# *Computing low-degree factors of lacunary polynomials: a Newton-Puiseux Approach*

**Bruno Grenet**
LIX — École Polytechnique

Groupe de travail MC2
Lyon, 18 juin 2014

**Factorization of a polynomial $f$**

Find $f_1, \dots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

**Factorization of a polynomial $f$**

Find $f_1, \ldots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

- Many algorithms
  - over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{Q}_p$, $\mathbb{F}_q$, $\mathbb{R}$, $\mathbb{C}$, $\ldots$;
  - in 1, 2, $\ldots$, $n$ variables.

**Factorization of a polynomial $f$**

Find $f_1, \ldots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

- Many algorithms
  - over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{Q}_p$, $\mathbb{F}_q$, $\mathbb{R}$, $\mathbb{C}$, $\ldots$;
  - in 1, 2, $\ldots$, $n$ variables.
- Complexity: **polynomial in** $\deg(f)$

**Factorization of a polynomial $f$**

Find $f_1, \ldots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

- Many algorithms
  - over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{Q}_p$, $\mathbb{F}_q$, $\mathbb{R}$, $\mathbb{C}$, $\ldots$;
  - in 1, 2, $\ldots$, $n$ variables.
- Complexity: **polynomial in** $\deg(f)$

$$X^{102}Y^{101} + X^{101}Y^{102} - X^{101}Y^{101} - X - Y + 1$$

**Factorization of a polynomial $f$**

Find $f_1, \dots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

- Many algorithms
  - over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{Q}_p$, $\mathbb{F}_q$, $\mathbb{R}$, $\mathbb{C}$, $\dots$;
  - in 1, 2, $\dots$, $n$ variables.
- Complexity: **polynomial in** $\deg(f)$

$$X^{102}Y^{101} + X^{101}Y^{102} - X^{101}Y^{101} - X - Y + 1$$
$$= (X + Y - 1) \times (X^{101}Y^{101} - 1)$$

**Factorization of a polynomial $f$**

Find $f_1, \dots, f_t$, irreducible, s.t. $f = f_1 \times \cdots \times f_t$.

- Many algorithms
  - over $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{Q}_p$, $\mathbb{F}_q$, $\mathbb{R}$, $\mathbb{C}$, $\dots$;
  - in 1, 2, $\dots$, $n$ variables.
- Complexity: **polynomial in $\deg(f)$**

$$\begin{aligned} X^{102}Y^{101} + X^{101}Y^{102} - X^{101}Y^{101} - X - Y + 1 \\ = (X+Y-1) \times (X^{101}Y^{101} - 1) \\ = (X+Y-1) \times (XY-1) \times (1 + XY + \cdots + X^{100}Y^{100}) \end{aligned}$$

**Definition**

$$f(X_1, \dots, X_n) = \sum_{j=1}^{k} c_j X_1^{\alpha_{1j}} \cdots X_n^{\alpha_{nj}}$$

- $\mathrm{size}(f) \simeq k\Big(\max_j(\mathrm{size}(c_j)) + n\log(\deg f)\Big)$

**Definition**

$$f(X_1,\dots,X_n)=\sum_{j=1}^{k} c_j X_1^{\alpha_{1j}}\cdots X_n^{\alpha_{nj}}$$

- $\text{size}(f)\simeq k\Big(\max_j(\text{size}(c_j))+n\log(\deg f)\Big)$

**Compute the degree-$d$ factors of $f$ in time $\text{poly}(\text{size}(f),d)$**

**Definition**

$$f(X_1,\dots,X_n)=\sum_{j=1}^{k} c_j X_1^{\alpha_{1j}}\cdots X_n^{\alpha_{nj}}$$

- $\mathrm{size}(f)\simeq k\Big(\max_j(\mathrm{size}(c_j))+n\log(\deg f)\Big)$

**Compute the degree-$d$ factors of $f$ in time $\mathrm{poly}(\mathrm{size}(f),d)$**

Let $f\in\mathbb{R}[X]$ with $k$ nonzero terms. Then $\#Z_{\mathbb{R}}(f)\leqslant 2k-1$.

**Theorems**

There exist deterministic polynomial-time algorithms computing

- **linear** factors (integer roots) of $f \in \mathbb{Z}[X]$; **[Cucker-Koiran-Smale'98]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X]$; **[H. Lenstra'99]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X_1, \dots, X_n]$. **[Kaltofen-Koiran'06]**

**Theorems**

There exist deterministic polynomial-time algorithms computing

- **linear** factors (integer roots) of $f \in \mathbb{Z}[X]$; **[Cucker-Koiran-Smale'98]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X]$; **[H. Lenstra'99]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X_1, \dots, X_n]$. **[Kaltofen-Koiran'06]**

It is NP-**hard** to compute **roots of** $f \in \mathbb{F}_p[X]$. **[Bi-Cheng-Rojas'13]**

# *Factorization of lacunary polynomials*

**Theorems**

There exist deterministic polynomial-time algorithms computing

- **linear** factors (integer roots) of $f \in \mathbb{Z}[X]$; **[Cucker-Koiran-Smale'98]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X]$; **[H. Lenstra'99]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X_1, \dots, X_n]$. **[Kaltofen-Koiran'06]**

It is **NP-hard** to compute **roots of** $f \in \mathbb{F}_p[X]$. **[Bi-Cheng-Rojas'13]**

- Only available for number fields
- Based on number-theoretic results $\rightsquigarrow$ theoretical algorithms

**Theorems**

There exist deterministic polynomial-time algorithms computing

- **linear** factors (integer roots) of $f \in \mathbb{Z}[X]$; **[Cucker-Koiran-Smale'98]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X]$; **[H. Lenstra'99]**
- **low-degree** factors of $f \in \mathbb{Q}(\alpha)[X_1, \dots, X_n]$. **[Kaltofen-Koiran'06]**

It is NP-**hard** to compute **roots of** $f \in \mathbb{F}_p[X]$. **[Bi-Cheng-Rojas'13]**

- Only available for number fields
- Based on number-theoretic results $\rightsquigarrow$ theoretical algorithms

Generalization to other fields? More practical algorithms?

Let $\mathbb{K}$ be any field of characteristic $0$.

**Theorem (G.'14)**

The computation of the degree-$d$ factors of $f \in \mathbb{K}[X_1, \dots, X_n]$ reduces to

- univariate lacunary factorizations plus post-processing, and
- multivariate low-degree factorizations,

in $\text{poly}(\text{size}(f), d)$ bit operations.

Let $\mathbb{K}$ be any field of characteristic $0$.

**Theorem (G.'14)**

The computation of the degree-$d$ factors of $f \in \mathbb{K}[X_1, \dots, X_n]$ reduces to

- univariate lacunary factorizations plus post-processing, and
- multivariate low-degree factorizations,

in $\text{poly}(\text{size}(f), d)$ bit operations.

- Case $d = 1$ [G.-Chattopadhyay-Koiran-Portier-Strozecki'13]

Let $\mathbb{K}$ be any field of characteristic $0$.

**Theorem (G.'14)**

The computation of the degree-$d$ factors of $f \in \mathbb{K}[X_1, \dots, X_n]$ reduces to

- univariate lacunary factorizations plus post-processing, and
- multivariate low-degree factorizations,

in $\mathrm{poly}(\mathrm{size}(f), d)$ bit operations.

- Case $d = 1$ [G.-Chattopadhyay-Koiran-Portier-Strozecki'13]
- New algorithm for $\mathbb{K} = \mathbb{Q}(\alpha)$; **some** factors for $\mathbb{K} = \overline{\mathbb{Q}}, \mathbb{R}, \mathbb{C}, \mathbb{Q}_p$

# *Linear factors of bivariate polynomials*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

**Observation**

$(Y - uX - v)$ divides $f(X, Y) \iff f(X, uX + v) \equiv 0$

# *Linear factors of bivariate polynomials*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

**Observation**

$(Y - uX - v)$ divides $f(X,Y) \iff f(X, uX+v) \equiv 0$

**Theorem**

$$\operatorname{val}\left(\sum_{j=1}^{\ell} c_j X^{\alpha_j}(uX+v)^{\beta_j}\right) \leqslant \alpha_1 + \binom{\ell}{2} \text{ if } f \neq 0 \text{ and } uv \neq 0.$$

# *Linear factors of bivariate polynomials*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

**Observation**

$(Y - uX - v)$ divides $f(X, Y) \iff f(X, uX + v) \equiv 0$

**Theorem**

$$\operatorname{val}\left(\sum_{j=1}^{\ell} c_j X^{\alpha_j}(uX+v)^{\beta_j}\right) \leqslant \alpha_1 + \binom{\ell}{2} \text{ if } f \neq 0 \text{ and } uv \neq 0.$$

**Gap Theorem**

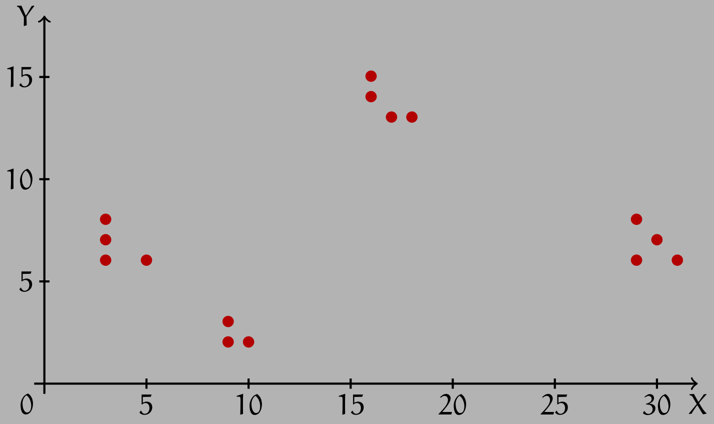
Let $f = f_1 + f_2 \in \mathbb{K}[X, Y]$. If $\operatorname{val}_X(f_2) > \operatorname{val}_X(f_1) + \binom{\#f_1}{2}$, then for all $uv \neq 0$, $(Y - uX - v)$ divides $f$ iff it divides both $f_1$ and $f_2$.

# *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

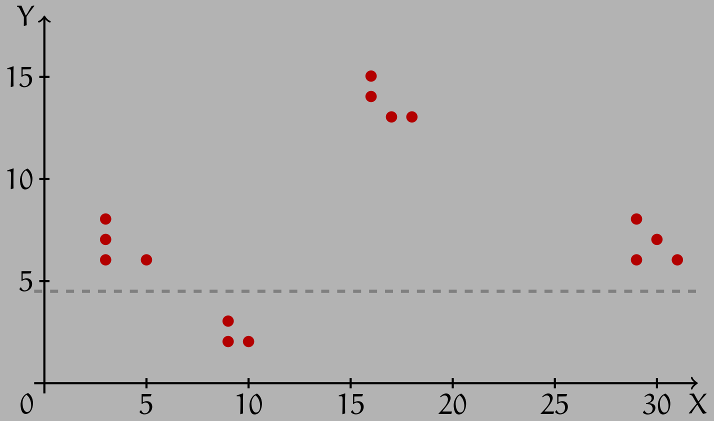
# Example

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$



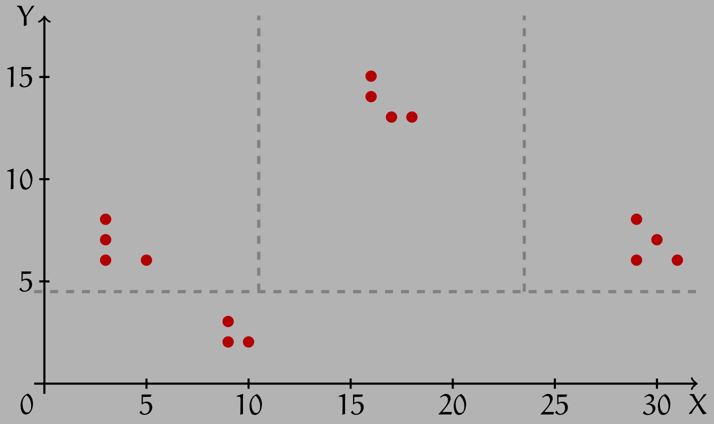
# *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

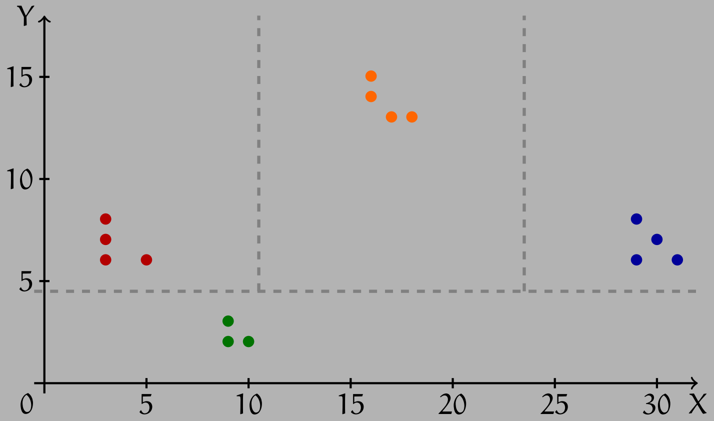


# *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$



$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$



## *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

$$f_1 = X^3Y^6(-X^2 + Y^2 - 2Y + 1)$$

$$f = X^{31}Y^{6} - 2X^{30}Y^{7} + X^{29}Y^{8} - X^{29}Y^{6} + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^{2} - X^{9}Y^{3} + X^{9}Y^{2} - X^{5}Y^{6} + X^{3}Y^{8} - 2X^{3}Y^{7} + X^{3}Y^{6}$$

$$f_1 = X^3Y^6(X - Y + 1)(1 - X - Y)$$

# *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

$$f_1 = X^3Y^6(X - Y + 1)(1 - X - Y)$$

$$f_2 = X^9Y^2(X - Y + 1)$$

$$f_3 = X^{16}Y^{13}(X + Y)(X - Y + 1)$$

$$f_4 = X^{29}Y^6(X + Y - 1)(X - Y + 1)$$

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

$$f_1 = X^3Y^6(X - Y + 1)(1 - X - Y)$$

$$f_2 = X^9Y^2(X - Y + 1)$$

$$f_3 = X^{16}Y^{13}(X + Y)(X - Y + 1)$$

$$f_4 = X^{29}Y^6(X + Y - 1)(X - Y + 1)$$

$\implies$ linear factors of f: $(X - Y + 1, 1)$

# *Example*

$$f = X^{31}Y^6 - 2X^{30}Y^7 + X^{29}Y^8 - X^{29}Y^6 + X^{18}Y^{13} - X^{16}Y^{15} + X^{17}Y^{13} + X^{16}Y^{14} + X^{10}Y^2 - X^9Y^3 + X^9Y^2 - X^5Y^6 + X^3Y^8 - 2X^3Y^7 + X^3Y^6$$

$$f_1 = X^3Y^6(X - Y + 1)(1 - X - Y)$$
$$f_2 = X^9Y^2(X - Y + 1)$$
$$f_3 = X^{16}Y^{13}(X + Y)(X - Y + 1)$$
$$f_4 = X^{29}Y^6(X + Y - 1)(X - Y + 1)$$

$\Longrightarrow$ linear factors of f: $(X - Y + 1, 1)$, $(X, 3)$, $(Y, 2)$

# *Algorithm for linear factors*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

Find linear factors of $f(X,Y) = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

Find linear factors of $f(X,Y) = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

monomials

$(X, \min_j \alpha_j)$
$(Y, \min_j \beta_j)$

# *Algorithm for linear factors*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

Find linear factors of $f(X,Y) = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

monomials

$(X, \min_j \alpha_j)$
$(Y, \min_j \beta_j)$

binomials

$(X - a)$
Factors of $\sum_j c_j X^{\alpha_j}$

$(Y - uX)$
Roots of $u \mapsto \sum_j c_j u^{\beta_j}$

Univariate lacunary factorization

[H. Lenstra'99]

# *Algorithm for linear factors*

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

Find linear factors of $f(X,Y) = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

- **monomials** → $(X, \min_j \alpha_j)$, $(Y, \min_j \beta_j)$
- **binomials** →
  - $(X - a)$: Factors of $\sum_j c_j X^{\alpha_j}$
  - $(Y - uX)$: Roots of $u \mapsto \sum_j c_j u^{\beta_j}$

  Univariate lacunary factorization [H. Lenstra'99]
- **trinomials** → Common factors of $f_t = \sum_{j=j_t}^{j_t+\ell_t-1} c_j X^{\alpha_j} Y^{\beta_j}$ $(\deg(f_t) \leqslant \mathcal{O}(\ell_t^2))$

  Low-degree factorization [Kaltofen'82, …, Lecerf'07]

*[Chattopadhyay-G.-Koiran-Portier-Strozecki'13]*

Let $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j} \in \mathbb{Q}(\alpha)[X, Y]$ be given in lacunary representation. There exists a **deterministic polynomial-time** algorithm to compute its linear factors, with multiplicities.

monomials → $(X, \min_j \alpha_j)$, $(Y, \min_j \beta_j)$

binomials → $(X - a)$: Factors of $\sum_j c_j X^{\alpha_j}$; $(Y - uX)$: Roots of $u \mapsto \sum_j c_j u^{\beta_j}$

Univariate lacunary factorization
[H. Lenstra'99]

trinomials → Common factors of $f_t = \sum_{j=j_t}^{j_t+\ell_t-1} c_j X^{\alpha_j} Y^{\beta_j}$ $(\deg(f_t) \leqslant \mathcal{O}(\ell_t^2))$

Low-degree factorization
[Kaltofen'82, ..., Lecerf'07]

**Observation for low-degree factors**

$g(X,Y)$ divides $f(X,Y) \iff f(X,\phi(X)) \equiv 0$

**Observation for low-degree factors**

$g(X,Y)$ divides $f(X,Y) \iff f(X,\phi(X)) \equiv 0$

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X))$$

**Observation for low-degree factors**

$g(X, Y)$ divides $f(X, Y) \iff f(X, \phi(X)) \equiv 0$

$$g(X, Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X))$$

- $g_0 \in \mathbb{K}[X]$
- $\phi_1, \ldots, \phi_d \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ are **Puiseux series**:

$$\phi(X) = \sum_{t \geqslant t_0} a_t X^{t/n} \text{ with } a_t \in \overline{\mathbb{K}},\ a_{t_0} \neq 0.$$

**Observation for low-degree factors**

$g(X,Y)$ divides $f(X,Y) \iff f(X,\phi(X)) \equiv 0$

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X))$$

- $g_0 \in \mathbb{K}[X]$
- $\phi_1, \dots, \phi_d \in \overline{\mathbb{K}}\langle\langle X\rangle\rangle$ are **Puiseux series**:

$$\phi(X) = \sum_{t \geqslant t_0} a_t X^{t/n} \text{ with } a_t \in \overline{\mathbb{K}},\ a_{t_0} \neq 0.$$

- If $g$ is irreducible, $g$ divides $f \iff \exists i,\ f(X,\phi_i) = 0$
  $\iff \forall i,\ f(X,\phi_i) = 0$

**Observation for low-degree factors**

$g(X,Y)$ divides $f(X,Y) \iff f(X,\phi(X)) \equiv 0$

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X))$$

- $g_0 \in \mathbb{K}[X]$
- $\phi_1, \ldots, \phi_d \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ are **Puiseux series**:

$$\phi(X) = \sum_{t \geqslant t_0} a_t X^{t/n} \text{ with } a_t \in \overline{\mathbb{K}},\ a_{t_0} \neq 0.$$

- If $g$ is irreducible, $g$ divides $f \iff \exists i,\ f(X,\phi_i) = 0$
  $\iff \forall i,\ f(X,\phi_i) = 0$
- **Valuation**: $\text{val}(\phi) = t_0/n$.

**Theorem**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $g$ a degree-$d$ irreducible polynomial with a root $\phi \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ of valuation $\nu$.
If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

$$\operatorname{val}(f_1(X, \phi)) \leqslant \min_j(\alpha_j + \nu\beta_j) + (2d(4d+1) - \nu)\binom{\ell}{2}.$$

**Theorem**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $g$ a degree-$d$ irreducible polynomial with a root $\phi \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ of valuation $\nu$.
If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

$$\mathrm{val}(f_1(X,\phi)) \leqslant \min_j(\alpha_j + \nu\beta_j) + (2d(4d+1) - \nu)\binom{\ell}{2}.$$

**Proof idea.** Let $\psi_j = X^{\alpha_j}\phi^{\beta_j}$ for all $j$.

**Theorem**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $g$ a degree-$d$ irreducible polynomial with a root $\phi \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ of valuation $\nu$.
If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

$$\mathrm{val}(f_1(X,\phi)) \leqslant \min_j(\alpha_j + \nu\beta_j) + (2d(4d+1) - \nu)\binom{\ell}{2}.$$

**Proof idea.** Let $\psi_j = X^{\alpha_j}\phi^{\beta_j}$ for all $j$.

- Wronskian: $\mathrm{wr}(\psi_1,\dots,\psi_\ell) = \det\left(\psi_j^{(i)}\right) = \frac{1}{c_1}\mathrm{wr}(f_1,\psi_2,\dots,\psi_\ell)$

**Theorem**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $g$ a degree-$d$ irreducible polynomial with a root $\phi \in \overline{\mathbb{K}}\langle\langle X \rangle\rangle$ of valuation $\nu$.
If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

$$\mathrm{val}(f_1(X,\phi)) \leqslant \min_j(\alpha_j + \nu\beta_j) + (2d(4d+1) - \nu)\binom{\ell}{2}.$$

**Proof idea.** Let $\psi_j = X^{\alpha_j}\phi^{\beta_j}$ for all $j$.

- Wronskian: $\mathrm{wr}(\psi_1, \ldots, \psi_\ell) = \det\left(\psi_j^{(i)}\right) = \frac{1}{c_1}\mathrm{wr}(f_1, \psi_2, \ldots, \psi_\ell)$
- $\mathrm{val}(\mathrm{wr}(f_1, \psi_2, \ldots, \psi_\ell)) \geqslant \mathrm{val}(f_1) + \sum_{j>1} \mathrm{val}(\psi_j)$

**Theorem**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $g$ a degree-$d$ irreducible polynomial with a root $\phi \in \overline{\mathbb{K}}\langle\langle X\rangle\rangle$ of valuation $\nu$.
If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

$$\operatorname{val}(f_1(X,\phi)) \leqslant \min_j(\alpha_j + \nu\beta_j) + (2d(4d+1)-\nu)\binom{\ell}{2}.$$

**Proof idea.** Let $\psi_j = X^{\alpha_j}\phi^{\beta_j}$ for all $j$.

- Wronskian: $\operatorname{wr}(\psi_1,\dots,\psi_\ell) = \det\left(\psi_j^{(i)}\right) = \frac{1}{c_1}\operatorname{wr}(f_1,\psi_2,\dots,\psi_\ell)$
- $\operatorname{val}(\operatorname{wr}(f_1,\psi_2,\dots,\psi_\ell)) \geqslant \operatorname{val}(f_1) + \sum_{j>1}\operatorname{val}(\psi_j)$
- $\operatorname{val}(\operatorname{wr}(\psi_1,\dots,\psi_\ell) \leqslant \sum_j \operatorname{val}(\psi_j) + (2d(4d+1)-\nu)\binom{\ell}{2}$

**Gap Theorem** [G.'14]

Let

$$f = \underbrace{\sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}}_{f_1} + \underbrace{\sum_{j=\ell+1}^{k} c_j X^{\alpha_j} Y^{\beta_j}}_{f_2}$$

with $uv \neq 0$, $\alpha_1 + v\beta_1 \leqslant \cdots \leqslant \alpha_k + v\beta_k$. Let g a degree-d irreducible poynomial, with a root of valuation $v$.
If $\ell$ is the smallest index s.t.

$$\alpha_{\ell+1} + v\beta_{\ell+1} > (\alpha_1 + v\beta_1) + (2d(4d+1) - v)\binom{\ell}{2},$$

then g divides f iff it divides both $f_1$ and $f_2$.

**Gap Theorem** [G.'14]

Let

$$f = \underbrace{\sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}}_{f_1} + \underbrace{\sum_{j=\ell+1}^{k} c_j X^{\alpha_j} Y^{\beta_j}}_{f_2}$$

with $uv \neq 0$, $\alpha_1 + v\beta_1 \leqslant \cdots \leqslant \alpha_k + v\beta_k$. Let g a degree-d irreducible poynomial, with a root of valuation $v$.
If $\ell$ is the smallest index s.t.

$$\alpha_{\ell+1} + v\beta_{\ell+1} > (\alpha_1 + v\beta_1) + (2d(4d+1) - v)\binom{\ell}{2},$$

then g divides f iff it divides both $f_1$ and $f_2$.

**Gap Theorem** [G.'14]

Let

$$f = \underbrace{\sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}}_{f_1} + \underbrace{\sum_{j=\ell+1}^{k} c_j X^{\alpha_j} Y^{\beta_j}}_{f_2}$$

with $uv \neq 0$, $\alpha_1 + v\beta_1 \leqslant \cdots \leqslant \alpha_k + v\beta_k$. Let $g$ a degree-$d$ irreducible poynomial, with a root of valuation $v$.
If $\ell$ is the smallest index s.t.

$$\alpha_{\ell+1} + v\beta_{\ell+1} > (\alpha_1 + v\beta_1) + (2d(4d+1) - v)\binom{\ell}{2},$$

then $g$ divides $f$ iff it divides both $f_1$ and $f_2$.

- Depends on $v$.
- Does not bound $\alpha_j$ nor $\beta_j$

**Proposition**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $v_1 \neq v_2$ such that for all j

$$\begin{cases} \alpha_j + v_1\beta_j \leqslant \alpha_1 + v_1\beta_1 + (2d(4d+1) - v_1)\binom{\ell}{2} \\ \alpha_j + v_2\beta_j \leqslant \alpha_2 + v_2\beta_2 + (2d(4d+1) - v_2)\binom{\ell}{2}. \end{cases}$$

Then for all $p, q$, $|\alpha_p - \alpha_q| \leqslant \mathcal{O}(\ell^2 d^4)$ and $|\beta_p - \beta_q| \leqslant \mathcal{O}(\ell^2 d^4)$.

**Proposition**

Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $\nu_1 \neq \nu_2$ such that for all $j$

$$\begin{cases} \alpha_j + \nu_1 \beta_j \leqslant \alpha_1 + \nu_1 \beta_1 + (2d(4d+1) - \nu_1)\binom{\ell}{2} \\ \alpha_j + \nu_2 \beta_j \leqslant \alpha_2 + \nu_2 \beta_2 + (2d(4d+1) - \nu_2)\binom{\ell}{2}. \end{cases}$$

Then for all $p, q$, $|\alpha_p - \alpha_q| \leqslant \mathcal{O}(\ell^2 d^4)$ and $|\beta_p - \beta_q| \leqslant \mathcal{O}(\ell^2 d^4)$.

Degree-$d$ factors of $f$ having two roots of valuation $\nu_1$ and $\nu_2$:

- Write $f = f_1 + \cdots + f_s$, using $\nu_1$ and then $\nu_2$;
- Write $f_t = X^a Y^b f_t^\circ$ with $\deg(f_t^\circ) \leqslant \mathcal{O}(\ell^2 d^4)$;
- Factor $\gcd(f_1^\circ, \dots, f_s^\circ)$.

**Proposition**

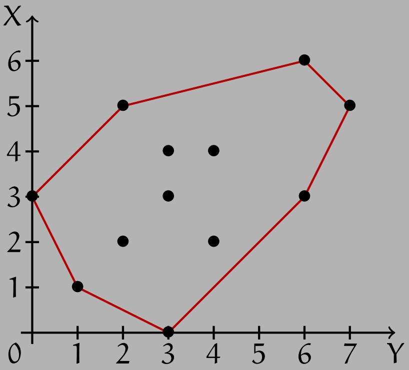
Let $f_1 = \sum_{j=1}^{\ell} c_j X^{\alpha_j} Y^{\beta_j}$ and $v_1 \neq v_2$ such that for all $j$

$$\begin{cases} \alpha_j + v_1\beta_j \leqslant \alpha_1 + v_1\beta_1 + (2d(4d+1) - v_1)\binom{\ell}{2} \\ \alpha_j + v_2\beta_j \leqslant \alpha_2 + v_2\beta_2 + (2d(4d+1) - v_2)\binom{\ell}{2}. \end{cases}$$

Then for all $p, q$, $|\alpha_p - \alpha_q| \leqslant \mathcal{O}(\ell^2 d^4)$ and $|\beta_p - \beta_q| \leqslant \mathcal{O}(\ell^2 d^4)$.

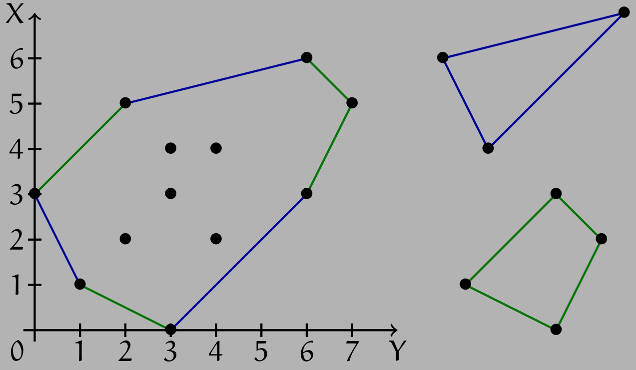
Degree-$d$ factors of $f$ having two roots of valuation $v_1$ and $v_2$:

- Write $f = f_1 + \cdots + f_s$, using $v_1$ and then $v_2$;
- Write $f_t = X^a Y^b f_t^\circ$ with $\deg(f_t^\circ) \leqslant \mathcal{O}(\ell^2 d^4)$;
- Factor $\gcd(f_1^\circ, \ldots, f_s^\circ)$. $\rightsquigarrow$ **low-degree bivariate factorization**

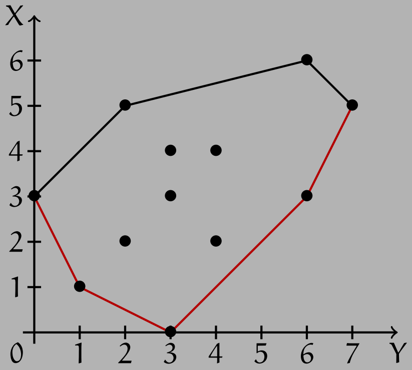



$$f = Y^3 + 2XY - X^2Y^4 + X^3Y^3 - 2X^2Y^2 - 4X^3 + 2X^4Y^3 - 2X^5Y^2 + X^3Y^6 + 2X^4Y^4 - X^5Y^7 + X^6Y^6$$

# *Newton polygon*

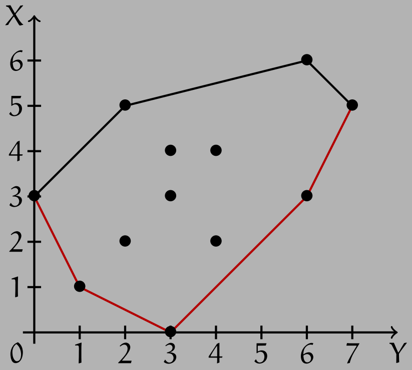



$$f = Y^3 + 2XY - X^2Y^4 + X^3Y^3 - 2X^2Y^2 - 4X^3 + 2X^4Y^3 - 2X^5Y^2$$
$$+ X^3Y^6 + 2X^4Y^4 - X^5Y^7 + X^6Y^6$$
$$= (Y - 2X^2 + X^3Y^4)(Y^2 + 2X - X^2Y^3 + X^3Y^2)$$




Newton-Puiseux Theorem

For each edge in the **lower hull** of slope $-\nu$, $f$ has a root $\phi \in \overline{\mathbb{K}}\langle\langle X\rangle\rangle$ of valuation $\nu$.

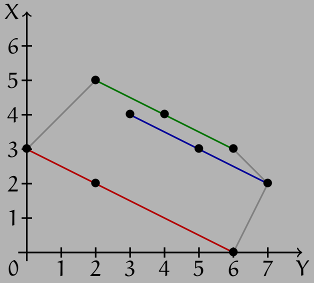



Newton-Puiseux Theorem

For each edge in the **lower hull** of slope $-\nu$, $f$ has a root $\phi \in \overline{\mathbb{K}}\langle\langle X\rangle\rangle$ of valuation $\nu$.

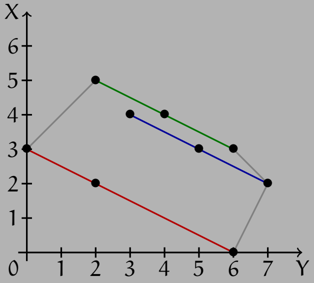
**Corollary**

A polynomial $f \in \mathbb{K}[X, Y]$ has a factor $g$ with a root of valuation $\nu$ iff the Newton polygon of $f$ has an edge of slope $-\nu$.




**Weighted-homogeneity**

A polynomial $g = \sum_j b_j X^{\gamma_j} Y^{\delta_j}$ is $(p, q)$**-homogeneous** of order $\omega$ if $p\gamma_j + q\delta_j = \omega$ for all j.




**Weighted-homogeneity**

A polynomial $g = \sum_j b_j X^{\gamma_j} Y^{\delta_j}$ is $(p, q)$-**homogeneous** of order $\omega$ if $p\gamma_j + q\delta_j = \omega$ for all $j$.

| Weighted-homogeneous factors | Non-homogeneous factors |
|---|---|
| ⇕ | ⇕ |
| Only one valuation | Two distinct valuations |
| ⇕ | ⇕ |
| Unidimensional Newton polygons | Bidimensional Newton polygons |
| ⇕ | ⇕ |
| Univariate lacunary factorization | Bivariate low-degree factorization |

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The non-homogeneous degree-$d$ factors of $f$.

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The non-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The non-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of non-parallel edges of slopes $-\nu_1$, $-\nu_2$:
   2.1 Write $f = X^{a_1} Y^{b_1} f_1 + \cdots + X^{a_s} Y^{b_s} f_s$ using the Gap Theorem with $\nu_1$ and $\nu_2$, s.t. $\sum_t \deg(f_t) \leqslant \mathcal{O}(k^2 d^4)$;

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The non-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of non-parallel edges of slopes $-\nu_1$, $-\nu_2$:
   2.1 Write $f = X^{a_1}Y^{b_1}f_1 + \cdots + X^{a_s}Y^{b_s}f_s$ using the Gap Theorem with $\nu_1$ and $\nu_2$, s.t. $\sum_t \deg(f_t) \leqslant \mathcal{O}(k^2d^4)$;
   2.2 Compute the degree-$d$ factors of $\gcd(f_1,\ldots,f_s)$;
   $\rightsquigarrow$ **low-degree bivariate factorization**

# *Non-homogeneous factors*

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The non-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of non-parallel edges of slopes $-\nu_1$, $-\nu_2$:
   2.1 Write $f = X^{a_1}Y^{b_1}f_1 + \cdots + X^{a_s}Y^{b_s}f_s$ using the Gap Theorem with $\nu_1$ and $\nu_2$, s.t. $\sum_t \deg(f_t) \leqslant \mathcal{O}(k^2 d^4)$;
   2.2 Compute the degree-$d$ factors of $\gcd(f_1, \ldots, f_s)$;
   $\rightsquigarrow$ **low-degree bivariate factorization**
3. Return the union of the sets of factors, with multiplicity.

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The weighted-homogeneous degree-$d$ factors of $f$.

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The weighted-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The weighted-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of parallel edges, of slope $-q/p$:
   2.1 Write $f = f_1 + \cdots + f_s$ as a sum of $(p, q)$-homogeneous polynomials;

**Input:** $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

**Output:** The weighted-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of parallel edges, of slope $-q/p$:
   - 2.1 Write $f = f_1 + \cdots + f_s$ as a sum of $(p, q)$-homogeneous polynomials;
   - 2.2 Compute the degree-$\frac{d}{q}$ factors of the $f_t(X^{\frac{1}{q}}, 1)$'s;
     $\rightsquigarrow$ **univariate lacunary factorization**

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The weighted-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of parallel edges, of slope $-q/p$:
   - 2.1 Write $f = f_1 + \cdots + f_s$ as a sum of $(p, q)$-homogeneous polynomials;
   - 2.2 Compute the degree-$\frac{d}{q}$ factors of the $f_t(X^{\frac{1}{q}}, 1)$'s;
     $\rightsquigarrow$ **univariate lacunary factorization**
   - 2.3 Compute $Y^{p\,\deg(g)} g(X^q/Y^p)$ for each common factor $g$.

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$ and $d \in \mathbb{Z}_+$.

Output: The weighted-homogeneous degree-$d$ factors of $f$.

1. Compute the Newton polygon $N_f$ of $f$;
2. For each pair of parallel edges, of slope $-q/p$:
   - 2.1 Write $f = f_1 + \cdots + f_s$ as a sum of $(p, q)$-homogeneous polynomials;
   - 2.2 Compute the degree-$\frac{d}{q}$ factors of the $f_t(X^{\frac{1}{q}}, 1)$'s; $\rightsquigarrow$ **univariate lacunary factorization**
   - 2.3 Compute $Y^{p\deg(g)} g(X^q/Y^p)$ for each common factor $g$.
3. Return the union of the sets of factors, with multiplicity.

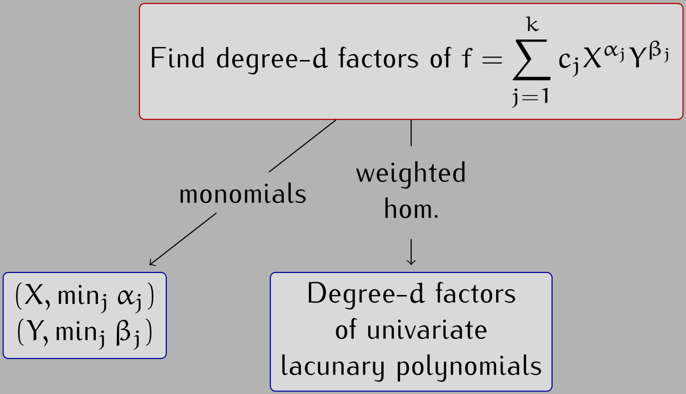
# *Complete algorithm*

Find degree-$d$ factors of $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

Find degree-$d$ factors of $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

monomials

$(X, \min_j \alpha_j)$
$(Y, \min_j \beta_j)$




Available for $\mathbb{Q}(\alpha)$ only
Impossible for $\overline{\mathbb{Q}}$, $\mathbb{C}$

Find degree-$d$ factors of $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$

- monomials → $(X, \min_j \alpha_j)$, $(Y, \min_j \beta_j)$
- weighted hom. → Degree-$d$ factors of univariate lacunary polynomials
  - Available for $\mathbb{Q}(\alpha)$ only
  - Impossible for $\overline{\mathbb{Q}}$, $\mathbb{C}$
- non-hom. → Common factors of $f_t = \sum_{j=j_t}^{j_t+\ell_t-1} c_j X^{\alpha_j} Y^{\beta_j}$ $(\deg(f_t) \leqslant \mathcal{O}(\ell_t^2 d^4))$
  - Low-degree factorization
  - $\mathbb{Q}(\alpha), \overline{\mathbb{Q}}, \mathbb{R}, \mathbb{C}, \mathbb{Q}_p$, etc.

$$\text{Degree-}d\text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

$$\text{Degree-}d\text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

- Do not compute the $n$-dimensional Newton polytope!

# *Multivariate polynomials*

$$\text{Degree-}d\text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

- Do not compute the $n$-dimensional Newton polytope!
- For all $i < j$, compute the Newton polygon $N_{i,j}$ of $f \in R[X_i, X_j]$ where $R = \mathbb{K}[\mathbf{X} \setminus X_i, X_j]$;

# *Multivariate polynomials*

Degree-$d$ factors of $f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$

- Do not compute the $n$-dimensional Newton polytope!
- For all $i < j$, compute the Newton polygon $N_{i,j}$ of $f \in R[X_i, X_j]$ where $R = \mathbb{K}[\mathbf{X} \setminus X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ unidimensional factors

$$\text{Degree-}d\text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

- Do not compute the $n$-dimensional Newton polytope!
- For all $i < j$, compute the Newton polygon $N_{i,j}$ of $f \in R[X_i, X_j]$ where $R = \mathbb{K}[\mathbf{X} \setminus X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ unidimensional factors
  - Every $N_{i,j}$ is unidimensional (or 0-dimensional)
  - **Univariate lacunary factorization**

$$\text{Degree-}d \text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

- Do not compute the $n$-dimensional Newton polytope!
- For all $i < j$, compute the Newton polygon $N_{i,j}$ of $f \in R[X_i, X_j]$ where $R = \mathbb{K}[\mathbf{X} \setminus X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ unidimensional factors
  - Every $N_{i,j}$ is unidimensional (or 0-dimensional)
  - **Univariate lacunary factorization**
- Non-homogeneous factors $\rightsquigarrow$ multidimensional factors

## *Multivariate polynomials*

$$\text{Degree-}d \text{ factors of } f = \sum_{j=1}^{k} c_j X_1^{\alpha_{1,j}} \cdots X_n^{\alpha_{n,j}}$$

- Do not compute the $n$-dimensional Newton polytope!
- For all $i < j$, compute the Newton polygon $N_{i,j}$ of $f \in R[X_i, X_j]$ where $R = \mathbb{K}[\mathbf{X} \setminus X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ unidimensional factors
  - Every $N_{i,j}$ is unidimensional (or 0-dimensional)
  - **Univariate lacunary factorization**
- Non-homogeneous factors $\rightsquigarrow$ multidimensional factors
  - At least one $N_{i,j}$ is multidimensional
  - **Multivariate low-degree factorization**

- Consider $f$ as before, and let $g$ be a multidimensional factor of $f$:
  - If "$X_i \notin g$", $g$ divides each coefficient of $f \in \mathbb{K}[\mathbf{X} \setminus X_i][X_i]$;
  - Else $N_{i,j}(g)$ is multidimensional for some $j$.

- Consider $f$ as before, and let $g$ be a multidimensional factor of $f$:
  - If "$X_i \notin g$", $g$ divides each coefficient of $f \in \mathbb{K}[\mathbf{X} \setminus X_i][X_i]$;
  - Else $N_{i,j}(g)$ is multidimensional for some $j$.

1. Let $\mathcal{H} = \{f\}$;
2. For each variable $X_i$ and each $h \in \mathcal{H}$:
   2.1 Partition $h = \sum_d h_i(\mathbf{X} \setminus X_i) X_i^d$;
   2.2 For each $X_j$ such that $N_{i,j}(h)$ is multidimensional, partition $h$ with respect to each pair of non-parallel edges in $N_{i,j}(h)$;
   2.3 *Merge* those $\mathcal{O}(nk^2)$ partitions to get $\mathcal{H}_h$;
   2.4 Replace $h$ by the elements of $\mathcal{H}_h$ in $\mathcal{H}$.
3. Return the degree-$d$ factors of $\gcd(\mathcal{H}^\circ)$.

# *Implementation - 1/2*

```
Mmx] use "lacunaryx";
     X == coordinate('x); x == mvpolynomial(1:>Integer, X);
     Y == coordinate('y); y == mvpolynomial(1:>Integer, Y);
                                                                  49 msec

Mmx] c () :Integer == (-1)^(random() rem 2)*(random() rem 10);
     lin () : MVPolynomial(Integer) == c()*x + c()*y + c();
     quad () : MVPolynomial(Integer) == c()*x^2+c()*x*y+c()*y^2+c()*x+c()*y+c();
     randpol (): MVPolynomial(Integer) == {
       p: MVPolynomial(Integer) := mvpolynomial(1:>Integer);
       q: MVPolynomial(Integer) := mvpolynomial(0:>Integer);
       for i:Int in 1 to 10 do {
         l == lin(); e == 1+random() rem 3; p*=l^e;
         mmout << "(" << l << ")^" << e << " ; ";}
       for i:Int in 1 to 30 do q+= c()*x^random()*y^random() * quad();
       p*q;};
     d (p: MVPolynomial(Integer)) == if deg(p) < 0 then deg(p)+2^32 else deg(p);
     test () : Void == { p == randpol(); mmout << lf << "Polynomial of degree
     " << d(p) << " with " << #(p) << " nonzero monomials." << lf << "Linear
     factors: " << linear_factors (p) << lf;};
                                                                  26 msec
```

# *Implementation - 2/2*

```
Mmx] test();
```

(5 y + 7)^3 ; (3 y + 2 x + 1)^1 ; (3 x − 4)^2 ; (−8 y + 7 x + 9)^2 ; (4 y − x − 6)^3 ; (4 y + 4 x + 5)^1 ; (−4 y + 6 x − 6)^1 ; (3 y − 2 x − 8)^3 ; (y + 7 x + 2)^1 ; (−8 y + 9 x − 2)^2 ;

Polynomial of degree 3181006535 with 6801 nonzero monomials.

Linear factors: $[[x, 61475114], [3\,x-4, 2], [y, 76556243], [5\,y+7, 3], [3\,y+2\,x+1, 1], [y+7\,x+2, 1], [4\,y+4\,x+5, 1], [-2\,y+3\,x-3, 1], [-8\,y+9\,x-2, 2], [-8\,y+7\,x+9, 2], [-3\,y+2\,x+8, 3], [-4\,y+x+6, 3]]$

963 msec

```
Mmx] test();
```

(−5 y − 3 x − 2)^3 ; (−8 y + 8)^1 ; (2 y + 8 x − 6)^2 ; (y + 6 x − 6)^2 ; (y + x + 4)^3 ; (2 y − 6 x + 3)^3 ; (7 y + 4 x)^3 ; (−y − 6 x + 1)^3 ; (7 x + 1)^3 ; (y + 7 x + 6)^2 ;

Polynomial of degree 3310508792 with 10976 nonzero monomials.

Linear factors: $[[x, 41780031], [7\,x+1, 3], [y, 436756], [y-1, 1], [7\,y+4\,x, 3], [y+6\,x-6, 2], [y+7\,x+6, 2], [y+4\,x-3, 2], [-2\,y+6\,x-3, 3], [y+6\,x-1, 3], [y+x+4, 3], [5\,y+3\,x+2, 3]]$

2.385 sec

# Conclusion

- Computing low-degree factors of lacunary multivariate polynomials

# Conclusion

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$

# Conclusion

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$
  - "Field-independent"
  - **Simpler** and **more general** than previous algorithms

# Conclusion

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$
  - “Field-independent”
  - **Simpler** and **more general** than previous algorithms
  - Implementation: work in progress

# Conclusion

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$
  - "Field-independent"
  - **Simpler** and **more general** than previous algorithms
  - Implementation: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: arithmetic circuits

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$
  - "Field-independent"
  - **Simpler** and **more general** than previous algorithms
  - Implementation: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: arithmetic circuits
  - Degree-$d$ factors in **positive characteristic**?
  - **Small positive characteristic**?

- Computing low-degree factors of lacunary multivariate polynomials
  - Reduction to $\begin{cases}\text{univariate lacunary polynomials}\\ \text{low-degree multivariate polynomials}\end{cases}$
  - "Field-independent"
  - **Simpler** and **more general** than previous algorithms
  - Implementation: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: arithmetic circuits
  - Degree-$d$ factors in **positive characteristic**?
  - **Small positive characteristic**?

# Thank you!

arXiv:1401.4720