Computing period matrices of algebraic curves

Christian Neurohr
University of Oldenburg

LMFDB meeting, Bristol,
28.03.2017

Definition

Let $\mathcal{C}$ be a smooth irreducible projective curve of genus g and let J be its Jacobian variety. Over the complex, J has the structure of a complex torus

$$J(\mathbb{C}) \cong \mathbb{C}^g/\Lambda,$$

where Λ is a full rank lattice, called *period lattice* of $\mathcal{C}$.

Definition

Let $\mathcal{C}$ be a smooth irreducible projective curve of genus g and let J be its Jacobian variety. Over the complex, J has the structure of a complex torus

$$J(\mathbb{C}) \cong \mathbb{C}^g/\Lambda,$$

where Λ is a full rank lattice, called *period lattice* of $\mathcal{C}$.
For every basis $\omega_1, \ldots, \omega_g$ of the space of holomorphic differentials $\Omega^1_{\mathcal{C}}$ we have that

$$\Lambda \cong \left\{ \int_\gamma \bar{\omega}, \gamma \in H_1(\mathcal{C}, \mathbb{Z}) \right\} \subset \mathbb{C}^g,$$

where $\bar{\omega} = (\omega_1, \ldots, \omega_g)$ and $H_1(\mathcal{C}, \mathbb{Z}) \cong \mathbb{Z}^{2g}$ is the first homology group of the curve.

Definition (cont.)

Choosing a symplectic basis α_i, β_j $(1 \leq i, j \leq g)$ of $H_1(\mathcal{C}, \mathbb{Z})$, we define the matrices

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

Definition (cont.)

Choosing a symplectic basis α_i, β_j $(1 \leq i, j \leq g)$ of $H_1(\mathcal{C}, \mathbb{Z})$, we define the matrices

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

and call *big period matrix* the concatenated matrix

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

such that $\Lambda = \Omega \mathbb{Z}^{2g}$.

Definition (cont.)

Choosing a symplectic basis α_i, β_j $(1 \leq i, j \leq g)$ of $H_1(\mathcal{C}, \mathbb{Z})$, we define the matrices

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

and call *big period matrix* the concatenated matrix

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

such that $\Lambda = \Omega \mathbb{Z}^{2g}$. We obtain a *small period matrix* in the Siegel upper half-space via

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

Applications in number theory

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

Applications in number theory

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

- the Abel-Jacobi map

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

Applications in number theory

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

- the Abel-Jacobi map

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

- Theta functions

$$\Theta(z, \tau) = \sum_{v \in \mathbb{Z}^g} \exp(2\pi i((1/2)\, v^T \tau v + v^T z)),$$

Applications in number theory

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

- the Abel-Jacobi map

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

- Theta functions

$$\Theta(z,\tau) = \sum_{v \in \mathbb{Z}^g} \exp(2\pi i((1/2)\, v^T \tau v + v^T z)),$$

- 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, are required for computing

- the Abel-Jacobi map

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

- Theta functions

$$\Theta(z, \tau) = \sum_{v \in \mathbb{Z}^g} \exp(2\pi i((1/2)\, v^T \tau v + v^T z)),$$

- 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, are required for computing

- the Abel-Jacobi map

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

- Theta functions

$$\Theta(z, \tau) = \sum_{v \in \mathbb{Z}^g} \exp(2\pi i((1/2)\, v^T \tau v + v^T z)),$$

- the endomorphism ring $\operatorname{End}(J)$ (numerical approximation),
- the real period of J (appearing in the BSD conjecture),
- the regulator pairing for K_2 of curves.

Existing work

For genus $1, 2$ and 3 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

For modular curves, termwise integration of modular forms is possible and very efficient.

Existing work

For genus $1, 2$ and 3 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

For modular curves, termwise integration of modular forms is possible and very efficient.

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 $1, 2$ and 3 period matrices can be computed in almost linear time to arbitrary precision (AGM, Borchardt mean).

For modular curves, termwise integration of modular forms is possible and very efficient.

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 from an affine equation for the curve

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

we obtain a period matrix by working through the following list:

Essential tasks

Starting from an affine equation for the curve

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

we obtain a period matrix by working through the following list:

- computing a basis of holomorphic differentials
- choosing integration path $\rightarrow$ analytic continuation
- numerical integration
- use the monodromy to compute a homology basis
- compute intersection matrix and symplectic base change

Our work (soon available)

Magma implementation for general algebraic curves (A1):

- based on the approach of Deconinck and van Hoeij,
- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths,
- analytic continuation is done via root approximation methods,
- employs the double-exponential integration scheme.

Our work (soon available)

Magma implementation for general algebraic curves (A1):

- based on the approach of Deconinck and van Hoeij,
- computes differentials using Magma's function fields,
- uses spanning tree methods to construct paths,
- analytic continuation is done via 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 supereliptic curves

Consider a superelliptic curve given by an equation of the form

$$\mathcal{C} : y^m = f(x),$$

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

New algorithm for supereliptic curves

Consider a superelliptic curve given by an equation of the form

$$\mathcal{C} : y^m = f(x),$$

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

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

New algorithm for supereliptic curves

Consider a superelliptic curve given by an equation of the form

$$\mathcal{C} : y^m = f(x),$$

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

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

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

Timings

Computation* of $\tau \in \mathfrak{H}_g$ for the family of curves given by

- $f_n = (x+y)^{n-1} + x^n y^2 + 1$ 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
t_{Maple}	2.1s	6s	39s	2m 10s	error	6m 45s	12m 58s	-	error
t_{A1}	0.3s	0.8s	4.5s	15s	44s	2m 22s	5m 14s	12m 37s	30m 45s

Timings

Computation* of $\tau \in \mathfrak{H}_g$ for the family of curves given by

- $f_n = (x+y)^{n-1} + x^n y^2 + 1$ 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
t_{Maple}	2.1s	6s	39s	2m 10s	error	6m 45s	12m 58s	-	error
t_{A1}	0.3s	0.8s	4.5s	15s	44s	2m 22s	5m 14s	12m 37s	30m 45s

- $f_{m,n} = y^m - \sum_{k=0}^{n} x^k$ up to 500 significant digits

(m,n)	(2,5)	(2,11)	(2,31)	(2,101)	(3,5)	(3,11)	(7,5)	(77,5)	(11,21)	(31,21)
g	2	5	15	50	4	10	12	152	100	300
t_{A1}	32s	2m 30s	33m	-	1m 4s	5m 27s	4m 36s	-	38m	-
t_{A2}	0.2s	0.6s	3.7s	39s	4.8s	15s	6.7s	1m 26s	2m 4s	11m 14s
t_{Magma}	1.6s	6.7s	1m 23s	-	/	/	/	/	/	/

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