\mathbb{K}[t, u]$*. Then* U *is a Zariski open subset of* $\mathbb{K}^{3n}$*, which is dense over the algebraic closure of* $\mathbb{K}$*.*

$\longrightarrow$ modulo random shifts, bivariate instead of univariate projections suffice

Hilbert-Bertini irreducibility theorem

Hilbert-Bertini irreducibility Theorem

Assume $F \in \mathbb{K}[x_1,\ldots,x_n] \setminus \mathbb{K}$ *irreducible. Let* U *be the set of points* $(\alpha_1,\ldots,\alpha_n, \beta_1,\ldots,\beta_n,\gamma_1,\ldots,\gamma_n) \in \mathbb{K}^{3n}$ *for which*

$$F(\alpha_1 t + \beta_1 u + \gamma_1, \ldots, \alpha_n t + \beta_n u + \gamma_n)$$

is irreducible in $\mathbb{K}[t,u]$. *Then* U *is a Zariski open subset of* $\mathbb{K}^{3n}$, *which is dense over the algebraic closure of* $\mathbb{K}$.

$\longrightarrow$ modulo random shifts, bivariate instead of univariate projections suffice

$\longrightarrow$ in practice a shift is often not necessary

Bivariate factorization

Notation

$2 \leqslant \omega \leqslant 3$: two $n \times n$ matrices can be multiplied in time n^ω

$\mathsf{F}_{\mathbb{K}}(d)$: cost to completely factor a polynomial of degree d in $\mathbb{K}[x]$

Theorem (Lecerf 2010)

Let $F \in \mathbb{K}[x,y]$ of bidegree (d_x, d_y) be square-free and content-free in both x and y. Assume that $\operatorname{char} \mathbb{K} = 0$ *or* $\operatorname{char} \mathbb{K} > d_y(2d_x - 1)$. *Then, with high probability, we can compute the irreducible factorization of F in time*

$$\tilde{O}(d_x^2 d_y + d_x^\omega) + \mathsf{F}_{\mathbb{K}}(d_x).$$

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\,B(x,c), \quad A,B$ coprime $\qquad (*)$

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\,B(x,c), \quad A,B$ coprime $\quad (*)$

$\longrightarrow$ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\,B(x,c), \quad A,B$ coprime $\qquad (*)$

$\longrightarrow$ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

$\longrightarrow$ factorization in $\mathbb{K}[x,y]$ if $(*)$ came from one (in quasi-linear time)

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\,B(x,c), \quad A,B$ coprime $\qquad (*)$

$\longrightarrow$ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

$\longrightarrow$ factorization in $\mathbb{K}[x,y]$ if $(*)$ came from one (in quasi-linear time)

Multivariate case

Assume $\qquad F(x_1,\ldots,x_n) = A(x_1,\ldots,x_n)\,B(x_1,\ldots,x_n)$, but A,B unknown

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y]$, $F(x,c) = A(x,c)\,B(x,c)$, A,B coprime (∗)

⟶ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

⟶ factorization in $\mathbb{K}[x,y]$ if (∗) came from one (in quasi-linear time)

Multivariate case

Assume $F(x_1,\ldots,x_n) = A(x_1,\ldots,x_n)\,B(x_1,\ldots,x_n)$, but A,B unknown

$F(x_1,\ldots,x_{n-1},c) = A_c(x_1,\ldots,x_{n-1})\,B_c(x_1,\ldots,x_{n-1})$ coprime fact.

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\, B(x,c), \quad A, B$ coprime $\qquad (*)$

$\longrightarrow$ unique $A, B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\, B(x,y)$

$\longrightarrow$ factorization in $\mathbb{K}[x,y]$ if $(*)$ came from one (in quasi-linear time)

Multivariate case

Assume $\quad F(x_1,\ldots,x_n) = A(x_1,\ldots,x_n)\, B(x_1,\ldots,x_n)$, but A, B unknown

$F(x_1,\ldots,x_{n-1},c) = A_c(x_1,\ldots,x_{n-1})\, B_c(x_1,\ldots,x_{n-1})$ coprime fact.

Consider $\quad F_\alpha(t,u) := F(\alpha_1 t, \ldots, \alpha_{n-1} t, u)$

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y], \quad F(x,c) = A(x,c)\,B(x,c), \quad A,B$ coprime $\qquad (*)$

$\longrightarrow$ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

$\longrightarrow$ factorization in $\mathbb{K}[x,y]$ if $(*)$ came from one (in quasi-linear time)

Multivariate case

Assume $F(x_1,\ldots,x_n) = A(x_1,\ldots,x_n)\,B(x_1,\ldots,x_n)$, but A,B unknown

$F(x_1,\ldots,x_{n-1},c) = A_c(x_1,\ldots,x_{n-1})\,B_c(x_1,\ldots,x_{n-1})$ coprime fact.

Consider $F_\alpha(t,u) := F(\alpha_1 t,\ldots,\alpha_{n-1} t,u)$

$A_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$ and $B_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$ are coprime with HP in $\boldsymbol{\alpha}$

Iterative method via Hensel lifting

Bivariate Hensel lifting

$F \in \mathbb{K}[x,y]$, $F(x,c) = A(x,c)\,B(x,c)$, A,B coprime (*)

$\longrightarrow$ unique $A,B \in \mathbb{K}[[y-c]][x]$ with $F(x,y) = A(x,y)\,B(x,y)$

$\longrightarrow$ factorization in $\mathbb{K}[x,y]$ if (*) came from one (in quasi-linear time)

Multivariate case

Assume $F(x_1,\ldots,x_n) = A(x_1,\ldots,x_n)\,B(x_1,\ldots,x_n)$, but A,B unknown

$F(x_1,\ldots,x_{n-1},c) = A_c(x_1,\ldots,x_{n-1})\,B_c(x_1,\ldots,x_{n-1})$ coprime fact.

Consider $F_\alpha(t,u) := F(\alpha_1 t,\ldots,\alpha_{n-1} t,u)$

$A_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$ and $B_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$ are coprime with HP in α

α →  → $F_\alpha(t,u) = A_\alpha(t,u)\,B_\alpha(t,u)$,

$$A_\alpha(t,c) = A_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$$
$$B_\alpha(t,c) = B_c(\alpha_1 t,\ldots,\alpha_{n-1} t)$$

Complexity of Hensel lifting

Theorem

Let $s := \min(s_A, s_B)$, $s' := \max(s_A, s_B)$ $\bar{s} := \max(s', s_F)$, $d := \deg F$, *and* $\delta := \max(\deg_{x_1} F, \ldots, \deg_{x_n} F)$. *Then Hensel lifting can be done with HP in time*

$$O(n((\bar{s}/s)\,\mathsf{S}(s') + \delta d\,\mathsf{S}(s) + s\,\mathsf{M}(\delta d) + s\,\mathsf{M}(d)\log d))$$

Complexity of Hensel lifting

Theorem

Let $s := \min(s_A, s_B)$, $s' := \max(s_A, s_B)$ $\bar{s} := \max(s', s_F)$, $d := \deg F$, *and* $\delta := \max(\deg_{x_1} F, \ldots, \deg_{x_n} F)$. *Then Hensel lifting can be done with HP in time*

$$O(n((\bar{s}/s)\,\mathsf{S}(s') + \delta d\,\mathsf{S}(s) + s\,\mathsf{M}(\delta d) + s\,\mathsf{M}(d)\log d))$$

In favorable cases, this leads to a HP algorithm to completely factor

$$F = A_1 \cdots A_\ell$$

in time

$$O(n((\bar{s}/s + \delta d)\,\mathsf{S}(s) + s\,\mathsf{M}(\delta d)\log \ell + s\,\mathsf{M}(d)\log d)) + \tilde{O}(\delta^3) + \mathsf{F}_{\mathbb{K}}(\delta)$$

General algorithm

$\alpha_1,\ldots,\alpha_n,\quad \beta_1,\ldots,\beta_n,\quad \gamma_1,\ldots,\gamma_n$ random

$$\begin{aligned}\hat{F}(x_1,\ldots,x_n,t,u,\lambda) \;:=\; F(&(1-\lambda+\alpha_1\lambda)(t+\beta_1 u+\gamma_1)\,x_1,\ldots,\\ &(1-\lambda+\alpha_n\lambda)(t+\beta_n u+\gamma_n)\,x_n).\end{aligned}$$

General algorithm

$\alpha_1,\ldots,\alpha_n, \quad \beta_1,\ldots,\beta_n, \quad \gamma_1,\ldots,\gamma_n$ random

$$\hat{F}(x_1,\ldots,x_n,t,u,\lambda) := F((1-\lambda+\alpha_1\lambda)(t+\beta_1 u+\gamma_1)x_1,\ldots,\\ (1-\lambda+\alpha_n\lambda)(t+\beta_n u+\gamma_n)x_n).$$

We use $\boldsymbol{\alpha}^k = (\alpha_1^k,\ldots,\alpha_n^k)$ as our interpolation points

$$F^{\langle i\rangle}(t,u,\lambda) := \hat{F}(\alpha_1^i,\ldots,\alpha_n^i,t,u,\lambda)$$

General algorithm

$\alpha_1,\ldots,\alpha_n,\quad \beta_1,\ldots,\beta_n,\quad \gamma_1,\ldots,\gamma_n$ random

$$\hat{F}(x_1,\ldots,x_n,t,u,\lambda) \ := \ F((1-\lambda+\alpha_1\lambda)(t+\beta_1 u+\gamma_1)x_1,\ldots,\\ (1-\lambda+\alpha_n\lambda)(t+\beta_n u+\gamma_n)x_n).$$

We use $\boldsymbol{\alpha}^k = (\alpha_1^k,\ldots,\alpha_n^k)$ as our interpolation points

$$F^{\langle i\rangle}(t,u,\lambda) \ := \ \hat{F}(\alpha_1^i,\ldots,\alpha_n^i,t,u,\lambda)$$

By construction

$$F^{\langle i+1\rangle}(t,u,0) \ = \ F^{\langle i\rangle}(t,u,1)$$

General algorithm — continued

By Hilbert-Bertini, with high probability,

we have the following “propagition” of irreducible factorizations:

$$\begin{array}{ccccccl}
F^{\langle i\rangle}(t,u,0) & = & A_1^{\langle i\rangle}(t,u,0) & \cdots & A_\ell^{\langle i\rangle}(t,u,0) & & \\
\downarrow & & \downarrow & & \downarrow & & \text{Hensel} \\
F^{\langle i\rangle}(t,u,\lambda) & = & A_1^{\langle i\rangle}(t,u,\lambda) & \cdots & A_\ell^{\langle i\rangle}(t,u,\lambda) & & \\
\downarrow & & \downarrow & & \downarrow & & \text{evaluate} \\
F^{\langle i\rangle}(t,u,1) & = & A_1^{\langle i\rangle}(t,u,1) & \cdots & A_\ell^{\langle i\rangle}(t,u,1) & & \\
\parallel & & \parallel & & \parallel & & \text{identify} \\
F^{\langle i+1\rangle}(t,u,0) & = & A_1^{\langle i+1\rangle}(t,u,0) & \cdots & A_\ell^{\langle i+1\rangle}(t,u,0) & & \\
\downarrow & & \downarrow & & \downarrow & & \\
\vdots & & \vdots & & \vdots & &
\end{array}$$

General algorithm — continued

By Hilbert-Bertini, with high probability,

we have the following “propagition” of irreducible factorizations:

$$
\begin{array}{ccccccl}
F^{\langle i\rangle}(t,u,0) & = & A_1^{\langle i\rangle}(t,u,0) & \cdots & A_\ell^{\langle i\rangle}(t,u,0) & & \\
\downarrow & & \downarrow & & \downarrow & & \text{Hensel} \\
F^{\langle i\rangle}(t,u,\lambda) & = & A_1^{\langle i\rangle}(t,u,\lambda) & \cdots & A_\ell^{\langle i\rangle}(t,u,\lambda) & & \\
\downarrow & & \downarrow & & \downarrow & & \text{evaluate} \\
F^{\langle i\rangle}(t,u,1) & = & A_1^{\langle i\rangle}(t,u,1) & \cdots & A_\ell^{\langle i\rangle}(t,u,1) & & \\
\| & & \| & & \| & & \text{identify} \\
F^{\langle i+1\rangle}(t,u,0) & = & A_1^{\langle i+1\rangle}(t,u,0) & \cdots & A_\ell^{\langle i+1\rangle}(t,u,0) & & \\
\downarrow & & \downarrow & & \downarrow & & \\
\vdots & & \vdots & & \vdots & &
\end{array}
$$

After one further tweak for normalization $\longrightarrow$ SLP for computing $A_1^{\langle i\rangle},\ldots,A_\ell^{\langle i\rangle}$

General algorithm — complexity

Theorem

Let $s := \max(s_{P_1}, \ldots, s_{P_\ell})$, $\bar{s} := \max(s, s_F)$, $d := \deg F$, *and* $e := \mathrm{ec}_w F \leqslant d^2$ *(for a suitable regularizing weight* w*). Assume* $\mathrm{char}\, \mathbb{K} = 0$ *or* $\mathrm{char}\, \mathbb{K} > 2d^2$*. Then, with high probability, we can compute the irreducible factorization of* F *in time*

$$O(\mathsf{S}(d^3 \bar{s}) + \mathsf{M}(d^3)\, s \log d) + \tilde{O}(e^5) + \mathsf{F}_{\mathbb{K}}(e + 3d)$$

Thank you !

http://www.TeXmacs.org