Towards fast Puiseux series computation

Adrien Poteaux[⋆], Marc Rybowicz[†]

[⋆]: LIFL - Université Lille 1

[†]: XLIM - Université de Limoges

Computer algebra and polynomials Workshop, Linz

November 25th, 2013

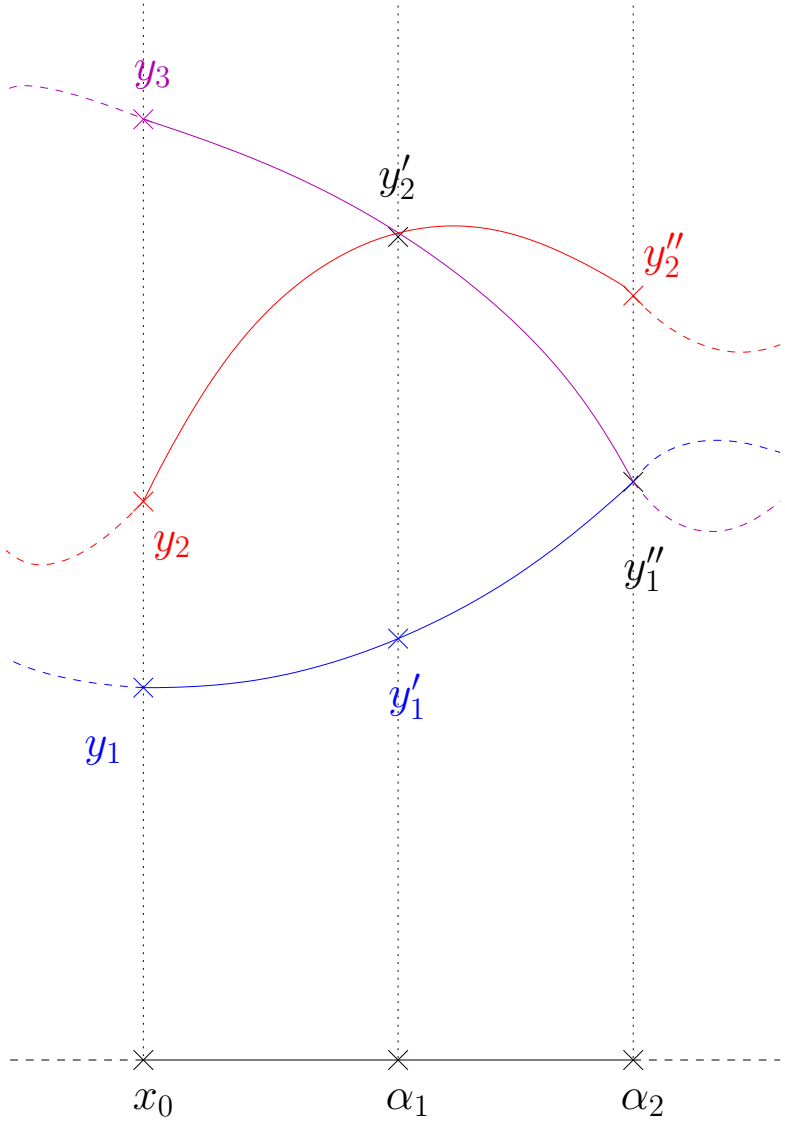
Algebraic plane curves: a projective point of view

- $\mathbb{K} = \mathbb{Q}(\alpha)$ a number field
- $F(X, Y) \in \mathbb{K}[X, Y]$
- $\mathcal{C} = \{(x, y) \in \mathbb{C}^2 \mid F(x, y) = 0\}$

Let $x_0 \in \mathbb{C}$:

- **Fiber** at x_0:
  $\mathcal{F}(x_0) = \{\text{roots of } F(x_0, Y) = 0\}$.
- **Regular point** : $\#\mathcal{F}(x_0) = d_Y$.
- **Critical point** : $\#\mathcal{F}(x_0) < d_Y$.

  $\Longrightarrow$ roots of $R_F = \mathtt{Res}_Y(F, F_Y)$.



Puiseux series: a generalization of formal power series

- x_0 regular ; $\mathcal{F}(x_0) = \{y_1, \cdots, y_{d_Y}\}$.

Theorem (Implicit function theorem)

There are d_Y *series* $Y_i(X) = \sum_{k=0}^{\infty} \alpha_{ik}(X - x_0)^k$ *s.t*

$F(X, Y_i(X)) = 0$ *around* x_0 *and* $Y_i(x_0) = y_i$.

Puiseux series: a generalization of formal power series

- x_0 regular ; $\mathcal{F}(x_0) = \{y_1, \cdots, y_{d_Y}\}$.

Theorem (Implicit function theorem)

There are d_Y series $Y_i(X) = \sum_{k=0}^{\infty} \alpha_{ik}(X - x_0)^k$ s.t

$F(X, Y_i(X)) = 0$ *around* x_0 *and* $Y_i(x_0) = y_i$.

- x_0 critical

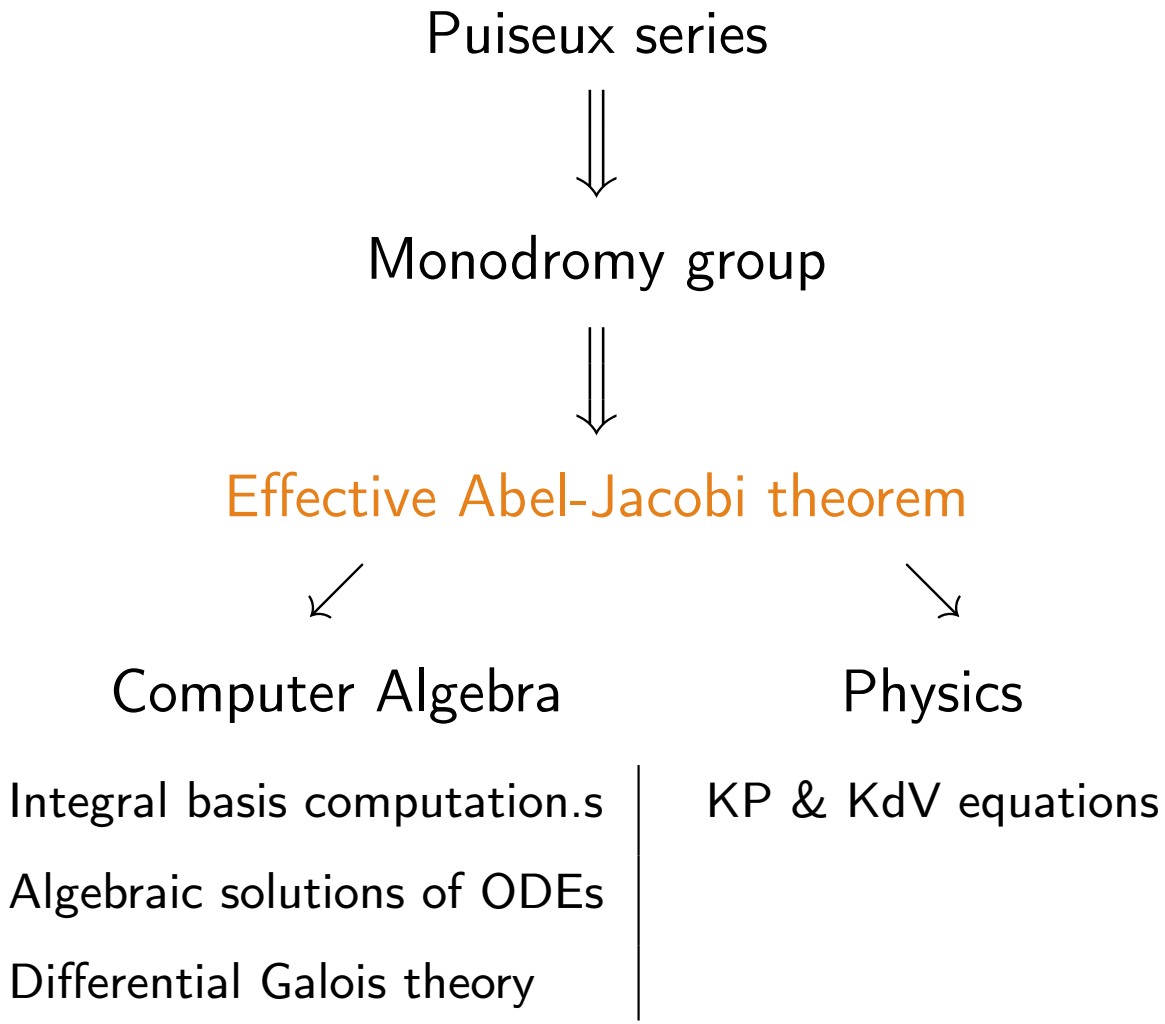
Theorem (Puiseux)

There are d_Y series $Y_{ij}(X) = \sum_{k=n_i}^{\infty} \alpha_{ik}\, \zeta_{e_i}^{jk}\, (X - x_0)^{\frac{k}{e_i}}$ *s.t.*

$F(X, Y_{ij}(X)) = 0$ *for all* $1 \leq j \leq e_i$, $1 \leq i \leq s$, *with*

- ζ_{e_i} *primitive e_i-th root of unity, e_i*
- $e_1, \ldots, e_s$ *partition of d_Y (ramification indices).*

Long term goal



Difficult part is the singular part

$$
\begin{aligned}
S_{ij}(X - x_0) &= \sum_{k=n_i}^{\infty} \alpha_{ik}\, \zeta_{e_i}^{jk}\, (X - x_0)^{\frac{k}{e_i}} \\
&= \sum_{k=n_i}^{r_{ij}} \alpha_{ik}\, \zeta_{e_i}^{jk}\, (X - x_0)^{\frac{k}{e_i}} + \text{ next terms}
\end{aligned}
$$

r_{ij} is the regularity index ; $r_i = r_{ij}$ for $1 \leq j \leq e_i$

Next terms can be computed using quadratic Newton iterations

Kung & Traub 1978, *All Algebraic Functions Can Be Computed Fast*

Singulart part computation: the Newton-Puiseux algorithm

- Newton, 1676 $\rightarrow$ introduction of the concept.
- Puiseux, 1850 $\rightarrow$ rediscovers ; first procedure.
- Chystov, 1986 $\rightarrow$ *"Newton-Puiseux bit complexity is polynomial"*.
- Duval, 1989 $\rightarrow$ rational algorithm ; arithmetic complexity $O(D^8)$.
- Walsh, 2000 $\rightarrow$ bit complexity $\tilde{O}(D^{36})$ (classical algorithm).
- Walsh, 1999 $\rightarrow$ polynomial size for rational coefficients, *no algorithm*.
- Rybowicz & P., 2008 $\rightarrow$ improved arithmetic complexity: $\tilde{O}(D^5)$.

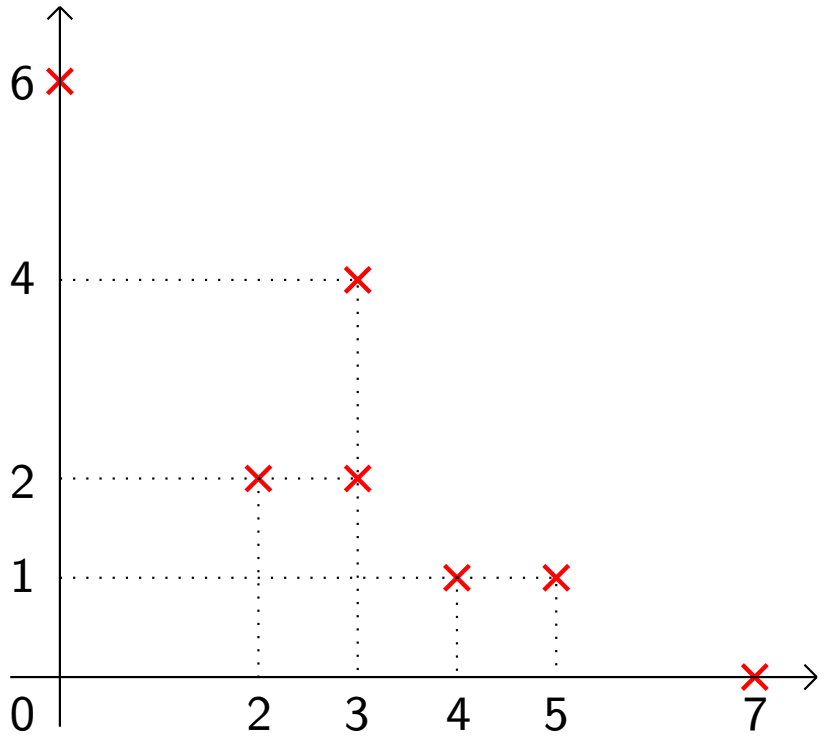
The Newton-Puiseux algorithm: main tools

$$F(X,Y) = Y^7 + Y^5X - 2\,Y^4X + 5\,Y^4X^3 + 4\,Y^2X^2 + X^6$$

Support of a polynomial

$$F(X,Y) = Y^7\,X^0 + Y^5\,X^1 - 2\,Y^4X^1 + 5\,Y^3X^4 - Y^3X^2 + 4\,Y^2X^2 + Y^0X^6$$

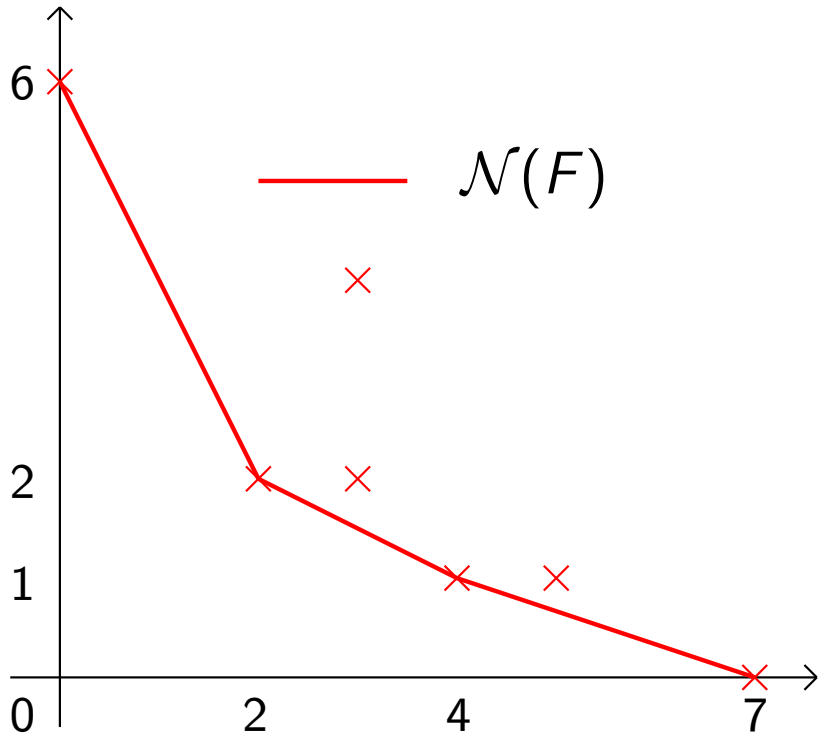
- Supp(F)$= \{(i,j) \in \mathbb{N}^2 \mid a_{ij} \neq 0\}$



Newton polygon

$$F(X,Y) = \sum_{i,j} a_{ij} Y^i X^j$$

- × Supp(F)$= \{(i,j) \in \mathbb{N}^2 \mid a_{ij} \neq 0\}$
- — $\mathcal{N}(F)$: lower part of the convex hull of Supp(F).



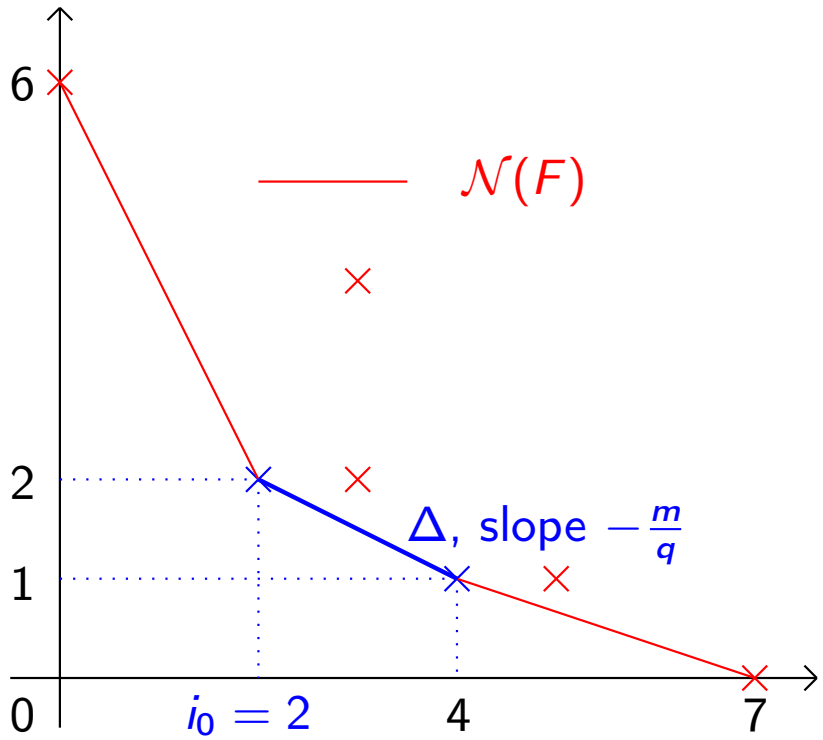
Characteristic polynomial

$$F(X,Y) = \sum_{i,j} a_{ij} Y^i X^j$$

- × Supp(F)$= \{(i,j) \in \mathbb{N}^2 \mid a_{ij} \neq 0\}$
- — $\mathcal{N}(F)$: lower part of the convex hull of Supp(F).

Characteristic polynomial:

$$\phi_\Delta(T) = \sum_{(i,j)\in\Delta} a_{ij} T^{\frac{i-i_0}{q}}$$



Rational Newton-Puiseux algorithm

D. Duval 1989, *Rational Puiseux Expansions*

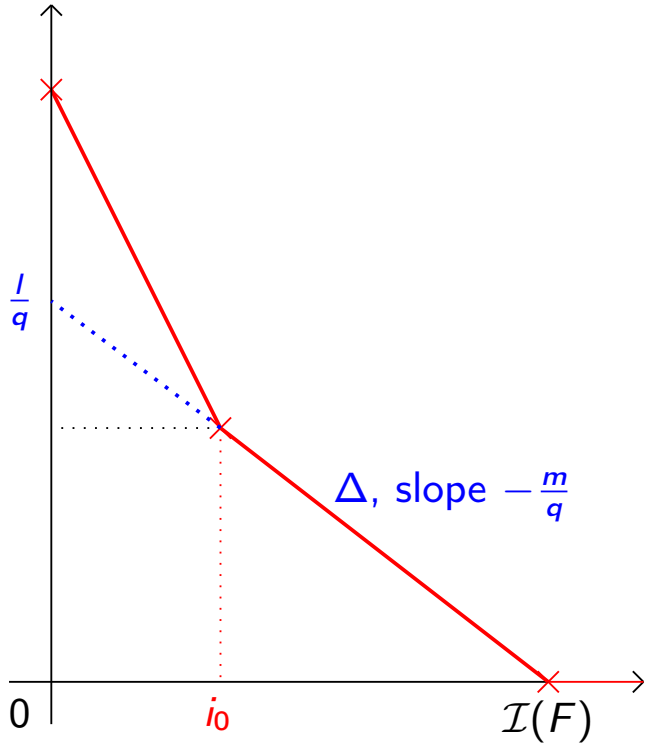
For each edge Δ of $\mathcal{N}(F)$

- $\phi_\Delta = \prod_{k=1}^{s} \phi_k^{M_k}$
- For each ϕ_k

$$F(X,Y) \leftarrow \frac{F(\xi_k^v X^q, X^m(\xi_k^u + Y))}{X^l}$$

with

- ξ_k s. t. $\phi_k(\xi_k) = 0$,
- (u,v) such that $uq - vm = 1$.



Rational Newton-Puiseux algorithm : first turn

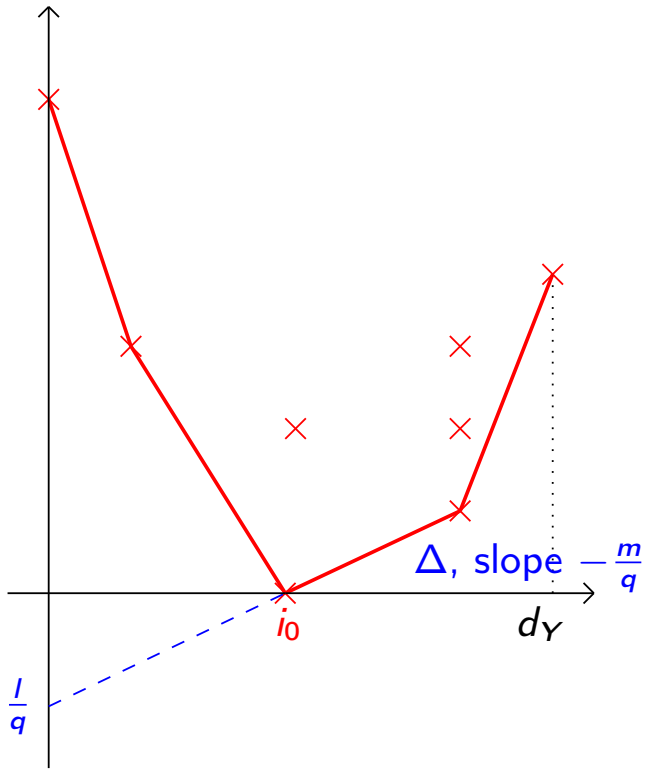
For each edge Δ of $\mathcal{N}_0(F)$

- $\phi_\Delta = \prod_{k=1}^{s} \phi_k^{M_k}$
- For each ϕ_k

$$F(X,Y) \leftarrow \frac{F(\xi_k^v X^q, X^m(\xi_k^u + Y))}{X^l}$$

with

- ξ_k s. t. $\phi_k(\xi_k) = 0$,
- (u,v) such that $uq - vm = 1$.



First turn: initial polygon $\mathcal{N}_0(F)$

Pure symbolic computation is costly

$$H(X,Y) = (Y^3 - X)((Y-1)^2 - X)(Y - 2 - X^2) + X^2\,Y^5$$

$$R_H(X) = X^3\,P(X),\ \deg_X(P) = 23;\ \beta \text{ s.t. } P(\beta) = 0$$

Singular parts of Puiseux series of H above β:

- $S_i(X) = \alpha_{i,0}$, $1 \leq i \leq 4$.
- $S_i(X) = \alpha_{i,0} + \alpha_{i,1}(X-\beta)^{\frac{1}{2}}$, $5 \leq i \leq 6$.

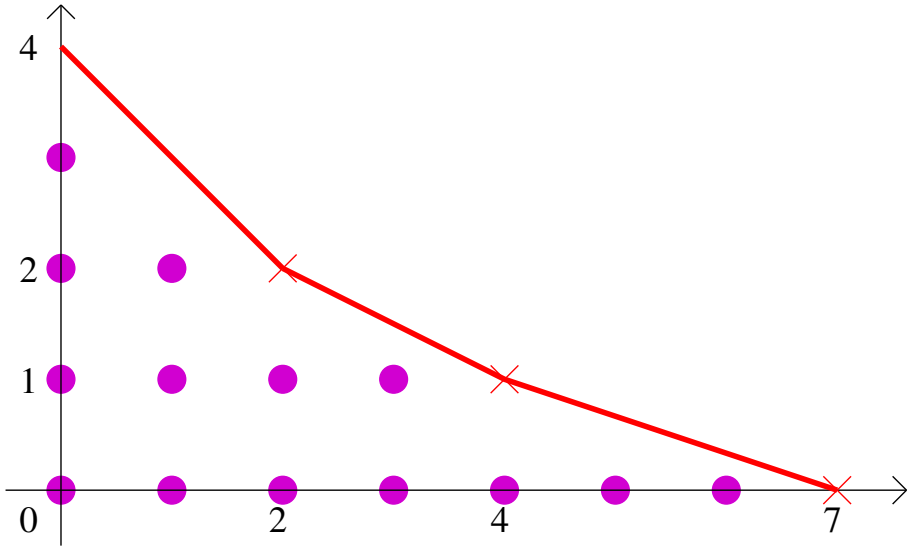
- Degree of the extension field
  - $i = 5, 6$: $\mathbb{K}(\alpha_{i,0}) = \mathbb{K}(\alpha_{i,1}) = \mathbb{K}(\beta) \rightarrow$ extension of degree 23,
  - $i = 1, \ldots, 4$: $[\mathbb{K}(\alpha_{i,0}) : \mathbb{K}(\beta)] = 4 \rightarrow$ extension of degree 92,
- Coefficient growth
  - $\alpha_{i,0} \rightarrow$ rational number with 98 digits,
  - $\alpha_{i,1} \rightarrow$ rational number with 132 digits.

Numerical computations ?

Direct computation: almost useless

Guessing the structure ? two difficulties:

Finding the *correct* Newton polygon



Factorising "well" ϕ_Δ

$$x^2 - 2.0\,x + 0.9999$$
$$\stackrel{?}{=} (x - 0.99)(x - 1.01)$$
$$\stackrel{?}{=} (x - 1.)^2$$

$\Longrightarrow$ Multiplicity structure ?

$\Longrightarrow$ Exact informations needed !

A symbolic-numeric approach:

1. Compute the singular part of Puiseux series modulo a well chosen prime number p

   This give us the structure of the Puiseux series, thus:

   - Newton polygons,
   - Multiplicity structures of the ϕ_Δ.

2. Use this information to conduct a numerical computation of the Puiseux series coefficients.

Modular part: main results

- Reduction criteria:
  1. One can reduce F mod p,
  2. $p > d_Y$
  3. $\mathrm{tc}(R_F) \not\equiv 0 \mod p$.
- If p satisfies that:
  1. Puiseux series can be reduced modulo p,
  2. The structure computed modulo p is the good one.
- Bounds for p ; $\log(p) \simeq \log(D)$ with probabilistic algorithms.
- References for details: Poteaux & Rybowicz 2008, *On the good reduction of Puiseux series and complexity of the Newton-Puiseux algorithm over finite fields* ; Poteaux & Rybowicz 2012, *On the good reduction of Puiseux series and Applications* ; Poteaux & Rybowicz 2011, *Complexity bounds for the rational Newton-Puiseux algorithm over finite fields and related problems*

Numerical part: following the structure

- If $P(X) = (X-\alpha)^m(X-\beta)^m = (X-\tilde{\gamma})^m(X-\tilde{\delta})^m$,

  $$\alpha \leftrightarrow \tilde{\gamma} \text{ or } \alpha \leftrightarrow \tilde{\delta} \ ?$$

  no answer $\Longrightarrow$ group of roots dealt together

- We separate the blocs later on:

  1. filter according to the Newton polygon

     (coefficient of F zero or not)

  2. filter according to the multiplicity structure

     first idea = approximate gcd : singular values

▸ One example

Fast computation ?

1. Use only "mandatory" monomials
   1. Truncating power of X
      - already used in the D^8 of Duval, improved in our D^5 version,
      - better ? $\rightarrow$ relaxed computations (*a-posteriori* bounds)
   2. Reducing the degree in Y
      - After first turn, we only look roots above $(0, 0)$
      - Get rid of roots above $(0, \alpha \neq 0)$? Factorization
2. Fast computation in between two branch separations

$$S(X) = 2 + X - X^2 + X^3 + 3X^{\frac{7}{2}} + 4X^4 - X^{\frac{9}{2}} - 2X^5 + X^{\frac{11}{2}} + 7X^{\frac{35}{6}} + \dots$$

A new algorithm

1. Substitutions $F(X, Y) \leftarrow F(X, Y + A_{d_Y - 1}(X))$

   (*compute common terms at once ; less recursive calls)*

2. Factorization of the polynomial during the algorithm

   (*Hensel lemma* $\rightarrow$ *recursive calls with smaller degrees)*

3. Using relax algorithms.

   (*no a priori bound required)*

4. Improved truncation bounds (thanks to relax algorithms).

leads to $\tilde{O}(D^4)$

Better again ?

The idea: all series do not need the same truncations

1. Compute half of the series in $\tilde{O}(D^3)$ (the one that requires the smallest truncations)
2. Factorize $F = G\,H$ with G corresponding to the computed series, and H to the other ones. $\tilde{O}(D^3)$?
3. Apply the same procedure to H.

Logarithmic number of recursive call: total complexity in $\tilde{O}(D^3)$.

Remaining problem: step 2 ; how to get the starting point in order to apply Hensel lemma ? or another idea ?

Conclusion

- A modular-numerical approach
  - Modular part: 100% done,
  - Numerical part: first strategy developped,
  - Maple implementation (prototype)
  - To be done: certified computations, error bounds, better implementation...
- A new faster algorithm
  - D^4 strategy looks to work,
  - Missing one idea for the D^3 algorithm,
  - C++ implementation (based on NTL) started,
  - To be done: finding this last missing part, writting and coding everything properly.
- Long term: a good effective way to compute Abel's map ?

Annexes

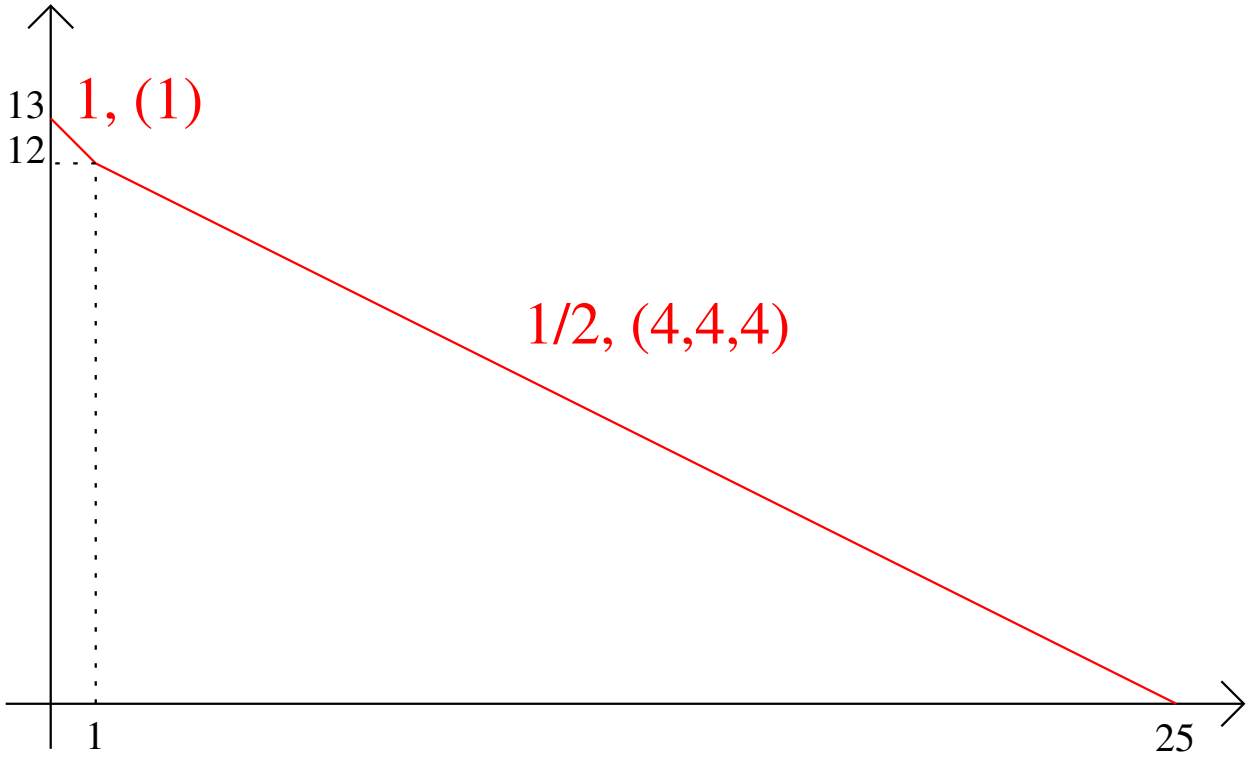
Following the structure numerically: one example

Puiseux series of F :

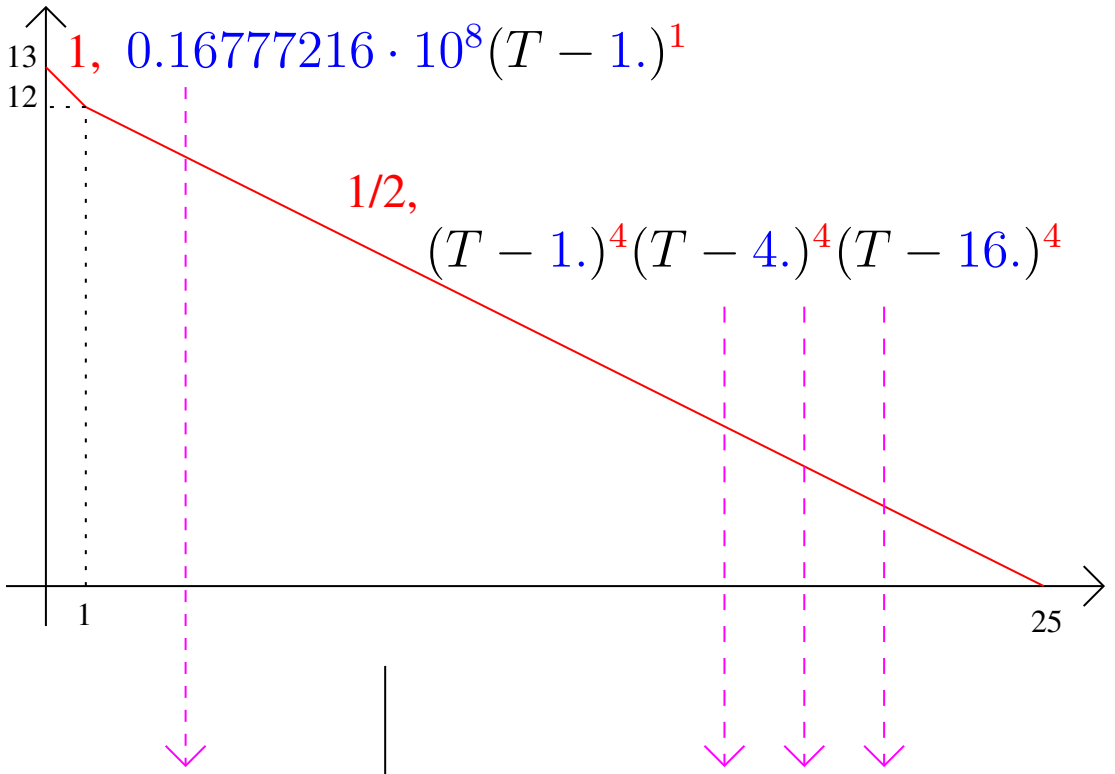
- $S_1(X) = X + \cdots$
- $S_2(X) = 4X^{\frac{1}{2}} + X^{\frac{7}{8}} + \cdots$
- $S_3(X) = 2X^{\frac{1}{2}} + 2X + \cdots$
- $S_4(X) = 2X^{\frac{1}{2}} + X + X^{\frac{7}{6}} + \cdots$
- $S_5(X) = X^{\frac{1}{2}} + 2X + X^{\frac{5}{4}} + \cdots$
- $S_6(X) = X^{\frac{1}{2}} + X + \cdots$
- $S_7(X) = X^{\frac{1}{2}} + 4X + \cdots$

$d_Y = 25, d_X = 26$; $1 \leq$ coefficients $\leq 10^{13}$; $\mathit{Digits} = 20$.

First Newton polygon



First Newton polygon



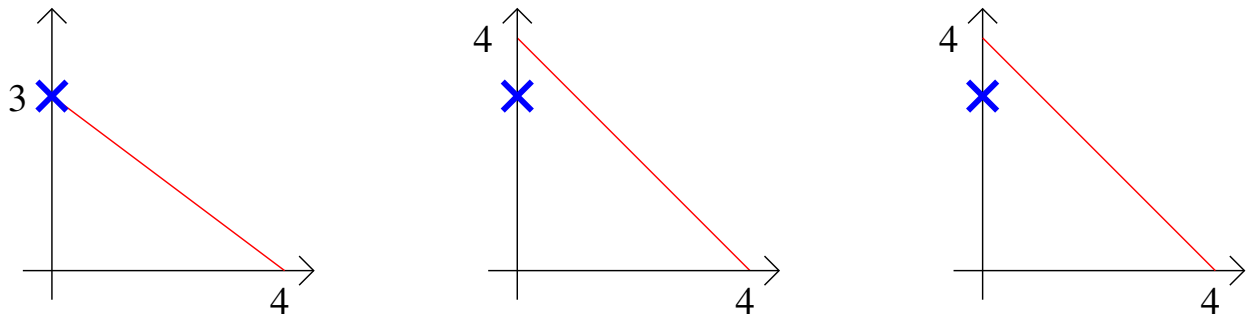
$$S_1(x) = 1.x^{\frac{1}{2}}$$

ZOOM

Filter according to the polygons

$$G_i(X,Y) \leftarrow \frac{F(X^2, X(Y+\xi_i^{1/2}))}{X}, \quad \xi_1 = 1. \ \ \xi_2 = 4. \ \ \xi_3 = 16.$$

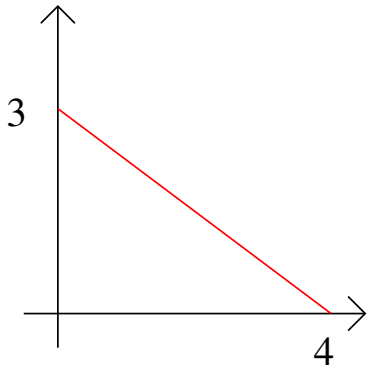
$$\{G_1, G_2, G_3\}$$



polynom	coefficient at X^3
G_1	0.
G_2	0.
G_3	-17199267840000.0

Filter according to the polygons

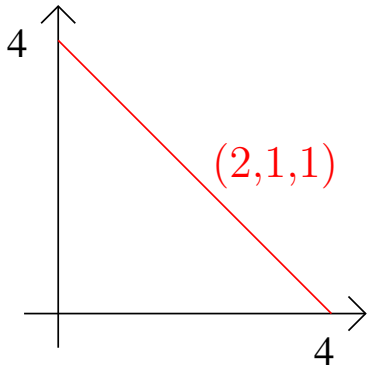
G_3

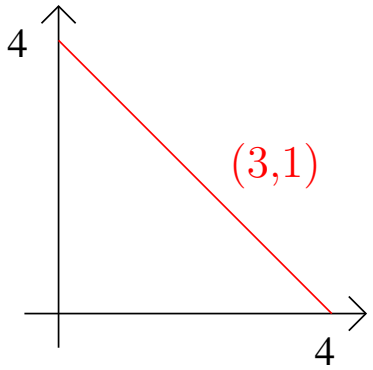


$\phi_3 = 171992678400000.0(T - 1.)^1$

$S_2(x) = 4.x^{\frac{1}{2}} + 1.x^{\frac{7}{8}}$

$\{G_1, G_2\}$





Sorting polynomials according to multiplicity structures

Filter according to the multiplicity structure

Multiplicity Structures:

- $(2,1,1) \Rightarrow \deg(pgcd(\phi,\phi')) = 1$
- $(3,1) \Rightarrow \deg(pgcd(\phi,\phi')) = 2$

Characteristic Polynomials:

$$\phi_1 = 1049760000.0 - 2361960000.0\,T + 1837080000.0\,T^2 - 590490000.0\,T^3 + 65610000.0\,T^4$$

$$\phi_2 = 1719926784.0 - 6019743744.0\,T + 7739670528.0\,T^2 - 4299816960.0\,T^3 + 859963392.0\,T^4$$

1. $S_i \leftarrow Syl(\phi_i, \phi_i')$
2. Computing the singular values of the S_i

Tri selon les multiplicités

Singular values associated to ϕ_1 :

[710694508.4327095884, 5827385163.0346368216, 3038236185.2953794346, 1140210769.8445335036,

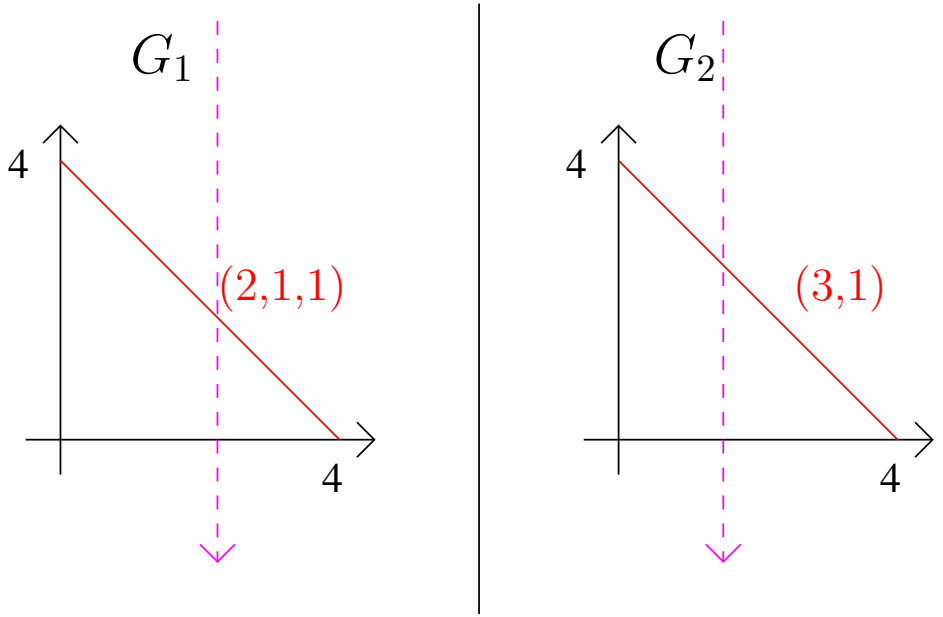
40759543.641844042087, 1882790.0681572535369, $3.8263754075532025314 \cdot 10^{-11}$]

Singular values associated to ϕ_2 :

[37445022322.189717034, 24644791488.066781055, 12101920587.793187214,

3915075466.8959244453, 31534726.725839766232, 0.000000000741011873586170890 31,

0.00000000027761147770454585021]



Result

```
mypuiseux(F,x,y,x,0);
```

$$[[[x = T, y = 1.0\ T], [x = T^2, y = 1.000000000000046423\ T^2 + 1.000000000000014628\ T], [x = T^2, y = 4.000000000000002662\ T^2 + 1.000000000000014628\ T], [x = T^4, y = 0.999999999999869303\ T^5 + 2.000000000000040470\ T^4 + 1.000000000000014628\ T^2], [x = T^2, y = 1.999999999993502275\ T^2 + 2.000000000000757425\ T], [x = T^6, y = 1.000000000036976678\ T^7 + 1.000000000047325425\ T^6 + 2.000000000000757425\ T^3], [x = T^8, y = 0.999999999483964356\ T^7 + 4.000000000009297336\ T^4]]]$$

back