Computing period matrices of algebraic curves

Christian Neurohr
University of Oldenburg

Computeralgebra-Tagung Kassel
05.05.2017

Algebraic curves & Riemann surfaces

Let X be a smooth irreducible projective curve of genus g defined by a homogeneous polynomial F defined over $\mathbb{C}$, i.e.

$$X : F(x, y, z) = 0 \,.$$

Algebraic curves & Riemann surfaces

Let X be a smooth irreducible projective curve of genus g defined by a homogeneous polynomial F defined over $\mathbb{C}$, i.e.

$$X : F(x, y, z) = 0\,.$$

Over the complex numbers, $X = X(\mathbb{C})$ has the structure of a compact *Riemann surface*, that is, a 1-dimensional complex manifold together with a complex structure.

Algebraic curves & Riemann surfaces

Let X be a smooth irreducible projective curve of genus g defined by a homogeneous polynomial F defined over $\mathbb{C}$, i.e.

$$X : F(x, y, z) = 0\,.$$

Over the complex numbers, $X = X(\mathbb{C})$ has the structure of a compact *Riemann surface*, that is, a 1-dimensional complex manifold together with a complex structure.

In fact, there is a triple category equivalence between

- compact Riemann surfaces
- smooth irreducible projective curves over $\mathbb{C}$
- algebraic function fields in one variable over $\mathbb{C}$

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x, y) = F(x, y, 1) = 0$$

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x, y) = F(x, y, 1) = 0$$

and a smooth branched covering of the complex projective line,

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x, y) = F(x, y, 1) = 0$$

and a smooth branched covering of the complex projective line, e.g.

$$\varphi : X \to \mathbb{P}^1(\mathbb{C}),\ [x : y : z] \mapsto [x : z]$$

which is a morphism of degree $m = \deg_y f$

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x, y) = F(x, y, 1) = 0$$

and a smooth branched covering of the complex projective line, e.g.

$$\varphi : X \to \mathbb{P}^1(\mathbb{C}),\ [x : y : z] \mapsto [x : z]$$

which is a morphism of degree $m = \deg_y f$ with branch points

$$\mathcal{B} = \{[x : z] \in P^1(\mathbb{C}) \mid \#\varphi^{-1}([x : z]) < m\}.$$

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x, y) = F(x, y, 1) = 0$$

and a smooth branched covering of the complex projective line, e.g.

$$\varphi : X \to \mathbb{P}^1(\mathbb{C}),\ [x : y : z] \mapsto [x : z]$$

which is a morphism of degree $m = \deg_y f$ with branch points

$$\mathcal{B} = \{[x : z] \in P^1(\mathbb{C}) \mid \#\varphi^{-1}([x : z]) < m\}.$$

The corresponding function field is $\mathbb{C}(X) = \mathrm{Quot}(\mathbb{C}[x, y]/f(x, y))$.

Notation & choices

In order to do computations we fix an affine model for X, e.g.

$$f(x,y) = F(x,y,1) = 0$$

and a smooth branched covering of the complex projective line, e.g.

$$\varphi : X \to \mathbb{P}^1(\mathbb{C}),\ [x:y:z] \mapsto [x:z]$$

which is a morphism of degree $m = \deg_y f$ with branch points

$$\mathcal{B} = \{[x:z] \in P^1(\mathbb{C}) \mid \#\varphi^{-1}([x:z]) < m\}.$$

The corresponding function field is $\mathbb{C}(X) = \mathrm{Quot}(\mathbb{C}[x,y]/f(x,y))$.
After these choices we view X as an m-sheeted Riemann surface and $\mathbb{C}(X)$ as an algebraic extension of $\mathbb{C}(x)$ of degree m.

Holomorphic differentials

The space of holomorphic differentials of X is a g-dimensional $\mathbb{C}$-vector space that can be described as

$$\Omega^1(X) = \{h\mathrm{d}x : h \in \mathbb{C}(X) \mid \mathrm{ord}_P(h\mathrm{d}x) \geq 0 \text{ for all } P \in X\}.$$

Holomorphic differentials

The space of holomorphic differentials of X is a g-dimensional $\mathbb{C}$-vector space that can be described as

$$\Omega^1(X) = \{h\mathrm{d}x : h \in \mathbb{C}(X) \mid \mathrm{ord}_P(h\mathrm{d}x) \geq 0 \text{ for all } P \in X\}.$$

Computing period matrices, we are primarily interested in a basis $\bar{\omega} = (\omega_1, \ldots, \omega_g)$ of $\Omega_1(X)$,

Holomorphic differentials

The space of holomorphic differentials of X is a g-dimensional $\mathbb{C}$-vector space that can be described as

$$\Omega^1(X) = \{h\mathrm{d}x : h \in \mathbb{C}(X) \mid \mathrm{ord}_P(h\mathrm{d}x) \geq 0 \text{ for all } P \in X\}.$$

Computing period matrices, we are primarily interested in a basis $\bar{\omega} = (\omega_1, \ldots, \omega_g)$ of $\Omega_1(X)$, e.g.

$$\frac{1}{y}\mathrm{d}x, \frac{x}{y}\mathrm{d}x, \ldots, \frac{x^{g-1}}{y}\mathrm{d}x \quad \text{if } F \text{ is a hyperelliptic curve}$$

Holomorphic differentials

The space of holomorphic differentials of X is a g-dimensional $\mathbb{C}$-vector space that can be described as

$$\Omega^1(X) = \{h\mathrm{d}x : h \in \mathbb{C}(X) \mid \mathrm{ord}_P(h\mathrm{d}x) \geq 0 \text{ for all } P \in X\}.$$

Computing period matrices, we are primarily interested in a basis $\bar{\omega} = (\omega_1, \ldots, \omega_g)$ of $\Omega_1(X)$, e.g.

$$\frac{1}{y}\mathrm{d}x, \frac{x}{y}\mathrm{d}x, \ldots, \frac{x^{g-1}}{y}\mathrm{d}x \quad \text{if } F \text{ is a hyperelliptic curve}$$

$$\text{or} \quad \frac{1}{\partial_y f}\mathrm{d}x, \frac{x}{\partial_y f}\mathrm{d}x, \frac{y}{\partial_y f}\mathrm{d}x \quad \text{if } F \text{ is a smooth plane quartic.}$$

Homology group

For $P \in X$, the fundamental group $G := \pi_1(X, P)$ of X consists of the homotopy classes of closed paths on X based at P.

Homology group

For $P \in X$, the fundamental group $G := \pi_1(X, P)$ of X consists of the homotopy classes of closed paths on X based at P.
The *first homology group* of X, defined as

$$H_1(X, \mathbb{Z}) = G/[G, G],$$

is a free abelian group of rank $2g$.

Homology group

For $P \in X$, the fundamental group $G := \pi_1(X, P)$ of X consists of the homotopy classes of closed paths on X based at P.
The *first homology group* of X, defined as

$$H_1(X, \mathbb{Z}) = G/[G, G],$$

is a free abelian group of rank $2g$. A basis $(\alpha_i,\, \beta_j)$ $(1 \leq i, j \leq g)$ of $H_1(X, \mathbb{Z})$ is called *canonical* if the intersections satsify

$$\alpha_i \circ \beta_j = \delta_{ij} \quad \text{and} \quad \alpha_i \circ \alpha_j = \beta_i \circ \beta_j = 0$$

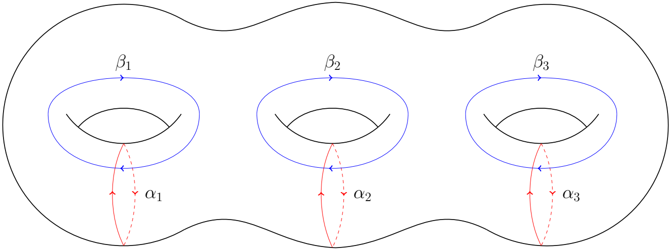
Homology group

For $P \in X$, the fundamental group $G := \pi_1(X, P)$ of X consists of the homotopy classes of closed paths on X based at P.
The *first homology group* of X, defined as

$$H_1(X, \mathbb{Z}) = G/[G, G],$$

is a free abelian group of rank $2g$. A basis $(\alpha_i,\ \beta_j)$ $(1 \le i, j \le g)$ of $H_1(X, \mathbb{Z})$ is called *canonical* if the intersections satsify

$$\alpha_i \circ \beta_j = \delta_{ij} \quad \text{and} \quad \alpha_i \circ \alpha_j = \beta_i \circ \beta_j = 0$$



Period matrices

With respect to such bases we define the matrices

$$\Omega_A = \left(\int_{\alpha_i} \bar{\omega} \right) \quad \text{and} \quad \Omega_B = \left(\int_{\beta_i} \bar{\omega} \right)$$

Period matrices

With respect to such bases we define the matrices

$$\Omega_A = \left(\int_{\alpha_i} \bar{\omega} \right) \quad \text{and} \quad \Omega_B = \left(\int_{\beta_i} \bar{\omega} \right)$$

and call *big period matrix* the concatenated matrix

$$\Omega = (\Omega_A, \Omega_B) \in \mathbb{C}^{g \times 2g}.$$

Period matrices

With respect to such bases we define the matrices

$$\Omega_A = \left(\int_{\alpha_i} \bar{\omega} \right) \quad \text{and} \quad \Omega_B = \left(\int_{\beta_i} \bar{\omega} \right)$$

and call *big period matrix* the concatenated matrix

$$\Omega = (\Omega_A, \Omega_B) \in \mathbb{C}^{g \times 2g}.$$

We obtain a *small period matrix* in the Siegel upper half-space via

$$\tau = \Omega_A^{-1} \Omega_B \in \mathfrak{H}_g,$$

i.e. τ is a symmetric matrix in $\mathbb{C}^{g \times g}$ with positive definite imaginary part.

Jacobian & Abel-Jacobi map

The Jacobian $J = \mathrm{Jac}(X)$ can now be explicitly described as

$$J \cong \mathbb{C}^g/\Lambda, \quad \text{where} \quad \Lambda \cong \Omega\mathbb{Z}^{2g}.$$

Jacobian & Abel-Jacobi map

The Jacobian $J = \mathrm{Jac}(X)$ can now be explicitly described as

$$J \cong \mathbb{C}^g/\Lambda, \quad \text{where} \quad \Lambda \cong \Omega\mathbb{Z}^{2g}.$$

J is an abelian variety of dimension g and, in particular, a complex torus.

Jacobian & Abel-Jacobi map

The Jacobian $J = \mathrm{Jac}(X)$ can now be explicitly described as

$$J \cong \mathbb{C}^g/\Lambda, \quad \text{where} \quad \Lambda \cong \Omega\mathbb{Z}^{2g}.$$

J is an abelian variety of dimension g and, in particular, a complex torus.

Example: If $g = 1$, then $J \cong \mathbb{C}/\Lambda \cong X$ is an elliptic curve.

Jacobian & Abel-Jacobi map

The Jacobian $J = \mathrm{Jac}(X)$ can now be explicitly described as

$$J \cong \mathbb{C}^g/\Lambda, \quad \text{where} \quad \Lambda \cong \Omega\mathbb{Z}^{2g}.$$

J is an abelian variety of dimension g and, in particular, a complex torus.

Example: If $g = 1$, then $J \cong \mathbb{C}/\Lambda \cong X$ is an elliptic curve.

For a base point $P_0 \in X$ the *Abel-Jacobi map* is defined as

$$\mathcal{A} : X \to J,\ P \mapsto \int_{P_0}^{P} \bar{\omega} \mod \Lambda.$$

Jacobian & Abel-Jacobi map

The Jacobian $J = \mathrm{Jac}(X)$ can now be explicitly described as

$$J \cong \mathbb{C}^g/\Lambda, \quad \text{where} \quad \Lambda \cong \Omega\mathbb{Z}^{2g}.$$

J is an abelian variety of dimension g and, in particular, a complex torus.

Example: If $g = 1$, then $J \cong \mathbb{C}/\Lambda \cong X$ is an elliptic curve.

For a base point $P_0 \in X$ the *Abel-Jacobi map* is defined as

$$\mathcal{A} : X \to J,\ P \mapsto \int_{P_0}^{P} \bar{\omega} \mod \Lambda.$$

By linearity $\mathcal{A}$ can be extended to divisors on X.

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),
- the regulator pairing for K_2 of curves.

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z, \tau) = \sum_{v \in \mathbb{Z}^g} e^{2\pi i \left(\frac{1}{2} v^T \tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),
- the regulator pairing for K_2 of curves.

For all of these applications numerical integration has to be performed

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),
- the regulator pairing for K_2 of curves.

For all of these applications numerical integration has to be performed

- rigorously (provable error bounds)

Applications in number theory

Period matrices or more generally, integration of differential forms on a Riemann surface X, are required for computing

- the Abel-Jacobi map
- Theta functions $\Theta(z,\tau) = \sum_{v\in\mathbb{Z}^g} e^{2\pi i\left(\frac{1}{2}v^T\tau v + v^T z\right)}$,
- the endomorphism ring $\mathrm{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),
- the regulator pairing for K_2 of curves.

For all of these applications numerical integration has to be performed

- rigorously (provable error bounds)
- to high numerical precision (hundreds or thousands of digits)

Existing work

For genus 1 and 2 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

Existing work

For genus 1 and 2 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

For hyperelliptic curves of arbitrary genus there exist

- a Magma implementation due to P. van Wamelen,
- a Matlab implementation due to Frauendiener and Klein.

Existing work

For genus 1 and 2 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

For hyperelliptic curves of arbitrary genus there exist

- a Magma implementation due to P. van Wamelen,
- a Matlab implementation due to Frauendiener and Klein.

For general algebraic curves there are

- a Maple implementation due to Deconinck and van Hoeij,
- a Python/Sage implementation due to Swierczewski,
- a Matlab implementation due to Frauendiener and Klein,
- a Sage implementation due to Bruin is in progress.

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$
- lift paths to X via analytic continuation to obtain a monodromy representation of φ

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$
- lift paths to X via analytic continuation to obtain a monodromy representation of φ
- use the monodromy to compute a basis of $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$
- lift paths to X via analytic continuation to obtain a monodromy representation of φ
- use the monodromy to compute a basis of $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- compute symplectic base change matrix

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$
- lift paths to X via analytic continuation to obtain a monodromy representation of φ
- use the monodromy to compute a basis of $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- compute symplectic base change matrix
- perform numerical integration of differentials

Essential tasks

Starting with an affine equation $f(x, y)$, we obtain a big period matrix by working through the following list:

- compute a basis $\bar{\omega}$ of $\Omega^1(X)$
- choose closed paths in the complex plane that generate $\pi_1(\mathbb{P}_1(\mathbb{C}) \setminus \mathcal{B}, p_0)$
- lift paths to X via analytic continuation to obtain a monodromy representation of φ
- use the monodromy to compute a basis of $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- compute symplectic base change matrix
- perform numerical integration of differentials

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths (with Stefan Hellbusch),

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths (with Stefan Hellbusch),
- analytic continuation is done via simultaneous root approximation methods,

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths (with Stefan Hellbusch),
- analytic continuation is done via simultaneous root approximation methods,
- employs the double-exponential integration scheme.

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths (with Stefan Hellbusch),
- analytic continuation is done via simultaneous root approximation methods,
- employs the double-exponential integration scheme.

Our work

Magma implementation (A1) for general algebraic curves defined over number fields, based on the approach of Deconinck and van Hoeij:

- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths (with Stefan Hellbusch),
- analytic continuation is done via simultaneous root approximation methods,
- employs the double-exponential integration scheme.

Compared to the Maple implementation, we compute period matrices

- much faster and more reliably,
- to higher precision,
- for higher genera.

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3)

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3) that rigorously compute period matrices and evaluate the Abel-Jacobi map to arbitrary precision.

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3) that rigorously compute period matrices and evaluate the Abel-Jacobi map to arbitrary precision. More precisely:

- 'arbitrary' precision (realistically ≈ 10000 digits)

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3) that rigorously compute period matrices and evaluate the Abel-Jacobi map to arbitrary precision. More precisely:

- 'arbitrary' precision (realistically ≈ 10000 digits)
- excellent scaling with the genus ($g \gg 1000$ possible)

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3) that rigorously compute period matrices and evaluate the Abel-Jacobi map to arbitrary precision. More precisely:

- 'arbitrary' precision (realistically ≈ 10000 digits)
- excellent scaling with the genus ($g \gg 1000$ possible)
- extremely fast and numerically robust

New algorithm for superelliptic curves

Consider a superelliptic curve X given by an affine equation of the form

$$y^m = p(x),$$

where $m \geq 2, n = \deg(p) \geq 3$ and $p \in \mathbb{C}[x]$ is separable.

For such curves (in joint work with Pascal Molin) we developed and implemented algorithms in Magma (A2) and Arb (A3) that rigorously compute period matrices and evaluate the Abel-Jacobi map to arbitrary precision. More precisely:

- 'arbitrary' precision (realistically ≈ 10000 digits)
- excellent scaling with the genus ($g \gg 1000$ possible)
- extremely fast and numerically robust
- better than Magma for hyperelliptic curves

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\dfrac{x^i}{y^j}\mathrm{d}x$

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\dfrac{x^i}{y^j}\mathrm{d}x$

  $\rightarrow$ can find upper bounds for integrals

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\frac{x^i}{y^j}\mathrm{d}x$

  $\rightarrow$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\frac{x^i}{y^j}\mathrm{d}x$

  $\rightarrow$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically
- the monodromy at all finite branch points is equal and cyclic of order m

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\frac{x^i}{y^j}\mathrm{d}x$
  $\rightarrow$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically
- the monodromy at all finite branch points is equal and cyclic of order m
- we can explicitly write down a generating set for $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\dfrac{x^i}{y^j}\mathrm{d}x$
  $\rightarrow$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically
- the monodromy at all finite branch points is equal and cyclic of order m
- we can explicitly write down a generating set for $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- desingularize integrals between branch points symbolically
  $\rightarrow$ integration between branch points possible

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\frac{x^i}{y^j}\mathrm{d}x$
  $\rightarrow$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically
- the monodromy at all finite branch points is equal and cyclic of order m
- we can explicitly write down a generating set for $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- desingularize integrals between branch points symbolically
  $\rightarrow$ integration between branch points possible
- we can integrate along the edges of a spanning tree

Advantages in the superelliptic case

Why is the superelliptic case so much easier?

- holomorphic differentials are of the form $\frac{x^i}{y^j}\mathrm{d}x$
  $\to$ can find upper bounds for integrals
- need to compute only one (!) analytic continuation $y(x) = \sqrt[m]{p(x)}$, which can be done symbolically
- the monodromy at all finite branch points is equal and cyclic of order m
- we can explicitly write down a generating set for $H_1(X, \mathbb{Z})$ and the corresponding intersection matrix
- desingularize integrals between branch points symbolically
  $\to$ integration between branch points possible
- we can integrate along the edges of a spanning tree
  $\to$ only $g \times (n-1)$ integrations

Complexity

Let $X : y^m = p(x)$ be a superelliptic curve of genus g, where $p \in \mathbb{C}[x]$ is separable of degree n.

Complexity

Let $X : y^m = p(x)$ be a superelliptic curve of genus g, where $p \in \mathbb{C}[x]$ is separable of degree n.

Theorem

We can compute a basis of the period lattice Λ to precision D digits using

$$O(n(g + \log D)D^2 \log^{2+\varepsilon} D)$$

binary operations, where $\epsilon > 0$ is chosen such that multiplication of precision D numbers has complexity $O(D \log^{1+\epsilon} D)$.

Complexity

Let $X : y^m = p(x)$ be a superelliptic curve of genus g, where $p \in \mathbb{C}[x]$ is separable of degree n.

Theorem

We can compute a basis of the period lattice Λ to precision D digits using

$$O(n(g + \log D)D^2 \log^{2+\varepsilon} D)$$

binary operations, where $\epsilon > 0$ is chosen such that multiplication of precision D numbers has complexity $O(D \log^{1+\epsilon} D)$.

The complexity analysis for the computation of the standard period matrices is dominated by linear algebra.

Even the absolute timings are dominated by linear algebra for very large genera.

Timings

Computation* of the big period matrix $\Omega = (\Omega_A, \Omega_B)$ for the family of curves given by

- $(x+y)^{n-1} + x^n y^2 + 1 = 0$ up to 20 significant digits

n	2	3	4	5	6	7	8	9	10
g	1	2	6	10	14	21	28	35	45
Maple	1.7s	5.6s	39s	2m 10s	error	6m 45s	12m 58s	1h 14m	error
(A1)	0.3s	0.9s	3.2s	10s	24s	1m	2m 4s	4m 43s	11m 18s

Timings

Computation* of the big period matrix $\Omega = (\Omega_A, \Omega_B)$ for the family of curves given by

- $(x+y)^{n-1} + x^n y^2 + 1 = 0$ up to 20 significant digits

n	2	3	4	5	6	7	8	9	10
g	1	2	6	10	14	21	28	35	45
Maple	1.7s	5.6s	39s	2m 10s	error	6m 45s	12m 58s	1h 14m	error
(A1)	0.3s	0.9s	3.2s	10s	24s	1m	2m 4s	4m 43s	11m 18s

- $y^m = x^n + 1$ up to 500 significant digits

(m,n)	(2,5)	(2,11)	(2,31)	(2,101)	(3,5)	(3,11)	(7,11)	(77,11)	(11,31)	(101,31)
g	2	5	15	50	4	10	30	375	150	1500
Maple	17m	1h 4m	-	-	42m	5h	-	-	-	-
(A1)	28s	1m 30s	10m	2h	50s	2m 47s	12m 18s	-	4h 32m	-
Magma	1.6s	6.7s	1m 18s	1h 51m	\	\	\	\	\	\
(A2)	0.15s	0.55s	3.7s	39s	3.8s	11s	14s	1m 35s	1m 32s	44m 32s
(A3)	0.035s	0.14s	1.2s	13s	1.7s	3.4s	3.8s	2m 19s	34s	2h 25m

*done on Intel Xeon(R) CPU E3-1275 V2 3.50GHz processor.

Computations for the LMFDB

During a meeting aimed at expanding the 'L-functions and modular forms database' (LMFDB) to genus 3 curves, (A2) was used, in conjunction with methods of Jeroen Sijsling (et al.), to successfully compute the endomorphism rings of Jacobians of $67,879$ hyperelliptic curves of genus 3.

Computations for the LMFDB

During a meeting aimed at expanding the 'L-functions and modular forms database' (LMFDB) to genus 3 curves, (A2) was used, in conjunction with methods of Jeroen Sijsling (et al.), to successfully compute the endomorphism rings of Jacobians of $67,879$ hyperelliptic curves of genus 3.

Moreover, we have confirmed the endomorphism rings for all $66,158$ genus 2 curves that are currently in the LMFDB.

Computations for the LMFDB

During a meeting aimed at expanding the 'L-functions and modular forms database' (LMFDB) to genus 3 curves, (A2) was used, in conjunction with methods of Jeroen Sijsling (et al.), to successfully compute the endomorphism rings of Jacobians of $67,879$ hyperelliptic curves of genus 3.

Moreover, we have confirmed the endomorphism rings for all $66,158$ genus 2 curves that are currently in the LMFDB.

In the future (A1) is going to be used for ≈ 60000 smooth plane quartics.

Computations for the LMFDB

During a meeting aimed at expanding the 'L-functions and modular forms database' (LMFDB) to genus 3 curves, (A2) was used, in conjunction with methods of Jeroen Sijsling (et al.), to successfully compute the endomorphism rings of Jacobians of $67,879$ hyperelliptic curves of genus 3.

Moreover, we have confirmed the endomorphism rings for all $66,158$ genus 2 curves that are currently in the LMFDB.

In the future (A1) is going to be used for ≈ 60000 smooth plane quartics.

For these applications big period matrices were computed to 300 digits precision.

Links

- Our package for superelliptic curves is available at github:

  github.com/pascalmolin/hcperiods

- LMFDB: www.lmfdb.org
- Arb: www.arblib.org