

```

gap> g:= SymmetricGroup( 4 );
Sym( [ 1 .. 4 ] )
gap> tbl:= CharacterTable( g );; HasIrr( tbl );
i5 : betti(t,Weights=>{1,0})
false
      0 1 2 3 4
o5 = total: 1 4 13 14 4
      0: 1 . . . .
      1: . 2 2 4 2
      2: . 2 5 6 .
      3: . . 4 . 2
      4: . . . 4 .
      5: . . 2 . .
gap> tblmod2:= CharacterTable( tbl, 2 );
BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
gap> tblmod2 = CharacterTable( tbl, 2 );
true
gap> tblmod2 = BrauerTable( tbl, 2 );
true
o5 : BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
i6 : betti(t,Weights=>{0,1})
      0 1 2 3 4
o6 = total: 1 4 13 14 4
      0: 1 . . . .
      1: . 2 2 4 2
      2: . 2 5 6 .
      3: . . 4 . 2
      4: . . . 4 .
      5: . . 2 . .
gap> libtbl:= CharacterTable( "M" );
CharacterTable( "M" )
gap> CharacterTableRegular( libtbl, 2 );
BrauerTable( "M", 2 )
gap> BrauerTable( libtbl, 2 );
fail
gap> CharacterTable( "Symmetric", 4 );
CharacterTable( "Sym(4)" )
i7 : t1 = betti(t,Weights=>{1,1})
gap> ComputedBrauerTables( tbl );
[ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ) ]
      0 1 2 3 4
o7 = total: 1 4 13 14 4
      0: 1 . . . .
      1: . . . . .
      2: . . . . .
      3: . 2 . . .
      4: . . . . .
      5: . 2 . . .
      6: . . 1 . .
      7: . . 8 6 .
      8: . . 4 8 4
      ring r1 = 32003,(x,y,z),ds;
      int a,b,c,t=11,5,3,0;
      poly f = x^a+y^b+z^(3*c)+x^(c+2)*y^(c-1)+x^(
      x^(c-2)*y^c*(y^2+t*x)^2;
      option(noprot);
      timer=1;
      ring r2 = 32003,(x,y,z),dp;
      poly f=imap(r1,f);
      ideal j=jacob(f);
      vdim(std(j));
==> 536
      vdim(std(j+f));
==> 195
      timer=0; // reset timer
o7 : BettiTally
o8 : peek t1
o8 = BettiTally{(0, {0, 0}, 0) => 1 }
      (1, {2, 2}, 4) => 2
      (1, {3, 3}, 6) => 2
      (2, {3, 7}, 10) => 2
      (2, {4, 4}, 8) => 1
      (2, {4, 5}, 9) => 4
      (2, {5, 4}, 9) => 4
      (2, {7, 3}, 10) => 2
      (3, {4, 7}, 11) => 4
      (3, {5, 5}, 10) => 6
      (3, {7, 4}, 10) => 6
      (4, {5, 7}, 12) => 2
      (4, {7, 5}, 12) => 2

```

Journal of Software for Algebra and Geometry

Computing theta functions with Julia

DANIELE AGOSTINI AND LYNN CHUA

Computing theta functions with Julia

DANIELE AGOSTINI AND LYNN CHUA

ABSTRACT: We present a new package `Theta.jl` for computing the Riemann theta function. It is implemented in Julia and offers accurate numerical evaluation of theta functions with characteristics and their derivatives of arbitrary order. Our package is optimized for multiple evaluations of theta functions for the same Riemann matrix, in small dimensions. As an application, we report on experimental approaches to the Schottky problem in genus 5.

1. INTRODUCTION. The *Riemann theta function* is the holomorphic function

$$\theta : \mathbb{C}^g \times \mathbb{H}_g \rightarrow \mathbb{C}, \quad \theta(z, \tau) = \sum_{n \in \mathbb{Z}^g} e\left(\frac{1}{2}n^t \tau n + n^t z\right) \quad (1)$$

where $e(x) = e^{2\pi i x}$ and \mathbb{H}_g is the *Siegel upper-half space*, which consists of all complex symmetric $g \times g$ matrices with positive definite imaginary part. Theta functions occupy a central role throughout mathematics, appearing in fields as diverse as algebraic geometry [Birkenhake and Lange 2004; Mumford 2007], number theory [Mumford 2007; Eichler and Zagier 1985], differential geometry [Agostini et al. 2020], integrable systems [Krichever and Shiota 2013; Segur 2008], discrete mathematics [Regev and Stephens-Davidowitz 2017], cryptography [Gaudry 2007] and statistics [Agostini and Améndola 2019].

We present a new package `Theta.jl` for numerical computations of theta functions, programmed in Julia [Bezanson et al. 2017]. Our package is specialized for multiple evaluations of theta functions for the same Riemann matrix $\tau \in \mathbb{H}_g$ and different z , for small values of the genus g . Our implementation is based on the algorithm from [Deconinck et al. 2004], which we extend to support computations of theta functions with characteristics and derivatives of arbitrary order. Our package is designed as an alternative to existing packages such as `algcurves` [Deconinck et al. 2004] in Maple, `abelfunctions` [Swierczewski and Deconinck 2016] in Sage and [Frauendiener et al. 2019] in Matlab, with additional functionalities and optimizations.

As an application, we study numerical approaches to the Schottky problem in genus 5. The Schottky problem seeks to recognize Jacobians of curves amongst principally polarized abelian varieties, and has been one of the central questions in algebraic geometry since the 19th century [Grushevsky 2012]. The first nontrivial case of the Schottky problem is in genus 4, which is completely solved [Igusa 1981]. For a recent approach linking computations and tropical geometry, see [Chua et al. 2019]. In this paper, we

MSC2010: 14-04, 14H42, 14K25, 32-04, 65E99.

Keywords: theta function, numerical, Julia, abelian varieties, Riemann surfaces, Schottky problem, Jacobian.

`Theta.jl` version 1.0

describe computational approaches for studying the Schottky problem in genus 5, using our new package. In particular, we use `Theta.jl` to compute the equations in [Farkas et al. 2021; Accola 1983] which give a weak solution to the Schottky problem in genus 5. We also use our package for computations on the genus 5 Schottky problem for Jacobians with a vanishing theta null, which is described in our companion paper [Agostini and Chua 2019].

2. THETA FUNCTIONS. We recall here the basic definitions about theta functions with characteristics. For a more detailed account we refer to [Birkenhake and Lange 2004; Mumford 2007; Igusa 1981]. A *characteristic* is an element $m \in (\mathbb{Z}/2\mathbb{Z})^{2g}$, which we represent as a vector $m = \begin{bmatrix} \varepsilon \\ \delta \end{bmatrix}$ where $\varepsilon, \delta \in \{0, 1\}^g$. The *Riemann theta function with characteristic m* is defined as

$$\theta[m](z, \tau) = \theta \begin{bmatrix} \varepsilon \\ \delta \end{bmatrix} (z, \tau) = \sum_{n \in \mathbb{Z}^g} e \left(\frac{1}{2} \left(n + \frac{\varepsilon}{2} \right)^t \tau \left(n + \frac{\varepsilon}{2} \right) + \left(n + \frac{\varepsilon}{2} \right)^t \left(z + \frac{\delta}{2} \right) \right) \quad (2)$$

and it is a holomorphic function $\theta[m] : \mathbb{C}^g \times \mathbb{H}_g \rightarrow \mathbb{C}$. The Riemann theta function in (1) is a special case of (2), where the characteristic is the all-zero vector. The *sign* of a characteristic m is defined as $e(m) = (-1)^{\varepsilon^t \delta}$, and we call a characteristic *even* or *odd* if the sign is 1 or -1 , respectively. As a function of z , $\theta[m](z, \tau)$ is even (respectively, odd) if and only if the characteristic m is even (respectively, odd). There are $2^{g-1}(2^g + 1)$ even theta characteristics and $2^{g-1}(2^g - 1)$ odd theta characteristics.

The *theta constants* are the functions on \mathbb{H}_g obtained by evaluating the theta functions with characteristics at $z = 0$,

$$\theta[m](\tau) = \theta[m](0, \tau). \quad (3)$$

Theta constants corresponding to odd characteristics vanish identically.

3. NUMERICALLY APPROXIMATING THETA FUNCTIONS. We describe in this section the algorithm that we use to compute theta functions in `Theta.jl`. In our implementation, we modify the algorithm from [Deconinck et al. 2004], generalizing it for theta functions with characteristics and derivatives of arbitrary order.

In this section, we separate $z \in \mathbb{C}^g$ and $\tau \in \mathbb{H}_g$ into real and imaginary parts, by writing $z = x + iy$, $\tau = X + iY$, where $x, y \in \mathbb{R}^g$ and X, Y are real symmetric $g \times g$ matrices. We also denote by $Y = T^t T$ the Cholesky decomposition of Y , where T is upper-triangular. For any real vector $V \in \mathbb{R}^g$, we use $[V]$ to denote the vector whose entries are the entries of V rounded to the closest integers, and we denote $\llbracket V \rrbracket = V - [V]$.

We set $v(n) = \sqrt{\pi} T(n + \llbracket Y^{-1}y \rrbracket)$ and we define the lattice $\Lambda = \{v(n) \mid n \in \mathbb{Z}^g\}$, letting ρ be the length of the shortest nonzero vector in Λ . We denote by $\Gamma(z, x) = \int_x^\infty t^{z-1} e^{-t} dt$ the incomplete Gamma function.

3A. Theta functions with characteristics. Deconinck et al. [2004] derive numerical approximations of the theta function and its first and second derivatives. We extend their results for computing theta functions with characteristics and derivatives of arbitrary order.

We denote the N -th order derivative of the theta function along the vectors $k^{(1)}, \dots, k^{(N)}$ as

$$D(k^{(1)}, \dots, k^{(N)})\theta(z, \tau) = \sum_{i_1, \dots, i_N=1}^g k_{i_1}^{(1)} \dots k_{i_N}^{(N)} \frac{\partial^N \theta(z, \tau)}{\partial z_{i_1} \dots \partial z_{i_N}}. \quad (4)$$

By the quasiperiodicity of the theta function, it suffices to consider inputs z of the form $z = a + \tau b$, for $a, b \in [0, 1)^g$.

Theorem 3.1. Fix $\tau \in \mathbb{H}_g$, $\epsilon > 0$. Let $k^{(1)}, \dots, k^{(N)} \in \mathbb{C}^g$ be unit vectors, and let R be the greater of $\frac{1}{2}\sqrt{g + 2N} + \sqrt{g^2 + 8N} + \frac{\rho}{2}$ and the real positive solution of R in

$$\epsilon = (2\pi)^N \frac{g}{2} \left(\frac{2}{\rho}\right)^g \sum_{j=0}^N \binom{N}{j} \frac{1}{\pi^{j/2}} \|T^{-1}\|^j \sqrt{g}^{N-j} \Gamma\left(\frac{g+j}{2}, \left(R - \frac{\rho}{2}\right)^2\right). \quad (5)$$

For z of the form $z = a + \tau b$, for $a, b \in [0, 1)^g$, and $\begin{bmatrix} \epsilon \\ \delta \end{bmatrix} \in \{0, 1\}^{2g}$, the N -th derivative

$$D(k^{(1)}, \dots, k^{(N)})\theta\begin{bmatrix} \epsilon \\ \delta \end{bmatrix}(z, \tau)$$

of the theta function with characteristic is approximated by

$$e^{\pi y^t Y^{-1} y} (2\pi i)^N \sum_{n \in C_R} (k^{(1)} \cdot (n - \eta)) \dots (k^{(N)} \cdot (n - \eta)) \times e\left(\frac{1}{2}(n - \eta)^t X(n - \eta) + (n - \eta)^t \left(x + \frac{\delta}{2}\right)\right) e^{-\|v(n + \frac{\epsilon}{2})\|^2}, \quad (6)$$

with an absolute error ϵ on the product of $(2\pi i)^N$ with the sum, where $\eta = [Y^{-1}y] - \frac{\epsilon}{2}$ and

$$C_R = \{n \in \mathbb{Z}^g \mid \pi(n - c)^t Y(n - c) < R^2, |c_j| < 1, \text{ for all } j = 1, \dots, g\}. \quad (7)$$

Proof. We first consider the case $N = 0$ without derivatives. Then the result for characteristics $\epsilon = \delta = 0$ is proven in [Deconinck et al. 2004, Theorem 2], where they replace the deformed ellipsoid C_R in (7) with the ellipsoid

$$S_R = \{n \in \mathbb{Z}^g \mid \|v(n)\| < R\}. \quad (8)$$

For arbitrary characteristics ϵ, δ , we see from (2) that we can compute the corresponding theta function in a similar way to the usual theta function, by translating z to $z + \frac{\epsilon}{2}$, and translating the lattice points in the sum from n to $n + \frac{\epsilon}{2}$. Note that this only changes the real part of z , while the imaginary part stays the same. Hence the approximation in Theorem 3.1 holds for theta functions with characteristics, if we take the sum over the ellipsoid

$$S_{R, \epsilon} = \left\{n \in \mathbb{Z}^g \mid \left\|v\left(n + \frac{\epsilon}{2}\right)\right\| < R\right\}. \quad (9)$$

To obtain a uniform approximation for any $z \in \mathbb{C}^g$ and any characteristic, we take the union of the ellipsoids $S_{R, \epsilon}$ from (9) as z and ϵ vary. Since $v(n + \frac{\epsilon}{2}) = \sqrt{\pi}T(n + \llbracket Y^{-1}y \rrbracket + \frac{\epsilon}{2})$, and the entries

of $\|Y^{-1}y\| + \frac{\varepsilon}{2}$ have absolute value at most 1, it follows that the deformed ellipsoid C_R from (7) is the union of the ellipsoids $S_{R,\varepsilon}$.

To prove the result in the case of derivatives of order N , it will be enough to prove it for the case of zero characteristics, and then follow the same strategy as above. More precisely, we are going to prove the same statement as in Theorem 3.1, where $\varepsilon = \delta = 0$ and C_R is replaced by

$$U_R = \{n \in \mathbb{Z}^g \mid \pi(n-c)^t Y(n-c) < R^2, |c_j| < 1/2, \text{ for all } j = 1, \dots, g\}. \quad (10)$$

To do so, we write the derivative as

$$D(k^{(1)}, \dots, k^{(N)})\theta(z, \tau) = (2\pi i)^N \sum_{n \in \mathbb{Z}^g} (k^{(1)} \cdot n) \cdots (k^{(N)} \cdot n) e(\frac{1}{2}n^t \tau n + n^t z)$$

and then the error in the approximation is

$$\epsilon = \left| (2\pi i)^N \sum_{n \in \mathbb{Z}^g \setminus U_R} (k^{(1)} \cdot (n - [Y^{-1}y])) \cdots (k^{(N)} \cdot (n - [Y^{-1}y])) \right. \\ \left. \times e(\frac{1}{2}(n - [Y^{-1}y])^t X(n - [Y^{-1}y]) + (n - [Y^{-1}y])^t x) e^{-\|v(n)\|^2} \right|$$

Since the $k^{(i)}$ have norm one, using the triangle inequality and the Cauchy–Schwarz inequality we can bound this by

$$\epsilon \leq (2\pi)^N \sum_{n \in \mathbb{Z}^g \setminus U_R} \|n - [Y^{-1}y]\|^N e^{-\|v(n)\|^2} = (2\pi)^N \sum_{n \in \mathbb{Z}^g \setminus U_R} \left\| \frac{1}{\sqrt{\pi}} T^{-1}v(n) - Y^{-1}y \right\|^N e^{-\|v(n)\|^2}.$$

Using again the triangle inequality and the binomial expansion, we get to the bound

$$\epsilon \leq (2\pi)^N \sum_{j=0}^N \binom{N}{j} \frac{1}{\pi^{j/2}} \|T^{-1}\|^j \|Y^{-1}y\|^{N-j} \sum_{n \in \mathbb{Z}^g \setminus U_R} \|v(n)\|^j e^{-\|v(n)\|^2}.$$

We then apply [Deconinck et al. 2004, Lemma 2] to get the bound

$$\epsilon \leq (2\pi)^N \sum_{j=0}^N \binom{N}{j} \frac{1}{\pi^{j/2}} \|T^{-1}\|^j \|Y^{-1}y\|^{N-j} \frac{g}{2} \left(\frac{2}{\rho}\right)^g \Gamma\left(\frac{g+j}{2}, \left(R - \frac{\rho}{2}\right)^2\right), \\ \leq (2\pi)^N \frac{g}{2} \left(\frac{2}{\rho}\right)^g \sum_{j=0}^N \binom{N}{j} \frac{1}{\pi^{j/2}} \|T^{-1}\|^j \|Y^{-1}y\|^{N-j} \Gamma\left(\frac{g+j}{2}, \left(R - \frac{\rho}{2}\right)^2\right).$$

For inputs z of the form $z = a + \tau b$, we can write z as $z = a + (X + iY)b = (a + Xb) + iYb = x + iy$. Then $\|Y^{-1}y\| = \|b\| \leq \sqrt{g}$. Substituting this into the expression for ϵ , the result follows. \square

Remark 3.2. The R appearing in Theorem 3.1 is computed numerically.

4. COMPUTING THETA FUNCTIONS IN JULIA.

4A. Interface. Our Julia package `Theta.jl` is available at the following website, which has instructions and a link to more detailed documentation:

<https://github.com/chualynn/Theta.jl>

We describe the basic interface of the package here. Starting with a matrix $\tau \in \mathbb{H}_g$, we first construct a `RiemannMatrix` from it. This is a type in `Theta.jl` which contains information needed to compute the theta function with input τ . As an example, we start with a genus 5 curve defined by the singular model

$$x^6y^2 - 4x^4y^2 - 2x^3y^3 - 2x^4y + 2x^3y + 4x^2y^2 + 3xy^3 + y^4 + 4x^2y + 2xy^2 + x^2 - 4xy - 2y^2 - 2x + 1. \quad (11)$$

We compute the Riemann matrix τ of the curve using the package [Bruin et al. 2019] in [SageMath], and we type it as an input in Julia. We then construct a `RiemannMatrix` in `Theta.jl`, where we specify in the input the options to compute a Siegel transformation, an error of 10^{-12} , and to compute derivatives up to the fourth order.

```
julia> R = RiemannMatrix($\tau$, siegel=true, $\epsilon=1.0e-12$, nderivs=4);
```

We pick some input z and compute the theta function $\theta(z, \tau)$ as follows:

```
julia> z = [1.041+0.996im; 1.254+0.669im; 0.591+0.509im; -0.301+0.599im; 0.388+0.051im];
julia> theta(z, R)
-854877.6514446283 + 2.3935081163150463e6im
```

We can compute derivatives of theta functions by specifying the directions using the optional argument `derivs`. For instance, to compute $\frac{\partial^3 \theta}{\partial z_3 \partial z_4}(z, \tau)$, we use

```
julia> theta(z, R, derivs=[[0,0,1,0,0], [0,0,0,1,0]])
1.0478325534969474e8 - 3.369999441122761e8im
```

We can also compute derivatives of theta functions with characteristics, where we specify the characteristic using the optional argument `char`.

```
julia> theta(z, R, derivs=[[1,0,0,0,0]], char=[[0,1,0,0,1], [1,1,0,0,1]])
-2.448093122926732e7 + 3.582557740667034e7im
```

4B. Algorithms. We describe here some details of the algorithms and the design choices that we made in our implementation.

Choice of ellipsoid. We optimize our package for multiple evaluations of theta functions at the same Riemann matrix τ , and with different inputs z , characteristics and derivatives. We do this using the approximation in Theorem 3.1, which allows us to compute derivatives of theta functions with characteristics, for inputs z of the form $z = a + \tau b$, for $a, b \in [0, 1]^g$. In this approximation, we take the sum over the deformed ellipsoid C_R of (7), which depends only on the order N of the derivative for a fixed τ . Hence it suffices to compute the ellipsoids C_R once for each order of the derivative that we are interested in, after which we can compute theta functions for any N -th order derivatives. These ellipsoids are stored in the `RiemannMatrix` type.

Lattice reductions. In [Deconinck et al. 2004], the authors approximate the length ρ of the shortest vector of the lattice generated by T using the LLL algorithm by Lenstra, Lenstra and Lovász [Lenstra et al. 1982].

This is a reasonable choice if g is large, since the LLL algorithm gives a polynomial time approximation, but with an error that grows exponentially with g . In our implementation, since we focus on lattices with small dimensions, we compute the shortest vector exactly using the enumeration algorithm in [Schnorr and Euchner 1994]. Moreover, by computing ρ exactly, we obtain a smaller ellipsoid (7) than if we use the LLL algorithm.

If we are interested in computing the theta function for a fixed τ at many values of z , it may be more efficient if we transform τ such that the ellipsoids in (7) contain fewer lattice points. For this purpose, we use Siegel’s algorithm, which iteratively finds a new matrix where the corresponding ellipsoid has a smaller eccentricity. In our implementation, we compute the Siegel transformation once for each Riemann matrix, and work with the Siegel-transformed matrix for all computations. We use the algorithm for Siegel reduction described in [Deconinck et al. 2004; Frauendiener et al. 2019], where we use the algorithm for HKZ reduction in [Zhang et al. 2012] as a subroutine.

4C. Comparisons with other packages. The main advantage of `Theta.jl` over other packages [Deconinck et al. 2004; Frauendiener et al. 2019; Swierczewski and Deconinck 2016] is that we support computations of theta functions with characteristics, as well as their derivatives, which to our knowledge is not implemented elsewhere. Moreover, we make optimizations described in Section 4B for faster computations with a fixed Riemann matrix of low genus.

We compare the performance of `Theta.jl` with the Sage package `abelfunctions` [Swierczewski and Deconinck 2016], by comparing the average time taken to compute the genus 5 FGSM relations of Section 5A, as well as to compute the Hessian matrix of Section 5C. For our experiments, we sample matrices in the Siegel upper-half space as follows. First we sample 5×5 matrices M_X, M_Y such that the entries are random floating point numbers between -1 and 1 , using the random number generators in Julia and NumPy. Then we sample $\tau \in \mathbb{H}_5$ as $\tau = \frac{1}{2}(M_X + M_X^t) + M_Y^t M_Y i$. This is implemented in `Theta.jl` for general dimensions g , in the function `random_siegel(g)`. In each experiment, we randomly sample 1000 such matrices, then we compute the FGSM relations and the Hessian matrix using both packages on a standard laptop. We list in the table below the average time and standard deviation.

experiment	package	average time (s)	standard deviation (s)
FGSM	<code>Theta.jl</code>	2.5	0.6
	<code>abelfunctions</code>	114.2	290.5
Hessian	<code>Theta.jl</code>	0.7	0.2
	<code>abelfunctions</code>	20.3	58.0

One major reason for the faster runtime on `Theta.jl` is the use of the Siegel transformation on the Riemann matrix, which is not implemented in `abelfunctions`. This also leads to the higher standard deviation in the computations for the latter.

5. APPLICATIONS TO THE SCHOTTKY PROBLEM IN GENUS 5. Here we describe the main application that we had in mind when designing our package: experiments around the Schottky problem in genus 5. We start with a brief account of the background of the problem; see [Grushevsky 2012] for more details.

An abelian variety is a projective variety that has the structure of an algebraic group, and it is a fundamental object in algebraic geometry. Especially important are principally polarized abelian varieties, which can all be described in terms of Riemann matrices. For every $\tau \in \mathbb{H}_g$, we define the corresponding *principally polarized abelian variety (ppav)* as the quotient $A_\tau = \mathbb{C}^g / \Lambda_\tau$, where $\Lambda_\tau = \mathbb{Z}^g \oplus \tau \mathbb{Z}^g$ is a sublattice of \mathbb{C}^g . The polarization on A_τ is given by the *theta divisor*

$$\Theta_\tau = \{z \in A_\tau \mid \theta(z, \tau) = 0\}. \quad (12)$$

Two ppavs A_τ and $A_{\tau'}$ are isomorphic if and only if the corresponding Riemann matrices are related via an action of the symplectic group $\Gamma_g = \text{Sp}(2g, \mathbb{Z})$. Hence, the quotient $\mathcal{A}_g = \mathbb{H}_g / \text{Sp}(2g, \mathbb{Z})$ is the *moduli space of principally polarized abelian varieties of dimension g* . This is a quasiprojective variety of dimension $\dim \mathcal{A}_g = \dim \mathbb{H}_g = \frac{1}{2}g(g+1)$, and we can look at the theta constants $\theta[m](0, \tau)$ as homogeneous coordinates on (a finite cover of) \mathcal{A}_g .

Perhaps the most important examples of abelian varieties are Jacobians of Riemann surfaces. The *Jacobian* of a Riemann surface C of genus g is defined as the quotient

$$J(C) = H^0(C, \omega_C)^\vee / H_1(C, \mathbb{Z}), \quad (13)$$

where the lattice $H^1(C, \mathbb{Z})$ is embedded in $H^0(C, \omega_C)^\vee$ via the integration pairing

$$H^0(C, \omega_C) \times H^1(C, \mathbb{Z}) \rightarrow \mathbb{C}, \quad (\omega, \alpha) \mapsto \int_\alpha \omega. \quad (14)$$

The Jacobian is a principally polarized abelian variety, and the corresponding Riemann matrix $\tau \in \mathcal{A}_g$ can be obtained by computing bases of $H^0(C, \omega_C)$ and $H^1(C, \mathbb{Z})$, as well as the integration pairing. This is implemented numerically in the packages `abelfunctions` [Swierczewski and Deconinck 2016] and `RiemannSurfaces` [Bruin et al. 2019] in Sage, and `algcurses` [Deconinck et al. 2004] in Maple.

The *Schottky locus* \mathcal{J}_g is the closure of the set of Jacobian varieties in \mathcal{A}_g , and the *Schottky problem* asks for a characterization of \mathcal{J}_g inside \mathcal{A}_g . It is one of the most celebrated questions in algebraic geometry, dating from the 19th century. There are many possible interpretations of and solutions to the Schottky problem. Here we focus on the most classical one, which asks for equations in the theta constants $\theta[m](0, \tau)$ that vanish exactly on the Schottky locus. In this form, the Schottky problem is completely solved only in genus 4, with an explicit equation given by Schottky and Igusa [Igusa 1981]. A computational implementation and analysis of this solution was presented in [Chua et al. 2019].

The *weak Schottky problem* asks for explicit equations that characterize Jacobians up to extra irreducible components. A solution to this problem was given in genus 5 by Accola [1983], and in a recent breakthrough, by Farkas, Grushevsky and Salvati Manni in all genera [Farkas et al. 2021]. In the rest of this section, we discuss briefly these two solutions, together with related algorithms that we implemented in `Theta.jl`. We also present a computational solution of a weak Schottky problem for genus 5 Jacobians with a theta null, from our companion paper [Agostini and Chua 2019].

5A. Farkas, Grushevsky and Salvati Manni's solution. In [Farkas et al. 2021], H. Farkas, Grushevsky and Salvati Manni give a solution to the weak Schottky problem in arbitrary genus. More precisely, for

every genus $g \geq 4$ they give $\binom{g-2}{2} = \frac{1}{2}(g-2)(g-3)$ explicit homogeneous equations of degree $2^{3 \cdot 2^{g-4} + 1}$ in the theta constants, such that their zero locus contains the Schottky locus as an irreducible component.

In the case of genus 5, this gives three equations of degree 128. We implement them in the function `fgsm()` in `Theta.jl`. Using the same example matrix τ from Section 4A, the function `fgsm(τ)` gives us the output 7.850462293418876e-16. This is expected since τ is the Jacobian of a genus 5 curve.

5B. Accola's equations in genus 5. A solution to the weak Schottky problem in genus 5 was given in [Accola 1983], in the form of eight equations of degree 32 in the theta constants whose zero locus contains the Schottky locus as an irreducible component. We implement these equations in the function `accola()` in `Theta.jl`. Again using the example τ from Section 4A, the function `accola(τ)` gives us the output 3.062334813867916e-9, which is expected since τ is in the Schottky locus.

5C. The Schottky problem for Jacobians with a vanishing theta null. We describe here a variant of the Schottky problem focusing on two-torsion points on Jacobians, referring to our companion article [Agostini and Chua 2019] for a more complete account. A *two-torsion point* on an abelian variety A_τ is a point $z \in A_\tau$ such that $2z = 0$. These can be written as

$$z = \frac{\varepsilon}{2} + \tau \frac{\delta}{2}, \quad \text{for } m = \begin{bmatrix} \varepsilon \\ \delta \end{bmatrix} \in (\mathbb{Z}/2\mathbb{Z})^{2g}. \quad (15)$$

Hence two-torsion points correspond to characteristics, and we say that such a point is *even* or *odd* if the corresponding characteristic is. Observe that

$$\theta\left(\frac{\varepsilon}{2} + \tau \frac{\delta}{2}, \tau\right) = 0 \quad \text{if and only if} \quad \theta \begin{bmatrix} \varepsilon \\ \delta \end{bmatrix} (0, \tau) = 0. \quad (16)$$

Thus the two-torsion points in Θ_τ correspond to the characteristics m such that the theta constants $\theta[m](0, \tau)$ vanish. For this reason, we say that A_τ has a *vanishing theta null* if it has an even two-torsion point in the theta divisor. The abelian varieties with this property have been intensely studied and they form a divisor θ_{null} in \mathcal{A}_g . The Jacobians with a vanishing theta null lie in the locus $\mathcal{J}_g \cap \theta_{\text{null}}$ and they correspond to Riemann surfaces with an effective even theta characteristic. The Schottky problem in this case becomes that of recognizing $\mathcal{J}_g \cap \theta_{\text{null}}$ inside θ_{null} .

The first observation is that a vanishing theta null is automatically a singular point of the theta divisor, because the partial derivatives $\partial\theta[m]/\partial z_i$ are odd. Hence one is led to study the local structure of Θ_τ around the singular point, and the first natural invariant is the *rank of the quadric tangent cone*, which corresponds to the rank of the Hessian matrix of θ at the theta null. In particular, if a Jacobian has a vanishing theta null, then the quadric tangent cone has rank at most 3. Hence

$$\mathcal{J}_g \cap \theta_{\text{null}} \subseteq \theta_{\text{null}}^3, \quad (17)$$

where we denote by θ_{null}^3 the locus of abelian varieties with a vanishing theta null whose quadric tangent cone has rank at most 3. Conversely, Grushevsky and Salvati Manni [2008] proved that this inclusion is actually an equality in genus 4, confirming a conjecture of Farkas. In the same paper, they ask

whether $\mathcal{J}_g \cap \theta_{\text{null}}$ is an irreducible component of θ_{null}^3 in higher genera, which would imply a solution to the weak Schottky problem for Jacobians with a vanishing theta null. The main result of our companion paper [Agostini and Chua 2019] is an affirmative answer in genus 5.

Theorem 5.1 [Agostini and Chua 2019]. *In genus 5, the locus $\mathcal{J}_5 \cap \theta_{\text{null}}$ is an irreducible component of θ_{null}^3 .*

We observe that the containment $\tau \in \theta_{\text{null}}^3$ can be checked explicitly. Indeed, the condition of having an even two-torsion point in the theta divisor can be checked by evaluating the finitely many theta constants $\theta[m](0, \tau)$, and then numerically computing the rank of the Hessian matrix. We present such a computation here, which is also in [Agostini and Chua 2019]. From the example in Section 4A, we use the function `schottky_null`(τ) in `Theta.jl`. The output gives the even characteristic

$$m = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \tag{18}$$

where the theta constant vanishes. The output also gives the corresponding Hessian matrix

$$\begin{pmatrix} -2.79665 + 5.29764i & -9.57825 - 9.04671i & 7.36305 + 2.28697i & 7.58338 + 5.34729i & 6.15667 - 1.90199i \\ -9.57825 - 9.04671i & 18.9738 + 8.34582i & -23.1027 - 3.10545i & -9.31944 - 0.822821i & 0.524289 - 3.64991i \\ 7.36305 + 2.28697i & -23.1027 - 3.10545i & 16.8441 - 1.15986i & 13.9363 - 4.56541i & -3.32248 + 4.10698i \\ 7.58338 + 5.34729i & -9.31944 - 0.822821i & 13.9363 - 4.56541i & 2.89309 + 1.21773i & 3.86617 - 0.546202i \\ 6.15667 - 1.90199i & 0.524289 - 3.64991i & -3.32248 + 4.10698i & 3.86617 - 0.546202i & -12.9726 - 1.928i \end{pmatrix}.$$

The Hessian has the eigenvalues

$$\begin{aligned} &47.946229109152995 + 9.491932144035298i, \\ &-15.491689246713147 + 3.3401255907497958i, \\ &-9.512858919129267 - 1.0587349322052013i, \\ &-2.7271385943272036 \times 10^{-15} - 1.1117459994936022i \times 10^{-14}, \\ &-5.698014266322794 \times 10^{-15} + 6.342925068807627i \times 10^{-15}, \end{aligned}$$

so it has numerical rank 3 as expected.

ACKNOWLEDGMENTS. We are grateful to Bernd Sturmfels for the suggestion to study the Schottky problem in genus 5, and for his continuous encouragement. We thank Paul Breiding, Gavril Farkas, Jörg Frauendiener, Sam Grushevsky, Christian Klein, Riccardo Salvati Manni, Andrey Soldatenkov, Sasha Timme and Sandro Verra for useful comments and discussions. We thank the anonymous referees for their suggestions, which improved the quality of the paper and of the code. This project was initiated at the Max Planck Institute for Mathematics in the Sciences in Leipzig, which both authors would like to thank for the hospitality and support at various stages of this work.

SUPPLEMENT. The online supplement contains version 1.0 of `Theta.jl`.

REFERENCES.

[Accola 1983] R. D. M. Accola, “On defining equations for the Jacobian locus in genus five”, *Proc. Amer. Math. Soc.* **89**:3 (1983), 445–448. MR

- [Agostini and Améndola 2019] D. Agostini and C. Améndola, “Discrete Gaussian distributions via theta functions”, *SIAM J. Appl. Algebra Geom.* **3**:1 (2019), 1–30. MR
- [Agostini and Chua 2019] D. Agostini and L. Chua, “On the Schottky problem for genus five Jacobians with a vanishing theta null”, 2019. To appear in *Ann. Scuola Norm. Sci.* arXiv
- [Agostini et al. 2020] D. Agostini, T. Çelik, J. Struwe, and B. Sturmfels, “Theta surfaces”, *Vietnam J. Math.* (2020).
- [Bezanson et al. 2017] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: a fresh approach to numerical computing”, *SIAM Rev.* **59**:1 (2017), 65–98. MR
- [Birkenhake and Lange 2004] C. Birkenhake and H. Lange, *Complex abelian varieties*, 2nd ed., Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences] **302**, Springer, 2004. MR
- [Bruin et al. 2019] N. Bruin, J. Sijssling, and A. Zotine, “Riemann matrices and endomorphism rings of algebraic Riemann surfaces”, 2019, http://doc.sagemath.org/html/en/reference/curves/sage/schemes/riemann_surfaces/riemann_surface.html.
- [Chua et al. 2019] L. Chua, M. Kummer, and B. Sturmfels, “Schottky algorithms: classical meets tropical”, *Math. Comp.* **88**:319 (2019), 2541–2558. MR
- [Deconinck et al. 2004] B. Deconinck, M. Heil, A. Bobenko, M. van Hoeij, and M. Schmies, “Computing Riemann theta functions”, *Math. Comp.* **73**:247 (2004), 1417–1442. MR
- [Eichler and Zagier 1985] M. Eichler and D. Zagier, *The theory of Jacobi forms*, Progress in Mathematics **55**, Birkhäuser, Boston, 1985. MR
- [Farkas et al. 2021] H. M. Farkas, S. Grushevsky, and R. S. Manni, “An explicit solution to the weak Schottky problem”, *Algebr. Geom.* **8**:3 (2021), 358–373. MR
- [Frauendiener et al. 2019] J. Frauendiener, C. Jaber, and C. Klein, “Efficient computation of multidimensional theta functions”, *J. Geom. Phys.* **141** (2019), 147–158. MR
- [Gaudry 2007] P. Gaudry, “Fast genus 2 arithmetic based on theta functions”, *J. Math. Cryptol.* **1**:3 (2007), 243–265. MR
- [Grushevsky 2012] S. Grushevsky, “The Schottky problem”, pp. 129–164 in *Current developments in algebraic geometry*, edited by M. M. Lucia Caporaso, James McKernan and M. Popa, Math. Sci. Res. Inst. Publ. **59**, Cambridge Univ. Press, 2012. MR
- [Grushevsky and Manni 2008] S. Grushevsky and R. S. Manni, “Jacobians with a vanishing theta-null in genus 4”, *Israel J. Math.* **164** (2008), 303–315. MR
- [Igusa 1981] J.-i. Igusa, “On the irreducibility of Schottky’s divisor”, *J. Fac. Sci. Univ. Tokyo Sect. IA Math.* **28**:3 (1981), 531–545. MR
- [Krichever and Shiota 2013] I. Krichever and T. Shiota, “Soliton equations and the Riemann–Schottky problem”, pp. 205–258 in *Handbook of moduli, II*, edited by G. Farkas and I. Morrison, Adv. Lect. Math. (ALM) **25**, International Press, Somerville, MA, 2013. MR
- [Lenstra et al. 1982] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, “Factoring polynomials with rational coefficients”, *Math. Ann.* **261**:4 (1982), 515–534. MR
- [Mumford 2007] D. Mumford, *Tata lectures on theta, I*, Birkhäuser, Boston, 2007. MR
- [Regev and Stephens-Davidowitz 2017] O. Regev and N. Stephens-Davidowitz, “An inequality for Gaussians on lattices”, *SIAM J. Discrete Math.* **31**:2 (2017), 749–757. MR
- [SageMath] The Sage Developers, “SageMath, the Sage Mathematics Software System (Version 8.6)”, <https://www.sagemath.org>.
- [Schnorr and Euchner 1994] C.-P. Schnorr and M. Euchner, “Lattice basis reduction: improved practical algorithms and solving subset sum problems”, *Math. Programming* **66**:2, Ser. A (1994), 181–199. MR
- [Segur 2008] H. Segur, “Integrable models of waves in shallow water”, pp. 345–371 in *Probability, geometry and integrable systems*, edited by M. Pinsky and B. Birnir, Math. Sci. Res. Inst. Publ. **55**, Cambridge Univ. Press, 2008. MR
- [Swierczewski and Deconinck 2016] C. Swierczewski and B. Deconinck, “Computing Riemann theta functions in Sage with applications”, *Math. Comput. Simulation* **127** (2016), 263–272. MR
- [Zhang et al. 2012] W. Zhang, S. Qiao, and Y. Wei, “HKZ and Minkowski reduction algorithms for lattice-reduction-aided MIMO detection”, *IEEE Trans. Signal Process.* **60**:11 (2012), 5963–5976. MR

RECEIVED: 21 Feb 2020

REVISED: 21 Oct 2020

ACCEPTED: 9 Nov 2020

DANIELE AGOSTINI:

daniele.agostini@mis.mpg.de

Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, Germany

LYNN CHUA:

lchua@caltech.edu

Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, United States

