Computing Riemann–Roch spaces via Puiseux expansions

S. Abelard, Elena Berardini, A. Couvreur and G. Lecerf

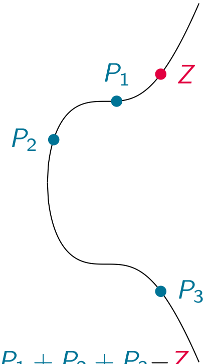
Laboratoire d'informatique de l'École polytechnique (LIX)
CNRS, École polytechnique, Institut Polytechnique de Paris

Part of a project funded by the Agence de l'Innovation de Défense

AGC^{2}T
4^{th} June 2021

Riemann–Roch problem

Divisor on a curve $\mathcal{C}$: $D = \sum_{P \in \mathcal{C}} n_P P$



D$=P_1 + P_2 + P_3 - Z$

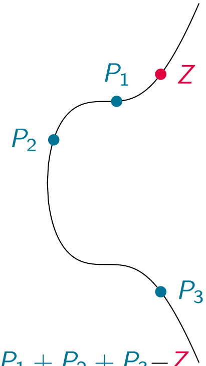
The **Riemann–Roch space** $L(D)$ is the space of all functions $\frac{G}{H} \in \mathbb{K}(\mathcal{C})$ s. t.:

- if $n_P < 0$ then P must be a zero of G (of multiplicity $\geqslant -n_P$)
- if $n_P > 0$ then P can be a zero of H (of multiplicity $\leqslant n_P$)
- G/H has not other poles outside the points P with $n_P > 0$

Here: Z must be a zero of G, the P_i's can be zeros of H

Riemann–Roch problem

Divisor on a curve $\mathcal{C}$: $D = \sum_{P \in \mathcal{C}} n_P P$



D=$P_1 + P_2 + P_3 - Z$

The **Riemann–Roch space** $L(D)$ is the space of all functions $\frac{G}{H} \in \mathbb{K}(\mathcal{C})$ s. t.:

- if $n_P < 0$ then P must be a zero of G (of multiplicity $\geqslant -n_P$)
- if $n_P > 0$ then P can be a zero of H (of multiplicity $\leqslant n_P$)
- G/H has not other poles outside the points P with $n_P > 0$

Here: Z must be a zero of G, the P_i's can be zeros of H

Riemann–Roch theorem $\rightsquigarrow$ dimension of $L(D)$

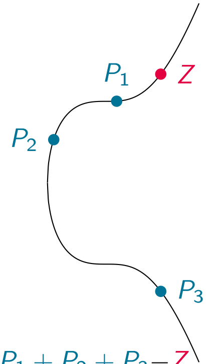
Riemann–Roch problem

Divisor on a curve $\mathcal{C}$: $D = \sum_{P \in \mathcal{C}} n_P P$

$D \geq 0$ if $n_P \geq 0$

$\deg(D) = \sum n_P \deg(P)$

$(H) = \sum \mathrm{ord}_P(H) P$



D$=P_1 + P_2 + P_3 - Z$

The **Riemann–Roch space** $L(D)$ is the space of all functions $\frac{G}{H} \in \mathbb{K}(\mathcal{C})$ s. t.:

- if $n_P < 0$ then P must be a zero of G (of multiplicity $\geqslant -n_P$)
- if $n_P > 0$ then P can be a zero of H (of multiplicity $\leqslant n_P$)
- G/H has not other poles outside the points P with $n_P > 0$

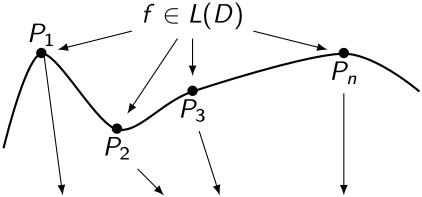
Here: Z must be a zero of G, the P_i's can be zeros of H

Riemann–Roch theorem $\rightsquigarrow$ dimension of $L(D)$

⚠ no explicit method to compute a basis of $L(D)$

Some motivation

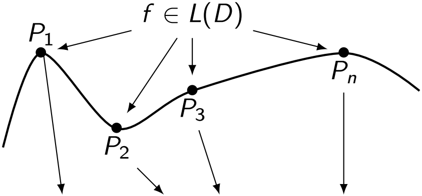
- Construction of algebraic geometry codes



$$\mathcal{C}((P_i)_i, D) := \{(f(P_1), f(P_2), f(P_3), \ldots, f(P_n)) \mid f \in L(D)\}$$

Some motivation

- Construction of algebraic geometry codes



$$\mathcal{C}((P_i)_i, D) := \{(f(P_1), f(P_2), f(P_3), \ldots, f(P_n)) \mid f \in L(D)\}$$

(Some) Recent applications of AG codes:

- Locally Recoverable Codes[1]
- Interactive Oracle Proofs[2]

[1] A. Barg, I. Tamo and S. Vladuts, *Locally recoverable codes on algebraic curves*, 2017

[2] S. Bordage, J. Nardi, *Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes*, 2021

Some motivation

- Construction of algebraic geometry codes
- Group operations on Jacobians of curves[1]
- Symbolic integration[2]
- Diophantine equations[3]

---

[1] K. Khuri-Makdisi, *Asymptotically fast group operations on Jacobians of general curves*, 2007

[2] J.H. Davenport, On the Integration of Algebraic Functions, 1981

[3] J. Coates, Construction of rational functions on a curve, 1970

Riemann-Roch problem: state of the art

Geometric methods:
(Brill–Noether theory ∼1874)

- Goppa, Le Brigand–Risler (80's)
- Huang–Ierardi (90's)
- Khuri-Makdisi (2007)
- Le Gluher–Spaenlehauer (2018)
- Abelard–Couvreur–Lecerf (2020)

Arithmetic methods:
(Ideals in function fields)

- Hensel–Landberg (1902)
- Coates (1970)
- Davenport (1981)
- Hess (2001)

Riemann-Roch problem: state of the art

Geometric methods:
(Brill–Noether theory ∼1874)
- Goppa, Le Brigand–Risler (80's)
- Huang–Ierardi (90's)
- Khuri-Makdisi (2007)
- Le Gluher–Spaenlehauer (2018)
- Abelard–Couvreur–Lecerf (2020)

Arithmetic methods:
(Ideals in function fields)
- Hensel–Landberg (1902)
- Coates (1970)
- Davenport (1981)
- Hess (2001)

Nodal/ordinary curves: Las Vegas algorithm computing $L(D)$ in $\tilde{\mathcal{O}}((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}})$ field operations[4]

Non-ordinary curves: ⚠ no explicit complexity exponent

[4] here $2 \leqslant \omega \leqslant 3$ is a feasible exponent for linear algebra ($\omega = 2.373$)

Today's menu[5]

Brill–Noether method	$\rightsquigarrow$	necessary and sufficient conditions on H and G such that $G/H \in L(D)$
Puiseux series	$\rightsquigarrow$	handling singular points on the curve $\mathcal{C}$
(Structured) Linear algebra	$\rightsquigarrow$	computing H and G in practice

Main course

Las Vegas algorithm computing $L(D)$ in $\tilde{\mathcal{O}}((\delta^2 + \deg D_+)^\omega)$ field operations.

[5]Sorry, Bouillabaisse is out of stock today!

Brill–Noether in a nutshell

Input

$\mathcal{C} : F(X, Y, Z) = 0$ a plane projective curve,
$D = D_+ - D_-$ a smooth divisor with D_+ and D_- effective.

Description of $L(D)$: non-zero elements are of the form $\frac{G_i}{H}$ where

- H satisfies $(H) \geqslant D_+$
- H passes through all the singular points of $\mathcal{C}$ with ad hoc multiplicities
- $\deg G_i = \deg H$, G_i coprime with F and $(G_i) \geqslant (H) - D$

Brill–Noether in a nutshell

Input

$\mathcal{C} : F(X, Y, Z) = 0$ a plane projective curve,
$D = D_+ - D_-$ a smooth divisor with D_+ and D_- effective.

Description of $L(D)$: non-zero elements are of the form $\frac{G_i}{H}$ where

- H satisfies $(H) \geqslant D_+$
- H passes through all the singular points of $\mathcal{C}$ with ad hoc multiplicities
- $\deg G_i = \deg H$, G_i coprime with F and $(G_i) \geqslant (H) - D$

How do we handle singular points?

Brill–Noether in a nutshell

Input

$\mathcal{C} : F(X, Y, Z) = 0$ a plane projective curve,
$D = D_+ - D_-$ a smooth divisor with D_+ and D_- effective.

Description of $L(D)$: non-zero elements are of the form $\frac{G_i}{H}$ where

- H satisfies $(H) \geqslant D_+$
- H satisfies $(H) \geqslant \mathcal{A}$ (we say that "H is adjoint to the curve")
- $\deg G_i = \deg H$, G_i coprime with F and $(G_i) \geqslant (H) - D$

How do we handle singular points?

$\rightsquigarrow$ the adjunction divisor $\mathcal{A}$ "encodes" the singular points of $\mathcal{C}$ with their multiplicities

Brill–Noether in a nutshell

Input

$\mathcal{C} : F(X,Y,Z) = 0$ a plane projective curve,
$D = D_+ - D_-$ a smooth divisor with D_+ and D_- effective.

Description of $L(D)$: non-zero elements are of the form $\frac{G_i}{H}$ where

- H satisfies $(H) \geqslant D_+$
- H satisfies $(H) \geqslant \mathcal{A}$
- $\deg G_i = \deg H$, G_i coprime with F and $(G_i) \geqslant (H) - D$

How do we handle singular points?

$\rightsquigarrow$ the adjunction divisor $\mathcal{A}$ "encodes" the singular points of $\mathcal{C}$ with their multiplicities

How do we handle divisors?

Brill–Noether in a nutshell

Input

$\mathcal{C} : F(X, Y, Z) = 0$ a plane projective curve,
$D = D_+ - D_-$ a smooth divisor with D_+ and D_- effective.

Description of $L(D)$: non-zero elements are of the form $\frac{G_i}{H}$ where

- H satisfies $(H) \geqslant D_+$
- H satisfies $(H) \geqslant \mathcal{A}$
- $\deg G_i = \deg H$, G_i coprime with F and $(G_i) \geqslant (H) - D$

How do we handle singular points?

$\rightsquigarrow$ the adjunction divisor $\mathcal{A}$ "encodes" the singular points of $\mathcal{C}$ with their multiplicities

How do we handle divisors?

series expansions of multi-set representations $((P_i)_i, m_i)$	$\rightsquigarrow$	routines on divisors have negligible cost

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$

Step 2: Compute a common denominator H

Step 3: Compute $(H) - D$

Step 4: Compute numerators G_i (similar to Step 2)

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$

Step 2: Compute a common denominator H

Step 3: Compute $(H) - D$

Step 4: Compute numerators G_i (similar to Step 2)

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$

Step 2: Compute a common denominator H

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i (similar to Step 2)

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$

Step 2: Compute a common denominator H

Step 3: $\checkmark \leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i (similar to Step 2)

Puiseux expansions

⚠work only in characteristic 0 or "big" characteristic[6]

Let $F \in \mathbb{K}[x, y]$ be absolutely irreducible, monic in y and of degree d_y in y. The roots of $F \in \mathbb{K}((x))[y]$ in $\cup_{e\geqslant 1}\overline{\mathbb{K}}((x^{1/e}))$ are its Puiseux expansions $\varphi_0, \ldots, \varphi_{d_y-1}$, so that F writes

$$F = \prod_{i=1}^{d_y-1} (y - \varphi_i).$$

Here $\varphi_i = \sum_{j=n}^{\infty} \beta_{i,j} x^{j/e_i}$, where e_i is taken to be as small as possible.

Toy example: $F = y^2 - x^3 \rightsquigarrow F = (y - x^{3/2})(y + x^{3/2})$

Let $\varphi_0 = \sum_{j=1}^{\infty} \beta_j x^{j/e_0}$ and ζ a primitive e_0-th root of unity. Then for $0 \leqslant k < e_0$

$$\sum_{j=n}^{\infty} \beta_j (\zeta^k x^{1/e_0})^j$$

are (pairwise distinct) Puiseux expansions of F. They are all equivalent...

[6]We will come back to this later...

Rational Puiseux expansions

For $k = 0, \ldots, e_0 - 1$ the e_0 Puiseux series in $\overline{\mathbb{K}}((x^{1/e_0}))$

$$\varphi_k(x) = \sum_{j=n}^{\infty} \beta_j (\zeta^k (x)^{1/e_0})^j$$

are all represented by a rational Puiseux expansion:

Definition

A rational Puiseux expansion of an absolutely irreducible polynomial $G \in \mathbb{E}((x))[y]$ is a pair $(X(t), Y(t)) \in \mathbb{E}((t))^2$ such that

- $(X(t), Y(t)) = (\gamma t^e, \sum_{j=n}^{\infty} \beta_j t^j)$ *with* $\gamma \beta_n \neq 0$
- $G(X(t), Y(t)) = 0$

Toy example: $F = y^2 - x^3 \rightsquigarrow F = (y - x^{3/2})(y + x^{3/2}) \rightsquigarrow (t^2, t^3)$

Rational Puiseux expansions of F correspond bijectively
to the places of the curve $F(x, y) = 0$

The adjoint condition

The local adjoint divisor is

$$\mathcal{A}_P = -\sum_{\mathcal{P}|P} \mathrm{val}_{\mathcal{P}} \left(\frac{dx}{F_y} \right) \mathcal{P}$$

Places $\iff$ RPE $(X(t), Y(t))$ and t is a uniformizing parameter

$$\rightsquigarrow \mathrm{val}_{\mathcal{P}} \left(\frac{dx}{F_y} \right) = \mathrm{val}_t \left(\frac{et^{e-\mathbf{1}}}{F_y(X(t),Y(t),1)} \right)$$

The adjoint condition

The local adjoint divisor is

$$\mathcal{A}_P = -\sum_{\mathcal{P}|P} \mathrm{val}_{\mathcal{P}}\left(\frac{dx}{F_y}\right)\mathcal{P}$$

Places $\iff$ RPE $(X(t), Y(t))$ and t is a uniformizing parameter

$$\rightsquigarrow \mathrm{val}_{\mathcal{P}}\left(\frac{dx}{F_y}\right) = \mathrm{val}_t\left(\frac{et^{e-1}}{F_y(X(t),Y(t),1)}\right)$$

Example

Consider $\mathcal{C} : y^2 - x^3 = 0$ *in the affine chart* $z = 1$.
$(0,0)$ *is the (only, non-ordinary) singular point.*
Puiseux series : $y = \pm x^{3/2}$
RPE: $(X(t), Y(t)) = (t^2, t^3) \rightsquigarrow$ *(unique) place* $\mathcal{P}$

$$\mathrm{val}_{\mathcal{P}}\left(\frac{dx}{F_y}\right) = \mathrm{val}_t\left(\frac{2t}{2t^3}\right) = -2$$

The adjoint condition

The local adjoint divisor is

$$\mathcal{A}_P = -\sum_{\mathcal{P}|P} \mathrm{val}_{\mathcal{P}} \left(\frac{dx}{F_y}\right) \mathcal{P}$$

Places $\iff$ RPE $(X(t), Y(t))$ and t is a uniformizing parameter

$$\rightsquigarrow \mathrm{val}_{\mathcal{P}} \left(\frac{dx}{F_y}\right) = \mathrm{val}_t \left(\frac{et^{e-1}}{F_y(X(t), Y(t), 1)}\right)$$

Computation:

Fast algorithms for Puiseux series expansions of germs of curves[7]

$\rightsquigarrow \mathcal{A}$ computed with an expected number of $\tilde{O}(\delta^3)$ field operations

[7] A. Poteaux and M. Weimann, *Computing Puiseux series: a fast divide and conquer algorithm*, 2021

Finding a denominator in practice

Straightforward linear solving

Let $d = \deg H$.

$$\text{Condition } (H) \geqslant \mathcal{A} + D_+$$

$\rightsquigarrow$ linear system with $\deg \mathcal{A} + \deg D_+$ equations

$\rightsquigarrow$ Gaussian elimination costs

$$\tilde{O}((d\delta + \delta^2 + \deg D_+)^\omega)$$

Finding a denominator in practice

Straightforward linear solving

Let $d = \deg H$.

Condition $(H) \geqslant \mathcal{A} + D_+$

$\rightsquigarrow$ linear system with $\deg \mathcal{A} + \deg D_+$ equations

$\rightsquigarrow$ Gaussian elimination costs

$$\tilde{O}((d\delta + \delta^2 + \deg D_+)^\omega)$$

How big is d?

Finding a denominator in practice

Straightforward linear solving

Let $d = \deg H$.

Condition $(H) \geqslant \mathcal{A} + D_+$

$\rightsquigarrow$ linear system with $\deg \mathcal{A} + \deg D_+$ equations

$\rightsquigarrow$ Gaussian elimination costs

$$\tilde{O}((d\delta + \delta^2 + \deg D_+)^\omega)$$

How big is d?

We proved that $d = \left\lceil \frac{(\delta-1)(\delta-2)+\deg D_+}{\delta} \right\rceil$ is enough

Finding a denominator in practice

Straightforward linear solving

Let $d = \deg H$.

Condition $(H) \geqslant \mathcal{A} + D_+$

$\rightsquigarrow$ linear system with $\deg \mathcal{A} + \deg D_+$ equations

$\rightsquigarrow$ Gaussian elimination costs

$$\tilde{O}((d\delta + \delta^2 + \deg D_+)^\omega)$$

How big is d?

We proved that $d = \left\lceil \frac{(\delta-1)(\delta-2)+\deg D_+}{\delta} \right\rceil$ is enough

$\rightsquigarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$ field operations

Second method: structured linear algebra

$$\mathrm{val}_t(H(X(t), Y(t), 1) \geqslant \mathrm{val}_t\left(\frac{et^{e-1}}{F_y(X(t), Y(t), 1)}\right)$$

$\rightsquigarrow$ space of polynomials $H(x, y)$ satisfying these conditions is a $\mathbb{K}[x]$-module

$\rightsquigarrow$ computing a basis[8] costs $\tilde{O}((\delta^2 + \deg D_+)^\omega)$

[8] C.-P. Jeannerod, V. Neiger, É. Schost and G. Villard, *Computing minimal interpolation bases*, 2017

Second method: structured linear algebra

$$\mathrm{val}_t(H(X(t), Y(t), 1) \geqslant \mathrm{val}_t\left(\frac{et^{e-1}}{F_y(X(t), Y(t), 1)}\right)$$

$\rightsquigarrow$ space of polynomials $H(x, y)$ satisfying these conditions is a $\mathbb{K}[x]$-module

$\rightsquigarrow$ computing a basis[8] costs $\tilde{O}((\delta^2 + \deg D_+)^\omega)$

Same complexity exponent but...

Benefits:

- bases with smaller representation size in general
- better complexity bound for algebraically closed fields
- possibility of future improvements

[8] C.-P. Jeannerod, V. Neiger, É. Schost and G. Villard, *Computing minimal interpolation bases*, 2017

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$ ✓ $\leftarrow \tilde{O}(\delta^3)$

Step 2: Compute H ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i (similar to Step 2)

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$ ✓ $\leftarrow \tilde{O}(\delta^3)$

Step 2: Compute H ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i (similar to Step 2)

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$ ✓ $\leftarrow \tilde{O}(\delta^3)$

Step 2: Compute H ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D
Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$ ✓ $\leftarrow \tilde{O}(\delta^3)$

Step 2: Compute H ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Return: a basis of $L(D)$ in terms of H and the G_i!

Sketch of the algorithm

Input: a plane curve $\mathcal{C}$ of degree δ and a smooth divisor D

Output: a basis of $L(D)$

Step 1: Compute the adjoint divisor $\mathcal{A}$ ✓ $\leftarrow \tilde{O}(\delta^3)$

Step 2: Compute H ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Step 3: Compute $(H) - D$ ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^2)$

Step 4: Compute numerators G_i ✓ $\leftarrow \tilde{O}((\delta^2 + \deg D_+)^\omega)$

Return: a basis of $L(D)$ in terms of H and the G_i!

Main complexity bound

Las Vegas algorithm computing $L(D)$ in $\tilde{\mathcal{O}}((\delta^2 + \deg D_+)^\omega)$ field operations[9].

[9] S. Abelard, E. Berardini, A. Couvreur et G. Lecerf, preprint coming soon!

What's next?

1. Computing Riemann–Roch spaces of non-ordinary curves in "small" positive characteristic (in progress with G. Lecerf)
2. Improving the complexity in the non-ordinary case ($\rightsquigarrow$ sub-quadratic?)
3. Implementation including fast structured linear algebra
4. Computing Riemann-Roch spaces of surfaces

Thank you for your attention!

Questions?

berardini@lix.polytechnique.fr

My first AGC^{2}T

My second AGC^{2}T

My third AGC^{2}T

zoom

My next AGC^{2}T

...on Mars (eille)?