Fast computation of the roots of polynomials over the ring of power series

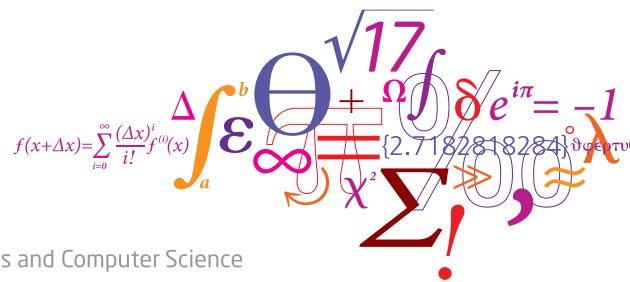
Vincent Neiger[†] — Johan Rosenkilde[†] — Éric Schost[§]

[†] Technical University of Denmark (DTU)

[§] University of Waterloo, Canada

ISSAC'17, Kaiserslautern, Germany

July 26, 2017



DTU Compute

Department of Applied Mathematics and Computer Science

Outline

- Overview of the contribution
- Structure of the set of roots
- Divide and conquer algorithm with degree control

Outline

- Overview of the contribution
- Structure of the set of roots
- Divide and conquer algorithm with degree control

Roots of polynomials with power series coefficients

Data:
- field $\mathbb{K}$ (e.g. finite field)
- precision $d \in \mathbb{Z}_{>0}$
- polynomial $Q(y) \in \mathbb{K}[\![x]\!][y]$

Roots of Q at precision d:

$$\mathrm{Roots}(Q,d) = \{\text{series } f(x) \in \mathbb{K}[\![x]\!] \mid Q(f) = 0 \bmod x^d\}$$

Examples

- Over $\mathbb{K} = \mathbb{F}_2$: $\mathrm{Roots}(y^2+y,1) = (0 + x\mathbb{F}_2[\![x]\!]) \cup (1 + x\mathbb{F}_2[\![x]\!]) = \mathbb{F}_2[\![x]\!]$
- For $Q = (y-x)(y-3+2x)$: $\mathrm{Roots}(Q,5) = (x + x^5\mathbb{K}[\![x]\!]) \cup (3 - 2x + x^5\mathbb{K}[\![x]\!])$
- For $Q = (y - \frac{1}{1-x})^2 = y^2 - 2(\sum_k x^k)y + (\sum_k (k+1)x^k)$, we have

$$\mathrm{Roots}(Q,d) = 1 + x + x^2 + \cdots + x^{d/2-1} + x^{d/2}\mathbb{K}[\![x]\!]$$

Applications from coding theory

list-decoding of Reed-Solomon codes [Guruswami-Sudan, 1999]

Step 1. compute some $Q(y)$ over $\mathbb{F}_q[x]$

Step 2. find all polynomial roots $Q(f) = 0$

a variant of Guruswami-Sudan's algorithm [Wu, 2008]

Step 1. compute some $Q(y)$ over $\mathbb{F}_q[x]$

Step 2. find all rational roots $Q(f/g) = 0$

decoding Hermitian algebraic-geometry codes [Rosenkilde-Beelen, 2015]

Step 1. compute some $Q(y)$ over $\mathbb{F}_q[\![x]\!]$

Step 2. find all series roots $Q(f) = 0 \bmod x^d$

Result

Problem

Given the precision $d \in \mathbb{Z}_{>0}$ and the polynomial $Q \in \mathbb{K}[\![x]\!][y]$,
Compute polynomials and integers $(f_i(x), t_i)_{1 \leqslant i \leqslant \ell}$ such that

$$\mathrm{Roots}(Q, d) = \bigcup_{1 \leqslant i \leqslant \ell} (f_i + x^{t_i}\mathbb{K}[\![x]\!])$$

Note: $Q(y)$ given by its coefficients truncated mod x^d

Result

Problem

Given the precision $d \in \mathbb{Z}_{>0}$ and the polynomial $Q \in \mathbb{K}[\![x]\!][y]$,
Compute polynomials and integers $(f_i(x), t_i)_{1 \leqslant i \leqslant \ell}$ such that

$$\mathrm{Roots}(Q, d) = \bigcup_{1 \leqslant i \leqslant \ell} (f_i + x^{t_i}\mathbb{K}[\![x]\!])$$

Note: $Q(y)$ given by its coefficients truncated mod x^d

Contribution = fastest known algorithm

Cost bound: $\tilde{O}(nd) + O(d\,\mathsf{R}_{\mathbb{K}}(n))$

deterministic, $n = \deg_y(Q)$

cost of root-finding over $\mathbb{K}$

Note: over $\mathbb{K} = \mathbb{F}_q$, Las Vegas root-finding in $\mathsf{R}_{\mathbb{K}}(n) = \tilde{O}(n \log(q))$

Previous algorithms

[Roth-Ruckenstein, 2000]

- **iteration** on the precision d
- **weak degree control** in iterations

$$O(\boldsymbol{n^2 d^2} + d\mathsf{R}_{\mathbb{K}}(n))$$

$\rightsquigarrow$ similar to algorithms for Puiseux series solutions [Poteaux-Rybowicz, 2011+2015]

[Alekhnovich, 2005] and [Rosenkilde-Beelen, 2015]

- **divide and conquer** on the precision d
- **weak degree control** in recursive calls

$$\tilde{O}(\boldsymbol{n^2 d}) + O(d\mathsf{R}_{\mathbb{K}}(n))$$

[Berthomieu-Lecerf-Quintin, 2013]

- general: $Q \in \mathcal{R}[y]$, local domain $\mathcal{R}$
- **degree control** in recursive calls

For $\mathcal{R} = \mathbb{F}_{p^k}[\![x]\!]$:
$\tilde{O}(\boldsymbol{nd^2} + n\log(p^k) + nd\log(k)/p)$

Outline

- Overview of the contribution
- Structure of the set of roots
- Divide and conquer algorithm with degree control

The case of simple roots

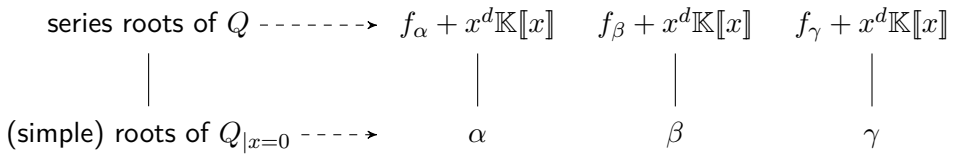
Series root's constant coefficient: root of $Q_{|x=0} \in \mathbb{K}[y]$
$(Q(f) = 0 \bmod x^d \quad \Rightarrow \quad Q_{|x=0}(f(0)) = 0)$

Simple roots and Newton iteration

α a simple root of $Q_{|x=0}$
$\rightsquigarrow$ one series root $f = \alpha + x\hat{f} \in \mathbb{K}[\![x]\!]$

computed via Newton iteration, $f \leftarrow f - \frac{Q(f)}{Q'(f)} \bmod x^{2^i}$ [Kung-Traub, 1978]

$\Rightarrow$ **if all roots of $Q_{|x=0}$ are simple, solution in $\tilde{O}(nd) + \mathsf{R}_{\mathbb{K}}(n)$**



The case of simple roots

Series root's constant coefficient: root of $Q_{|x=0} \in \mathbb{K}[y]$
$(Q(f) = 0 \bmod x^d \quad \Rightarrow \quad Q_{|x=0}(f(0)) = 0)$

Simple roots and Newton iteration

α a simple root of $Q_{|x=0}$
$\rightsquigarrow$ one series root $f = \alpha + x\hat{f} \in \mathbb{K}[\![x]\!]$

computed via Newton iteration, $f \leftarrow f - \frac{Q(f)}{Q'(f)} \bmod x^{2^i}$ [Kung-Traub, 1978]

$\Rightarrow$ **if all roots of $Q_{|x=0}$ are simple, solution in $\tilde{O}(nd) + \mathsf{R}_{\mathbb{K}}(n)$**

Important case: if Q is over $\mathbb{K}[x]$ and we look for $\mathbb{K}[x]$ roots:

1. choose $a \in \mathbb{K}$ randomly (possibly in an extension of $\mathbb{K}$)
   $\rightsquigarrow$ with good probability, roots of $Q_{x=a}$ are simple
2. solve $Q(f) = 0 \bmod (x-a)^d$

General case: recursive structure

Consider a root α of $Q_{|x=0} \quad \Rightarrow Q(\alpha + xy)$ has x-valuation $\boldsymbol{v_\alpha \geqslant 1}$

For the **shift** $\boldsymbol{\hat{Q}_\alpha(y)} = x^{-v_\alpha} Q(\alpha + xy)$ and a series $f = \alpha + x\hat{f}$:

$$Q(f) = 0 \bmod x^d \iff \hat{Q}_\alpha(\hat{f}) = 0 \bmod x^{d-v_\alpha}$$

General case: recursive structure

Consider a root α of $Q_{|x=0} \quad \Rightarrow Q(\alpha + xy)$ has x-valuation $\boldsymbol{v_\alpha \geqslant 1}$

For the **shift** $\boldsymbol{\hat{Q}_\alpha(y)} = x^{-v_\alpha} Q(\alpha + xy)$ and a series $f = \alpha + x\hat{f}$:

$$Q(f) = 0 \bmod x^d \iff \hat{Q}_\alpha(\hat{f}) = 0 \bmod x^{d-v_\alpha}$$

Algorithm [Roth-Ruckenstein, 2000]

roots of $Q_{|x=0}$ ------> α β γ

General case: recursive structure

Consider a root α of $Q_{|x=0} \quad \Rightarrow Q(\alpha + xy)$ has x-valuation $\boldsymbol{v_\alpha \geqslant 1}$

For the **shift** $\boldsymbol{\hat{Q}_\alpha(y)} = x^{-v_\alpha} Q(\alpha + xy)$ and a series $f = \alpha + x\hat{f}$:

$$Q(f) = 0 \bmod x^d \iff \hat{Q}_\alpha(\hat{f}) = 0 \bmod x^{d - v_\alpha}$$

Algorithm [Roth-Ruckenstein, 2000]

shifts $(\hat{Q}_\alpha, \hat{Q}_\beta, \hat{Q}_\gamma)$

roots of $Q_{|x=0}$ ------> α β γ

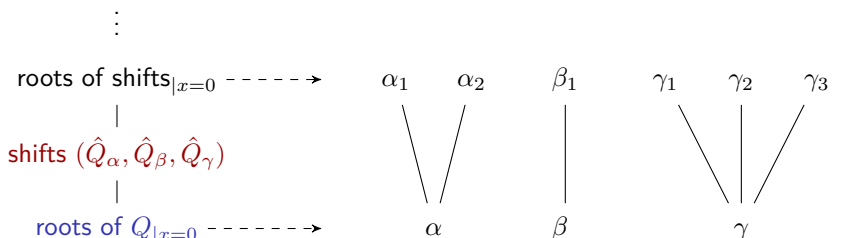
General case: recursive structure

Consider a root α of $Q_{|x=0}$ $\Rightarrow Q(\alpha + xy)$ has x-valuation $v_\alpha \geqslant 1$

For the **shift** $\hat{Q}_\alpha(y) = x^{-v_\alpha} Q(\alpha + xy)$ and a series $f = \alpha + x\hat{f}$:

$$Q(f) = 0 \bmod x^d \iff \hat{Q}_\alpha(\hat{f}) = 0 \bmod x^{d-v_\alpha}$$

Algorithm [Roth-Ruckenstein, 2000]



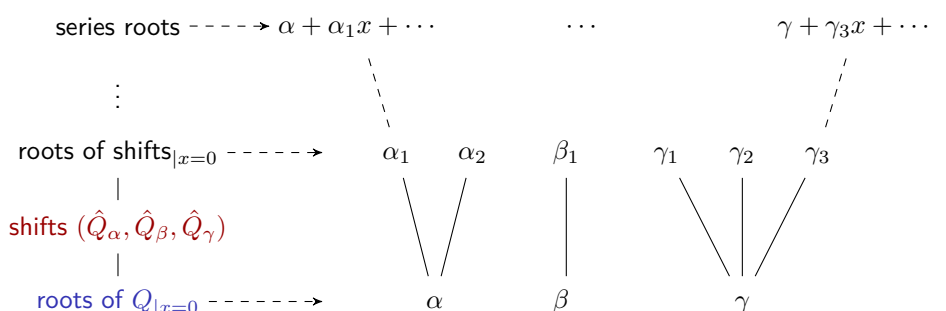
General case: recursive structure

Consider a root α of $Q_{|x=0} \quad \Rightarrow Q(\alpha + xy)$ has x-valuation $v_\alpha \geqslant 1$

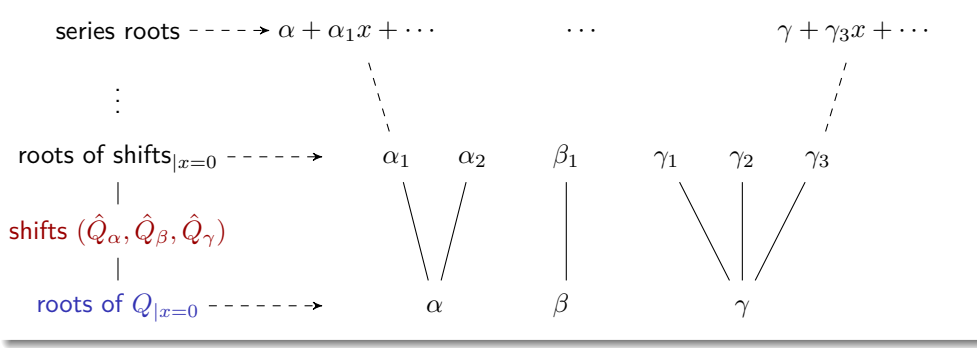
For the **shift** $\hat{Q}_\alpha(y) = x^{-v_\alpha} Q(\alpha + xy)$ and a series $f = \alpha + x\hat{f}$:

$$Q(f) = 0 \bmod x^d \iff \hat{Q}_\alpha(\hat{f}) = 0 \bmod x^{d - v_\alpha}$$

Algorithm [Roth-Ruckenstein, 2000]



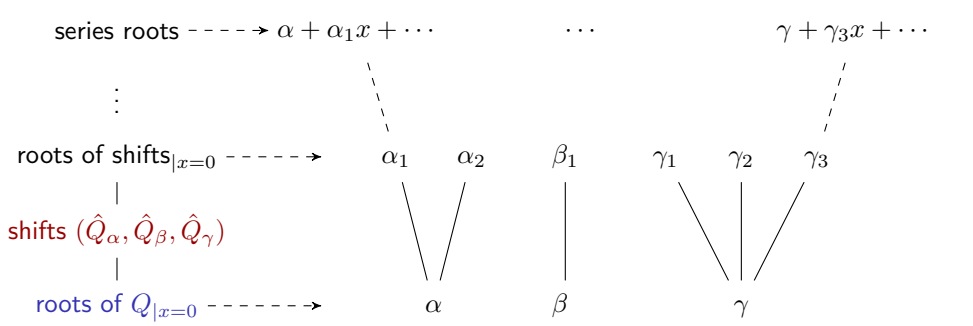
Representation of the roots



$$\mathrm{Roots}(Q, d) = \bigcup_{1 \leqslant i \leqslant \ell} (f_i + x^{t_i} \mathbb{K}[\![x]\!])$$

with $\deg(f_i) < t_i$ and $\ell \leqslant \deg(Q_{|x=0})$

Representation of the roots

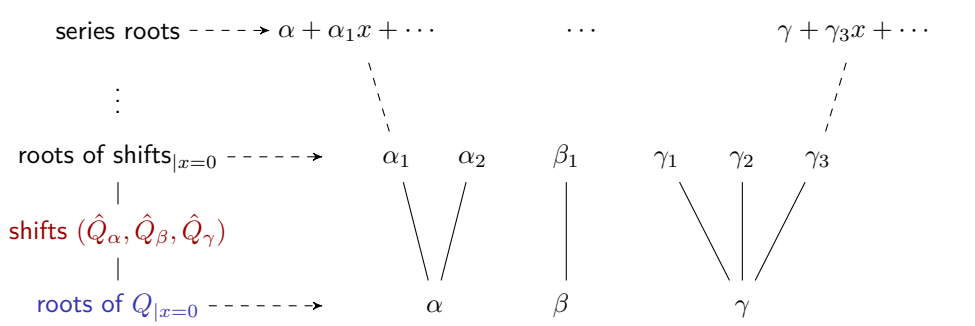


$$\mathrm{Roots}(Q,d) \;=\; \bigcup_{1\leqslant i\leqslant \boldsymbol{\ell}} (\boldsymbol{f_i} + x^{\boldsymbol{t_i}}\mathbb{K}[\![x]\!])$$

with $\deg(\boldsymbol{f_i}) < \boldsymbol{t_i}$ and $\boldsymbol{\ell} \leqslant \deg(Q_{|x=0})$

No explosion: $\deg(\hat{Q}_{\alpha|x=0}) + \deg(\hat{Q}_{\beta|x=0}) + \deg(\hat{Q}_{\gamma|x=0}) \leqslant \deg(Q_{|x=0})$

Representation of the roots



$$\mathrm{Roots}(Q,d) \;=\; \bigcup_{1\leqslant i\leqslant \boldsymbol{\ell}} (\boldsymbol{f_i} + x^{\boldsymbol{t_i}}\mathbb{K}[\![x]\!])$$

with $\deg(\boldsymbol{f_i}) < \boldsymbol{t_i}$ and $\boldsymbol{\ell} \leqslant \deg(Q_{|x=0})$

No explosion: $\deg(\hat{Q}_{\alpha|x=0}) + \deg(\hat{Q}_{\beta|x=0}) + \deg(\hat{Q}_{\gamma|x=0}) \leqslant \deg(Q_{|x=0})$

. . . but weak degree control: degree of shifts $\deg(\hat{Q}_\alpha)$ as large as $\deg(Q)$

Outline

- Overview of the contribution
- Structure of the set of roots
- Divide and conquer algorithm with degree control

Divide and conquer algorithm

Roth-Ruckenstein algorithm

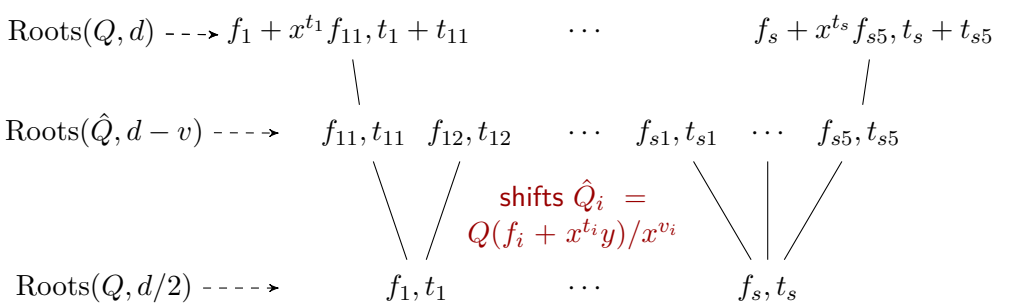
- $\leqslant d$ iterations at precision $\leqslant d$
- each dealing with $\leqslant n$ shifts of degree $\leqslant n$

cost $O(n^2d^2)$

Alekhnovich algorithm

- divide and conquer on d
- fast computation of the shifts

cost $O\tilde{}(n^2d)$



Degree control

Obstacle: at most n shifts $\hat{Q}_i$, each of degree n
$\rightsquigarrow$ cannot be computed in time quasi-linear in n

But we have $\sum_i \deg(\hat{Q}_{i|x=0}) \leqslant \deg(Q_{|x=0})$

Main ingredient

Compute $\boldsymbol{A_i} \in \mathbb{K}[\![x]\!][y]$ such that

- A_i has the same series roots as $\hat{Q}_i$
- $\deg(A_i) = \deg(\hat{Q}_{i|x=0})$

$\rightsquigarrow$ recursive calls with $(A_i)_i$ rather than $(\hat{Q}_i)_i$

Factorization: $\hat{Q}_i = A_i B_i$, with A_i monic and $B_{i|x=0}$ constant
[Musser's Algorithm Q, 1975]: given $\hat{Q}_i$, find A_i, B_i via Hensel lifting

we cannot compute $\hat{Q}_i$ or B_i $\rightsquigarrow$ we obtain $(A_i)_i$ directly from Q
degrees controlled via modular computations

Conclusion and perspectives

Contribution = fastest known algorithm

Cost bound: $\tilde{O}(nd) + O(d\,\mathsf{R}_{\mathbb{K}}(n))$

deterministic, $n = \deg_y(Q)$

cost of root-finding over $\mathbb{K}$

Perspectives

- computing **Puiseux series** roots of $Q(y)$
- computing roots of a polynomial $Q(y)$ over the p**-adic integers**
- . . . and over more general local domains?