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

Bruno Grenet
LIX — École Polytechnique

ISSAC 2014 — Kobe, Japan
July 23., 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 $f_1, \dots, f_t$, 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** $\deg(f)$

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** $\deg(f)$

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

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

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

Definition

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

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

Definition

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

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

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]

Definition

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

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

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]

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

Theorem

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

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

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

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

Theorem

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.

- New algorithm for $\mathbb{K} = \mathbb{Q}(\alpha)$; **some** factors for $\mathbb{K} = \overline{\mathbb{Q}}, \mathbb{R}, \mathbb{C}, \mathbb{Q}_p$

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

Theorem

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.

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

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

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

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

Gap Theorem

Suppose that $f = f_1 + f_2$ with $\mathrm{val}_X(f_2) > \mathrm{val}_X(f_1) + \binom{\# f_1}{2}$. Then for all $uv \neq 0$, $(Y - uX - v)$ divides f iff it divides both f_1 and f_2.

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

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X)) \in \overline{\mathbb{K}(X)}[Y]$$

Observation for low-degree factors

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

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X)) \in \overline{\mathbb{K}(X)}[Y]$$

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

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

Observation for low-degree factors

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

$$g(X,Y) = g_0(X) \prod_{i=1}^{\deg_Y(g)} (Y - \phi_i(X)) \in \overline{\mathbb{K}(X)}[Y]$$

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

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

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

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 ν.

If the family $(X^{\alpha_j}\phi^{\beta_j})_j$ is linearly independent,

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

Gap Theorem

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 $\alpha_1 + \nu\beta_1 \leqslant \cdots \leqslant \alpha_k + \nu\beta_k$. Let g a degree-d irreducible polynomial, with a root of valuation ν.
If ℓ is the smallest index s.t.
$$\alpha_{\ell+1} + \nu\beta_{\ell+1} > (\alpha_1 + \nu\beta_1) + (2d(4d+1) - \nu)\binom{\ell}{2},$$

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

Gap Theorem

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 $\alpha_1 + \nu\beta_1 \leqslant \cdots \leqslant \alpha_k + \nu\beta_k$. Let g a degree-d irreducible polynomial, with a root of valuation ν.
If ℓ is the smallest index s.t.

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

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

- Depends (only) on ν.
- Bounds the growth of $\alpha_j + \nu\beta_j$ in f_1 (neither α_j nor β_j)

Technical 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 $|\beta_p - \beta_q| \leqslant \mathcal{O}(\ell^2 d^4)$.

Technical 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

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

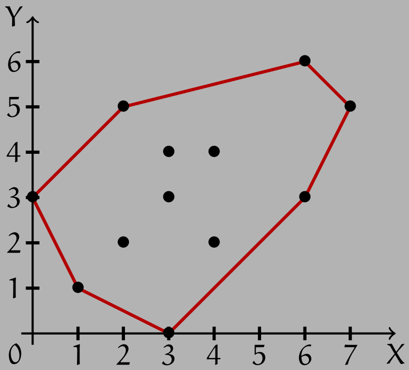
Then for all p, q, $|\alpha_p - \alpha_q| \leqslant \mathcal{O}(\ell^2 d^4)$ and $|\beta_p - \beta_q| \leqslant \mathcal{O}(\ell^2 d^4)$.

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$, $d \in \mathbb{Z}_+$ and $\nu_1, \nu_2 \in \mathbb{Q}$

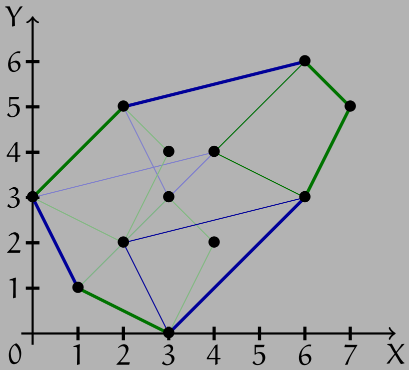
Output Degree-d factors of f, having roots of valuations ν_1 and ν_2

1. Write $f = f_1 + \cdots + f_s$, using the Gap Theorem w.r.t. ν_1 and ν_2;
2. Write each $f_t = X^a Y^b f_t^\circ$, where $\deg(f_t^\circ) \leqslant \mathcal{O}(\ell^2 d^4)$;
3. Factor $\gcd(f_1^\circ, \ldots, f_t^\circ)$. $\rightsquigarrow$ low-degree bivariate factorization

Newton polygon and Puiseux series



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



Ostrowski Theorem

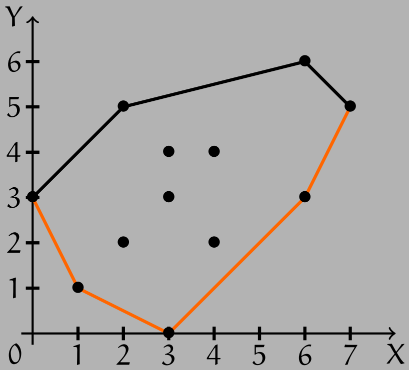
If $f = gh$, then

$\mathrm{Newt}(f) = \mathrm{Newt}(g) + \mathrm{Newt}(h)$.

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

$$= (X - 2Y^2 + Y^3X^4)(X^2 + 2Y - Y^2X^3 + Y^3X^2)$$

Newton polygon and Puiseux series



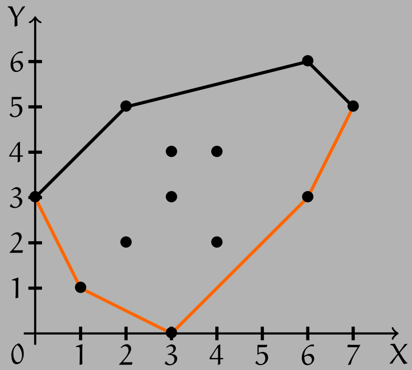
Ostrowski Theorem

If $f = gh$, then $\mathrm{Newt}(f) = \mathrm{Newt}(g) + \mathrm{Newt}(h)$.

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 ν.

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



Ostrowski Theorem

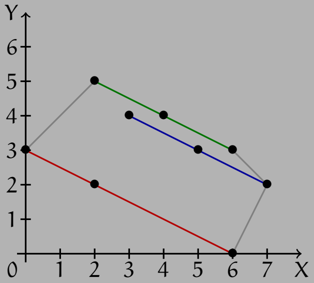
If $f = gh$, then
$\mathrm{Newt}(f) = \mathrm{Newt}(g) + \mathrm{Newt}(h)$.

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 ν.

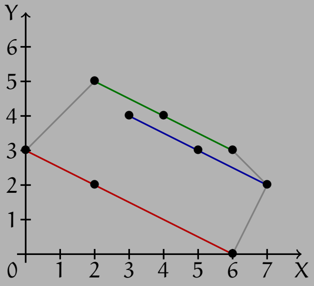
Corollary

For $f \in \mathbb{K}[X, Y]$ to have a factor g with a root of valuation ν, its Newton polygon needs to have 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 ω if $p\gamma_j + q\delta_j = \omega$ for all j.



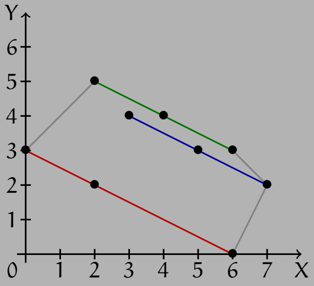
Weighted-homogeneity

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

If f, g are (p, q)-homogeneous:

g divides $f \iff$

$g(X^{1/q}, 1)$ divides $f(X^{1/q}, 1)$



Weighted-homogeneity

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

If f, g are (p,q)-homogeneous:

g divides $f \iff$

$g(X^{1/q}, 1)$ divides $f(X^{1/q}, 1)$

Input: $f = \sum_{j=1}^{k} c_j X^{\alpha_j} Y^{\beta_j}$, $d \in \mathbb{Z}_+$ and $\nu = p/q \in \mathbb{Q}$

Output Degree-d (p,q)-homogeneous factors of f

1. Write $f = f_1 + \cdots + f_s$ as a sum of (p,q)-hom. polynomials;
2. Compute the common degree-(d/q) factors of the $f_t(X^{1/q}, 1)$'s;
   $\rightsquigarrow$ **univariate lacunary factorization**
3. Return $Y^{p \deg(g)} g(X^q / Y^p)$ for each factor g.

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

Output: The irreducible degree-d factors of f, with multiplicity.

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

Output: The irreducible degree-d factors of f, with multiplicity.

1. Compute $\mathrm{Newt}(f)$, and the possible valuations $\nu = p/q$ of its roots, with $p, q \leqslant d$;

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

Output: The irreducible degree-d factors of f, with multiplicity.

1. Compute Newt(f), and the possible valuations $\nu = p/q$ of its roots, with $p, q \leqslant d$;
2. For each $\nu = p/q$, compute the (p, q)-homogeneous factors;
   - Lacunary univariate polynomials
   - Known polytime algorithm for $\mathbb{Q}(\alpha)$ only; exponential for $\overline{\mathbb{Q}}$, $\mathbb{C}$

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

Output: The irreducible degree-d factors of f, with multiplicity.

1. Compute Newt(f), and the possible valuations $\nu = p/q$ of its roots, with $p, q \leqslant d$;
2. For each $\nu = p/q$, compute the (p, q)-homogeneous factors;
   - Lacunary univariate polynomials
   - Known polytime algorithm for $\mathbb{Q}(\alpha)$ only; exponential for $\overline{\mathbb{Q}}$, $\mathbb{C}$
3. For each pair (ν_1, ν_2), compute the non-homogeneous factors with roots of valuations ν_1 and ν_2;
   - Low-degree bivariate polynomials
   - Known polytime algorithms for $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Q}_p$, etc.

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

Output: The irreducible degree-d factors of f, with multiplicity.

1. Compute $\mathrm{Newt}(f)$, and the possible valuations $\nu = p/q$ of its roots, with $p, q \leqslant d$;
2. For each $\nu = p/q$, compute the (p, q)-homogeneous factors;
   - Lacunary univariate polynomials
   - Known polytime algorithm for $\mathbb{Q}(\alpha)$ only; exponential for $\overline{\mathbb{Q}}$, $\mathbb{C}$
3. For each pair (ν_1, ν_2), compute the non-homogeneous factors with roots of valuations ν_1 and ν_2;
   - Low-degree bivariate polynomials
   - Known polytime algorithms for $\mathbb{Q}(\alpha)$, $\overline{\mathbb{Q}}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Q}_p$, etc.
4. Return the union of the sets of factors, with multiplicity.

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

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

- **Do not** compute the n-dimensional Newton polytope!
- Compute the Newton polygons $N_{i,j}$ of $f \in \mathbb{K}[\mathbf{X} \setminus X_i, X_j][X_i, X_j]$;

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

- **Do not** compute the n-dimensional Newton polytope!
- Compute the Newton polygons $N_{i,j}$ of $f \in \mathbb{K}[\mathbf{X} \setminus X_i, X_j][X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ 1-dimensional factors
  - Every $N_{i,j}$ is 1-dimensional (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!
- Compute the Newton polygons $N_{i,j}$ of $f \in \mathbb{K}[\mathbf{X} \setminus X_i, X_j][X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ 1-dimensional factors
  - Every $N_{i,j}$ is 1-dimensional (or 0-dimensional)
  - **Univariate lacunary factorization**
- Non-homogeneous factors $\rightsquigarrow$ multidimensional factors
  - At least one $N_{i,j}$ is 2-dimensional
  - **Multivariate low-degree 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!
- Compute the Newton polygons $N_{i,j}$ of $f \in \mathbb{K}[\mathbf{X} \setminus X_i, X_j][X_i, X_j]$;
- Weighted homogeneous factors $\rightsquigarrow$ 1-dimensional factors
  - Every $N_{i,j}$ is 1-dimensional (or 0-dimensional)
  - **Univariate lacunary factorization**
- Non-homogeneous factors $\rightsquigarrow$ multidimensional factors
  - At least one $N_{i,j}$ is 2-dimensional
  - **Multivariate low-degree factorization**
- New ingredient: *Merge* the partitions of f, to avoid exponential growth in the number of low-degree polynomials

`http://www.mathemagix.org/ > Packages > Lacunaryx`

Factorization-related algorithms for lacunary polynomials

- Integer roots of lacunary univariate polynomials
- Linear factors of lacunary univariate and bivariate polynomials
- Very large degree polynomials (G. Lecerf)

- Example: Integer roots of p with $\deg(p) \simeq 2^{185}$ and $\#p \simeq 100\,000$ in < 10 seconds

```
Mmx] use "lacunaryx"; x : LPolynomial Integer == lpolynomial(1,1);
     p == x^3*(x-2)*(2*x+3)^2*(-x+3)*(2*x+7)*(x^2+x+1)*(3*x+5);
     q == x^3 - 6 - 2*x^4 + 12*x + x^5 - 6*x^2+ 3*x^1345 - 6*x^1346 + 3*x^1347 +
          8*x^432534 - 18*x^432535 + 12*x^432536 - 2*x^432537 + 1 - 2*x + x^2;
     e : Integer == 3515401450404015230143514;
     r == 1 + 3*x^1345 - 2*(x-4)*x^e + (x^3-6)*x^(2*e);
     pqr == p*q*r; (log deg pqr/log 2, #pqr)
```

$(85.861891823199, 149)$

49 msec

```
Mmx] roots pqr
```

$[[2, 1], [3, 1], [0, 3], [1, 2]]$

43 msec

```
Mmx] X == coordinate ('x); x : LMVPolynomial Integer == lmvpolynomial(1, X);
     Y == coordinate ('y); y : LMVPolynomial Integer == lmvpolynomial(1, Y);
     f == x^2*y*(x-2)*(2*y+3)^2*(y-x+3)*(2*x+7*y)*(x*y+x+1)*(3*x-6*y+5);
     g == x^3*y^54354165 - 6*y^54354165 - 2*x^4*y^54354164 + 12*x*y^54354164
     + x^5*y^54354163 - 6*x^2*y^54354163 + 3*x^1345*y^54336 - 6*x^1346*y^54335
     + 3*x^1347*y^54334 + 8*x^432534*y^5 - 18*x^432535*y^4 + 12*x^432536*y^3 -
     2*x^432537*y^2 + y^2 - 2*x*y + x^2;
     h == 1 + 3*x^1345*y^54334 - 2*(x-4*y)*x^e*y^2 + (x^3-6)*y^(2*e);
     fgh == f*g*h; (log deg fgh/log 2, #fgh)
```

$(85.861891823199, 1028)$

60 msec

```
Mmx] linear_factors fgh
```

$[[x, 2], [-x+2, 1], [y, 1], [2\,y+3, 2], [-y+x, 2], [-7\,y-2\,x, 1], [-y+x-3, 1], [-6\,y+3\,x+5, 1]]$

299 msec

Conclusion

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

- 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
  - Partial results in **large positive characteristic** [CGKPS'13]
  - Implementation within `Mathemagix`: 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
  - Partial results in **large positive characteristic** [CGKPS'13]
  - Implementation within `Mathemagix`: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: SLP/arithmetic circuits

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
  - Partial results in **large positive characteristic** [CGKPS'13]
  - Implementation within `Mathemagix`: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: SLP/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
  - Partial results in **large positive characteristic** [CGKPS'13]
  - Implementation within `Mathemagix`: work in progress
- Open questions:
  - **Lacunary factors** in polynomial time?
  - More general settings: SLP/arithmetic circuits

  - Degree-d factors in **positive characteristic**?
  - **Small positive characteristic**?

ありがとう