A Geometric Approach for the Computation of Riemann-Roch Spaces : Algorithm and Complexity

Aude Le Gluher and Pierre-Jean Spaenlehauer
Université de Lorraine / INRIA Nancy – Grand Est / CNRS
CARAMBA team

JNCF, Luminy, 2019

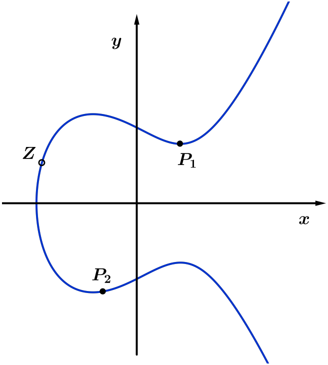
Riemman-Roch Problem

$\mathbf{K}$: perfect field of characteristic sufficiently large or zero.

C : irreducible projective curve described by $Q \in \mathbf{K}[X, Y]$, not necessarily smooth.

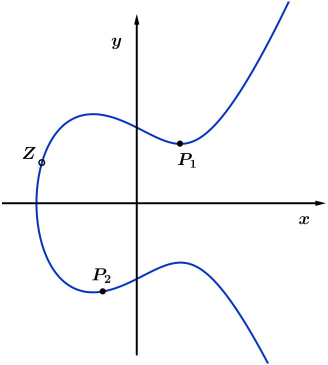
K : perfect field of characteristic sufficiently large or zero.

C : irreducible projective curve described by $Q \in \mathbf{K}[X, Y]$, not necessarily smooth.



$\mathbf{K}$: perfect field of characteristic sufficiently large or zero.

C : irreducible projective curve described by $Q \in \mathbf{K}[X, Y]$, not necessarily smooth.



Goal : find all functions $R(X,Y)/S(X,Y) \in \mathbf{K}(C) = \mathrm{Frac}(\mathbf{K}[X,Y]/(Q))$ such that :

$$\begin{cases} R(Z) = 0 \\ S \text{ may cancel at } P_1 \\ S \text{ may cancel at } P_2 \\ \text{ no poles at infinity} \end{cases}$$

$\boxed{\text{Prescribed zeroes, authorized poles}}$

Motivations

Riemann-Roch spaces are vector spaces useful in particular for :

- Computing the group law of the Jacobian of a curve.
  Volcheck (1994), Huang et Ierardi (1994), Khuri-Makdisi (1995).
- Building algebraic geometric error-correcting codes.
  Goppa (1983), Haché (1995).
- Integration of algebraic functions. Davenport (1981).

State of the art

Here, C is a curve of degree d and genus g and $D = D_+ - D_-$ is a divisor on C (D_+ and D_- are effective divisors).

Computation of general Riemann-Roch spaces :

- Huang and Ierardi (1994) : geometric algorithm in $O(d^6 \deg(D_+)^6)$.
- Haché (1995).
- Hess's arithmetic algorithm (2002).

Computation of the group law in Jacobians ($\deg(D_+) = O(g)$) :

- Volcheck (1994) : arithmetic algorithm in $O(\max(d, g)^7)$.
- Khuri-Makdisi (2007) : algorithm in $O(g^{\omega+\epsilon})$ where ω is a feasible exponent for matrix multiplication and $\epsilon > 0$.
- Possible improvements for specific curves (for instance $\widetilde{O}(g)$ for hyperelliptic curves, Cantor).

Main results

- Variant of the Brill-Noether algorithm : geometric probabilistic Las Vegas algorithm for computing Riemann-Roch spaces in the case of divisors not involving singular points. Mild assumptions when the curve is singular.
- Bound on the probability of failure :

  $$O(\max(\deg(C)^4, \deg(D_+)^2)/|E|)$$

  where E is a finite subset of $\mathbf{K}$ in which we can pick elements at random uniformly.
- Proof of complexity :
  Number of arithmetic operations in $\mathbf{K}$ bounded by :

  $$O(\max(\deg(C)^2, \deg(D_+))^\omega)$$

  where ω is a feasible exponent for matrix multiplication.
- C++/NTL implementation of this algorithm.

Plan

1. Algorithm
2. Representation of divisors
3. Complexity

Input and output

Input :

- A polynomial $q \in \mathbf{K}[X, Y]$ describing an irreducible projective plane curve C.
- The representations of two effective divisors D_+ and D_- both not involving any singular point of C.

Output : A basis of the vector space $L(D)$ where $D = D_+ - D_-$.

Remark

More on the representation of effective divisors later.

Construction of a suitable denominator

Common denominator of degree d.

$\longrightarrow$ Choice of a random polynomial h of degree d which vanishes with the right multiplicities at all points prescribed by D_+ : h is solution of an underdetermined linear system.

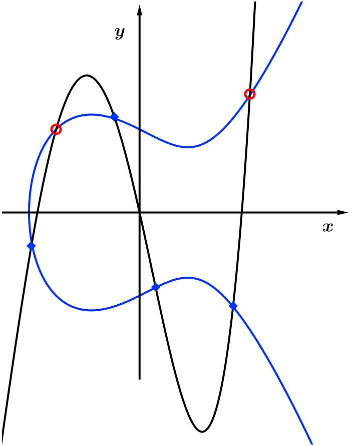
$\longrightarrow$ Computation of a representation for the effective divisor (h).

Remark

- We build h such that its degree in Y is lesser than $\deg(C)$.
- The degree d is tuned to be as small as possible while guaranteeing an underdetermined linear system. We have :

$$d < \frac{\deg(D_+)}{\deg(C)} + \deg(C)$$

Readjusting the zeroes



Non exact interpolation : h has non desired zeroes.

$\longrightarrow$ Find those non desired zeroes : they are represented by $(h) - D_+$.

$\longrightarrow$ Add them to D_-.

Counterbalance the unwanted zeros of the denominator by the same zeros for the numerators.

Remark

We assume (h) does not involve any singular point of C.

Construction of the numerators

From last step : $D' = D_- + (h) - D_+$ imposes the zeros of numerators.

$\longrightarrow$ Computation of a base B of polynomials of degree at most $\deg(h)$ and vanishing at all points prescribed by D' with the right multiplicities : again a linear system.

Correction

The set $\{b/h \mid b \in B\}$ is a base of the Riemann-Roch space $L(D)$.

Proof : $\mathrm{Vect}(\{b/h \mid b \in B\}) \subset L(D)$ by construction. For the converse : use a variant of Brill-Noether residue theorem.

Sum up of the algorithm

- Choose an interpolating polynomial h as denominator.
- Compute the representation of (h).
- Identify the unwanted zeros of h.
- Find the new constraints on the zeroes of numerators.
- Compute a base of numerators.

Plan

1. Algorithm
2. Representation of divisors
   - Polynomial representation
   - Operations on divisors
3. Complexity

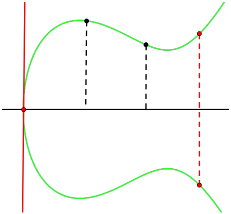
Main idea

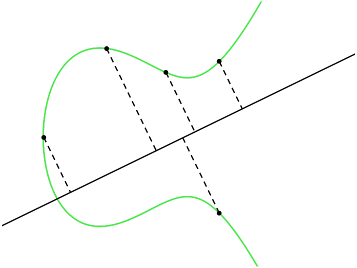
Remark

We only represent effective divisors D with no singular points.

The representation of D is :

- Similar to Mumford Coordinates in the case of hyperelliptic curves,
- Encodes the effective divisor by univariate polynomials (Giusti, Lecerf, Salvy, 1999). In particular :
- Finds a univariate polynomial χ such that $\mathbf{K}[C]/(I) \cong \mathbf{K}[S]/\chi(S)$ where I is an ideal such that $\mathbf{K}[C]/(I)$ is the description of the algebraic set corresponding to the support of D.





Potential problems :

- Points of the divisor with the same projection.
- Tangents to the curve perpendicular to the direction of projection at some divisor points.

Solution : Find a suitable direction of projection.

Primitive representation of effective divisors

An effective divisor D is represented by $(\lambda, \chi, u, v) \in \mathbf{K} \times \mathbf{K}[S]^3$ such that :

1. The degree of χ is the degree of D and $\deg(u), \deg(v) < \deg(D)$.
2. $q(u(S), v(S)) \equiv 0 \mod \chi(S)$.
3. $\lambda u(S) + v(S) = S$.
4. $GCD\left(\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S)), \chi(S)\right) = 1$.

Existence of the representation

Warning

Such a representation does not always exist !
BUT
It does exist if the field $\mathbf{K}$ is large enough.

Idea of the proof :

- If $\mathbf{K}$ is large enough, there is a $\lambda \in \mathbf{K}$ such that $\lambda X + Y$ is a primitive element of $\mathbf{K}[C]/(m)$.
- Build representations for each point P involved.
- Lift those representations to encode multiplicities (Hensel's lemma).
- Use the CRT to find the final representation.

Operations needed on representations

Our algorithm requires us to know how to :

- Sum two representations.
- Subtract two representations (knowing that the result will remain an effective divisor).
- Compute the representation of the divisor (h).

Remark

The first two operations first require the two input representations to agree on a common λ. Need to change the primitive element (Giusti, Lecerf, Salvy, 1999).

Example : the subtraction

Input : Two representations $(\lambda, \chi_1, u_1, v_1)$ and $(\lambda, \chi_2, u_2, v_2)$ of effective divisors D_1 and D_2.

Output : The representation of $D_1 - D_2$ if this divisor remains effective.

Algorithm :

- Suppress the common factors of χ_1 and χ_2 by computing $\chi = \chi_1 / GCD(\chi_1, \chi_2)$
- Reduce u_1 and v_1 modulo χ.
- Return (λ, χ, u, v).

Main idea

With this representation, operations on divisors are operations on polynomials !

Plan

Translation of the operations needed

- Choose polynomial h as denominator : build + solve linear system.
- Compute the representation of (h) : resultant and subresultant.
- Identify the unwanted zeros of h : GCD.
- Find the new constraints on the zeroes of numerators : CRT.
- Compute a base of numerators : build + solve linear system.

Remark

The cost of linear algebra dominates the costs of the others steps.
Confirmed in practice.

Final complexity and comparisons

Final complexity

Our algorithm requires at most

$$O(\max(\deg(C)^2, \deg(D_+))^\omega)$$

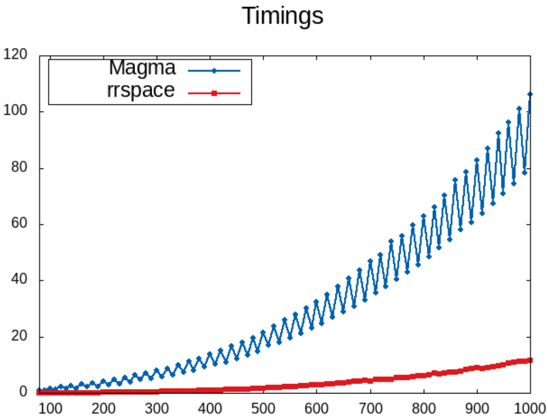
arithmetic operations in $\mathbf{K}$.

- Improves the complexity in $O(\deg(C)^6 \deg(D_+)^6)$ of the geometric algorithm of Huang and Ierardi.
- When $\deg(D_+) \leq \deg(C)^2$, complexity in $O(\deg(C)^{2\omega})$. Slightly improves Khuri-Makdisi in the special case of computing in Jacobians of smooth plane curves.

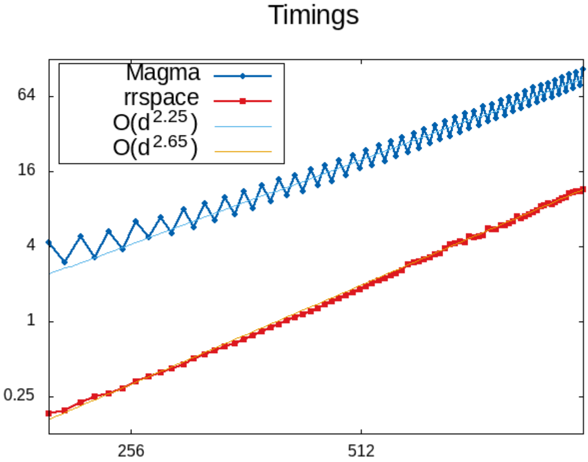
Experimental results

- Comparison of the C++/NTL implementation rrspace and the Magma implementation RiemannRochSpace.
- Experiments done with $\mathbf{K} = \mathrm{GF}(65521)$.

Time needed to compute the Riemann-Roch space of an effective divisor of increasing degree on a curve of degree 10.



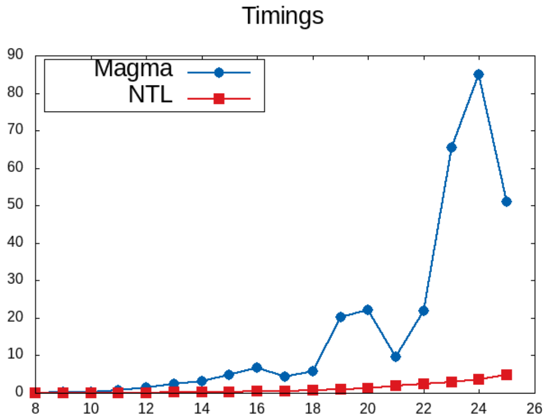
Logarithmic scales.



Experimental results

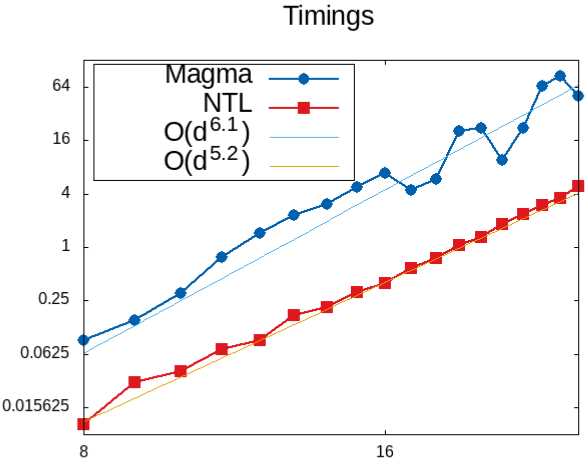
- Comparison of the C++/NTL implementation rrspace and the Magma implementation RiemannRochSpace.
- Experiments done with $\mathbf{K} = GF(65521)$.

Time needed to compute the sum of two elements of the Jacobian of a curve of increasing degree.



Experimental results

Logarithmic scales.



Future works

- Structure of the linear systems ?
- What happens when we cannot avoid singularities ?
  $\rightarrow$ Local desingularisation (Haché, 1995).

Code available : https ://gitlab.inria.fr/pspaenle/rrspace
ArXiv link : https ://arxiv.org/abs/1811.08237

Future works

- Structure of the linear systems ?
- What happens when we cannot avoid singularities ?
  $\rightarrow$ Local desingularisation (Haché, 1995).

Code available : https ://gitlab.inria.fr/pspaenle/rrspace
ArXiv link : https ://arxiv.org/abs/1811.08237

Thank you !