Efficient computation of multidimensional theta functions

Jörg Frauendiener, Carine Jaber, Christian Klein

EFFICIENT COMPUTATION OF MULTIDIMENSIONAL THETA FUNCTIONS

JÖRG FRAUENDIENER, CARINE JABER, AND CHRISTIAN KLEIN

Abstract. An important step in the efficient computation of multi-dimensional theta functions is the construction of appropriate symplectic transformations for a given Riemann matrix assuring a rapid convergence of the theta series. An algorithm is presented to approximately map the Riemann matrix to the Siegel fundamental domain. The shortest vector of the lattice generated by the Riemann matrix is identified exactly, and the algorithm ensures that its length is larger than $\sqrt{3}/2$. The approach is based on a previous algorithm by Deconinck et al. using the LLL algorithm for lattice reductions. Here, the LLL algorithm is replaced by exact Minkowski reductions for small genus and an exact identification of the shortest lattice vector for larger values of the genus.

1. Introduction

Multidimensional theta functions are important in many fields of mathematics and in applications, for instance in the theory of integrable partial differential equations (PDEs), see e.g. [2], in conformal field theories and cryptography. Since they can be seen as the main building block of meromorphic functions on Riemann surfaces, they appear naturally where Riemann surfaces and algebraic curves are of importance. They are conveniently defined as a multi-dimensional series,

$$\Theta_{\mathrm{pq}}(\mathrm{z},\mathbb{B}) = \sum_{\mathrm{N}\in\mathbb{Z}^g} \exp\left\{\mathrm{i}\pi\left\langle \mathbb{B}\left(\mathrm{N}+\mathrm{p}\right),\mathrm{N}+\mathrm{p}\right\rangle + 2\pi\mathrm{i}\left\langle \mathrm{z}+\mathrm{q},\mathrm{N}+\mathrm{p}\right\rangle\right\}\ , \tag{1}$$

with $\mathrm{z}\in\mathbb{C}^g$ and the *characteristics* $\mathrm{p},\mathrm{q}\in\mathbb{R}^g$, where $\langle\cdot,\cdot\rangle$ denotes the Euclidean scalar product $\langle\mathrm{N},\mathrm{z}\rangle = \sum_{i=1}^g N_i z_i$. The matrix $\mathbb{B} = X + \mathrm{i}Y$ is a *Riemann matrix*, i.e., it is symmetric and has a positive definite imaginary part Y. The latter property ensures that the series (1) converges uniformly for all z, and that the theta function with characteristics is an entire function of $\mathrm{z}\in\mathbb{C}^g$.

The goal of the present article is the description of an efficient treatment of theta functions appearing in the context of integrable PDEs such as the Korteweg-de Vries (KdV) and nonlinear Schrödinger (NLS) equations. Quasiperiodic solutions to these PDEs were given at the beginning of the 1970s by Novikov, Dubrovin, Its, Matveev, van Moerbeke, Krichever and others in terms of such theta functions on compact Riemann surfaces, see [2, 7] and references therein for a historic account. To study such solutions numerically, theta functions of the form (1) have to be evaluated for a given Riemann matrix and characteristics for many values of the argument $\mathrm{z}\in\mathbb{C}^g$. To this end, efficient numerical tools to evaluate theta functions are needed. The `algcurves` package distributed with Maple originally due to Deconinck and van Hoeij, see [4, 6], has integrated theta functions as outlined in [5] starting from Maple 7. Note that the `algcurves` package is currently being transferred to a new platform [39]. A purely numerical approach to compact Riemann surfaces via algebraic curves was given in [10–15], see [22, 23] for applications to integrable PDEs. For a review of the current state of the art of computational approaches to Riemann surfaces, the reader is referred to [3].

The basic idea of the algorithms for theta functions in [3] is to approximate the expression (1) via a truncated series[1]. Obviously the convergence of the series (1) depends on the bilinear term, more precisely on the *shortest vector* N_{min} of the lattice $\mathbb{Z}^g$ equipped with the inner product

[1] For alternative approaches based on arithmetic-geometric means and Newton iterations, see [8, 28], which were developed for studying modular functions, i.e., the dependence of theta functions with z = 0 on the Riemann matrix; in practice these approaches appear to be mainly interesting if precisions of several thousand bits are needed.

defined by the imaginary part Y of the Riemann matrix $\mathbb{B}$: $\langle \mathrm{N}, \mathrm{M}\rangle_Y := \langle Y\mathrm{N}, \mathrm{M}\rangle$, $\mathrm{N}, \mathrm{M} \in \mathbb{Z}^g$. For a given Riemann matrix the shortest vector N_{min} is then defined in terms of its squared length

$$y_{min} = \langle \mathrm{N}_{min}, \mathrm{N}_{min}\rangle_Y := \min_{\mathrm{N}\in\mathbb{Z}^g/\{0\}} \langle Y\mathrm{N}, \mathrm{N}\rangle \,. \tag{2}$$

Obviously, the longer the shortest vector, the more rapid the convergence of the theta series. Changing the shortest vector can be achieved by changing the homology basis of the underlying Riemann surface which yields a different but symplectically equivalent Riemann matrix. This can be achieved by using modular transformations, i.e., symplectic transformations with integer coefficients to generate larger norms of the shortest vector in order to accelerate the convergence of a theta series for given $\mathbb{B}$. Since the behavior of theta functions under modular transformations is explicitly known, such transformations can dramatically increase the rate of convergence which is especially important for larger values of g. This approach was for the first time implemented in an algorithm by Deconinck et. al. in [5].

The main task in this context is the identification of the shortest vector in a given g-dimensional lattice known as the shortest vector problem (SVP). Currently, there is no known algorithm that would solve this problem in polynomial time. The LLL algorithm [30] yields an approximation to the shortest vector in polynomial time but with an error growing exponentially with the dimension g (though in practice slowly with g such that it can be used for small genus as an approximation). For these reason, in [5] the SVP was solved approximately via the LLL algorithm. However, since we are interested in an evaluation of theta functions in a large number of points, it can be beneficial to identify the shortest vector exactly even for small g. Though it is computationally demanding this knowledge will accelerate the ensuing evaluation of the theta function (2). Therefore, we replace the LLL algorithm in [5] with an exact Minkowski reduction for $g \leq 5$, and with an exact solution to the SVP for higher genus.

The paper is organized as follows: in section 2 we summarize mathematical facts about symplectic transformations, theta functions, and Siegel's fundamental domain. In section 3 we review various notions of lattice reductions and discuss the algorithms used for the LLL and Minkowski reduction. In section 4 we present Siegel's algorithm to ensure that the imaginary part of the transformed Riemann matrix has a shortest lattice vector of squared length greater than $\sqrt{3}/2$ and discuss examples. We add some concluding remarks in section 5.

2. Theta functions and symplectic transformations

In this section we summarize important properties of multi-dimensional theta functions and symplectic transformations. In particular we are interested in the behavior of theta functions under symplectic transformations and in the Siegel fundamental domain.

2.1. Symplectic transformations.

A Riemann matrix $\mathbb{B} = X + \mathrm{i}Y$ lies in $\mathbb{H}^g$, the Siegel space of symmetric $g \times g$ matrices with positive definite imaginary part Y and real part X. Riemann matrices are only unique up to modular transformations, i.e., transformations $\mathcal{A}_g \in Sp(2g, \mathbb{Z})$, the symplectic or modular group, of the form

$$\mathcal{A}_g = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \tag{3}$$

where A, B, C, D are $g \times g$ integer matrices satisfying

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^T \begin{pmatrix} 0_g & I_g \\ -I_g & 0_g \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 0_g & I_g \\ -I_g & 0_g \end{pmatrix}; \tag{4}$$

here 0_g and I_g are the $g \times g$ null and identity matrix respectively. The Riemann matrix transforms under these modular transformations $\mathcal{A}_g$ (3) as

$$\mathbb{H}^g \mapsto \mathbb{H}^g : \quad \mathbb{B} \mapsto \tilde{\mathbb{B}} = (A\mathbb{B} + B)(C\mathbb{B} + D)^{-1}. \tag{5}$$

Siegel [38] gave the following fundamental domain for the modular group in which each Riemann surface characterized by its Riemann matrix $\mathbb{B}$ corresponds to exactly one point:

Definition 2.1. *Siegel's fundamental domain is the subset of $\mathbb{H}^g$ such that $\mathbb{B} = X + \mathrm{i}Y \in \mathbb{H}^g$ satisfies:*

(1) $|X_{nm}| \leq 1/2$, *$n, m = 1, \ldots, g$,*
(2) *Y is in the fundamental region of Minkowski reductions,*
(3) *$|\det(C\mathbb{B} + D)| \geq 1$ for all C, D (4).*

Roughly speaking, the three conditions address different parts of the modular transformation in (3). The first condition in Def. 2.1 fixes the matrices B in (3). The second condition refers to Minkowski reductions [32,33] and fixes the unimodular matrices A in (3). Minkowski reductions of the lattice generated by a positive definite matrix Y are equivalent to the introduction of a reduced lattice basis, i.e., a collection of vectors of minimal length which can be extended to a basis of the lattice. This condition will be discussed in more detail in the following section. The third condition in Def. 2.1 fixes the matrices C, D in (3). Since $|\det \mathbb{B}|$ can be viewed as the 'height' of $\mathbb{B}$ (see [26]), this can be seen as a condition of maximal height.

In genus 1, the above conditions 2.1 give the well known elliptic fundamental domain,

$$|X| \leq 1/2, \quad X^2 + Y^2 \geq 1. \tag{6}$$

Note that parts of the boundary of the fundamental domain in Def. 2.1 have to be excluded in order to assure that no two different points on the boundary can be related by a symplectic transformation. We will not address this problem in this paper since we are mainly interested in the convergence of the theta series.

Siegel [38] showed that the third of the conditions in Def. 2.1 is equivalent to a finite number of conditions, i.e., just a finite number of matrices C and D has to be considered. But it is not known how to efficiently obtain this finite set of matrices. The only case $g > 1$ where this has been achieved appears to be Gottschling's work [18] for genus 2. In this case, the fundamental domain is defined by the following set of inequalities: the standard limits for the real part of the Riemann matrix,

$$|X_{11}| \leq \frac{1}{2}, \quad |X_{12}| \leq \frac{1}{2}, \quad |X_{22}| \leq \frac{1}{2},$$

the Minkowski ordering conditions:

$$Y_{22} \geq Y_{11} \geq 2Y_{12} \geq 0, \tag{7}$$

and the following set of 19 inequalities corresponding to the third condition in Def. 2.1:

$$|\mathbb{B}_{11}| \geq 1, \quad |\mathbb{B}_{22}| \geq 1, \quad |\mathbb{B}_{11} + \mathbb{B}_{22} - 2\mathbb{B}_{12} + \mathbf{e}| \geq 1, \tag{8}$$

where $\mathbf{e} = \pm 1$, and

$$|\det(\mathbb{B} + S)| \geq 1, \tag{9}$$

where S are the matrices

$$\begin{gathered} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{e} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & 0 \\ 0 & \mathbf{e} \end{pmatrix}, \\ \begin{pmatrix} \mathbf{e} & 0 \\ 0 & -\mathbf{e} \end{pmatrix}, \quad \begin{pmatrix} 0 & \mathbf{e} \\ \mathbf{e} & 0 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{e} & \mathbf{e} \\ \mathbf{e} & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & \mathbf{e} \\ \mathbf{e} & \mathbf{e} \end{pmatrix}. \end{gathered} \tag{10}$$

These conditions are important if modular functions expressed in terms of theta functions are studied, see for instance [8,24,25,36] and references therein.

2.2. **Theta functions.** Theta functions with characteristics are defined as an infinite series (1). A characteristic is called *singular* if the corresponding theta function vanishes identically. Of special interest are half-integer characteristics with $2\mathrm{p}, 2\mathrm{q} \in \mathbb{Z}^g$. Such a half-integer characteristic is called *even* if $4\langle \mathrm{p}, \mathrm{q}\rangle = 0 \bmod 2$ and *odd* otherwise. It can be easily shown that theta functions with odd (even) characteristic are odd (even) functions of the argument z. The theta function with characteristic is related to the Riemann theta function Θ, the theta function with zero characteristic $\Theta := \Theta_{00}$, via

$$\Theta_{\mathrm{pq}}(\mathrm{z}, \mathbb{B}) = \Theta(\mathrm{z} + \mathbb{B}\mathrm{p} + \mathrm{q}) \exp\left\{ \mathrm{i}\pi \left\langle \mathbb{B}\mathrm{p}, \mathrm{p} \right\rangle + 2\pi \mathrm{i} \left\langle \mathrm{p}, \mathrm{z} + \mathrm{q} \right\rangle \right\} . \tag{11}$$

From its definition, a theta function has the periodicity properties

$$\Theta_{\mathrm{pq}}(\mathrm{z} + \mathrm{e}_j) = e^{2\pi \mathrm{i} p_j} \Theta_{\mathrm{pq}}(\mathrm{z}) \ , \quad \Theta_{\mathrm{pq}}(\mathrm{z} + \mathbb{B}\mathrm{e}_j) = e^{-2\pi \mathrm{i}(z_j + q_j) - \mathrm{i}\pi B_{jj}} \Theta_{\mathrm{pq}}(\mathrm{z}) \ , \tag{12}$$

where e_j is a vector in $\mathbb{R}^g$ consisting of zeros except for a 1 in j$^{\text{th}}$ position. These periodicity properties (12) can be conveniently used in the computation of the theta function: an arbitrary vector $\mathrm{z} \in \mathbb{C}^g$ can be written in the form $\mathrm{z} = \hat{\mathrm{z}} + \mathrm{N} + \mathbb{B}\mathrm{M}$ with $\mathrm{N}, \mathrm{M} \in \mathbb{Z}^g$, where $\hat{\mathrm{z}} = \mathbb{B}\hat{\mathrm{p}} + \hat{\mathrm{q}}$ with $|\hat{p}_i| \leq 1/2$, $|\hat{q}_i| \leq 1/2$. Thus, it is enough to compute the theta function for arguments $\hat{\mathrm{z}}$ lying in the fundamental domain of the Jacobian, i.e., $\mathbb{C}^g/\Lambda$, where Λ is the period lattice[2] formed by $\mathbb{B}$ and the g-dimensional identity matrix, $\hat{\mathrm{z}} = \mathbb{B}\hat{\mathrm{p}} + \hat{\mathrm{q}}$ with $|\hat{p}_i| \leq 1/2$, $|\hat{q}_i| \leq 1/2$. For general arguments z one computes $\Theta(\hat{\mathrm{z}}, \mathbb{B})$ and obtains $\Theta(\mathrm{z}, \mathbb{B})$ from the periodicity properties (12) by multiplying with an appropriate exponential factor.

To compute the series (11), it will be approximated by a sum, $|N_i| \leq \mathcal{N}_\epsilon$, $i = 1, \dots, g$, where the constant $\mathcal{N}_\epsilon$ is chosen such that all omitted terms in (1) are smaller than some prescribed value of ϵ. Since we work in double precision, we typically choose $\epsilon = 10^{-16}$, i.e., of the order of the smallest difference between floating point numbers that can be handled in Matlab. Note that in contrast to [5], we do not give a specific bound for each N_i, $i = 1, \dots, g$, i.e., we sum over a g-dimensional sphere instead of an ellipsoid. The reason for this is that it does not add much to the computational cost, but that it simplifies a parallelization of the computation of the theta function in which we are interested. Taking into account that we can choose z in the fundamental domain of the Jacobian because of (12), we get with (2) for the Riemann theta function the estimate

$$\mathcal{N}_\epsilon > \sqrt{-\frac{\ln \epsilon}{\pi y_{min}}} + \frac{1}{2}. \tag{13}$$

Thus the greater the norm of the shortest lattice vector, the more rapid will be the convergence of the theta series.

The action of the modular group on theta functions is known, see for instance [2, 9, 34]. One has

$$\Theta_{\tilde{\mathrm{p}}\tilde{\mathrm{q}}}(\mathcal{M}^{-1}\mathrm{z}, \tilde{\mathbb{B}}) = k\sqrt{\det(\mathcal{M})} \exp\left(\frac{1}{2}\sum_{i\leq j} z_i z_j \frac{\partial}{\partial \mathbb{B}_{ij}} \ln\det\mathcal{M}\right) \Theta_{\mathrm{pq}}(\mathrm{z}), \tag{14}$$

where $\tilde{\mathbb{B}}$ is given by (5), where k is a constant with respect to z, and where

$$\mathcal{M} = C\mathbb{B} + D, \quad \begin{pmatrix} \tilde{\mathrm{p}} \\ \tilde{\mathrm{q}} \end{pmatrix} = \begin{pmatrix} D & -C \\ -B & A \end{pmatrix} \begin{pmatrix} \mathrm{p} \\ \mathrm{q} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mathrm{diag}(CD^T) \\ \mathrm{diag}(AB^T) \end{pmatrix}, \tag{15}$$

where diag denotes the diagonal of the matrices AB^T and CD^T.

3. Lattices reductions

In this section we address the second condition in the definition 2.1 of Siegel's fundamental domain, the Minkowski reduction of the imaginary part of the Riemann matrix. This classical problem is related to Euclidean lattices in g dimensions and the search for efficient bases for them, i.e., a basis consisting of shortest possible lattice vectors. Finding such a basis is known as *lattice reduction*. Below we will summarize basic facts on lattices and their reductions and give a brief review of approximative and exact approaches to lattice reductions.

3.1. Lattices and lattice reductions.

In the context of the present paper, we are concerned with lattices, i.e., discrete additive subgroups of $\mathbb{R}^g$, of the form

$$\mathcal{L}(t_1, \dots, t_g) = \left\{ TN \,\middle|\, N \in \mathbb{Z}^g \right\}, \tag{16}$$

where $T = [t_1, t_2, \dots, t_g] \in \mathbb{R}^{g\times g}$ has rank g. Thus, the lattice consists of all linear combinations with integer coefficients of the g linearly independent vectors t_i, the columns of T. The vectors t_i form the *lattice basis*. Different lattice bases are related via unimodular transformations, $\tilde{T} = TA$ where A is an integer matrix with $|\det(A)| = 1$. This implies in particular that $\det(T^T T)$ is an invariant of such unimodular transformations. The length of a lattice vector t_i, $i = 1, \dots, g$, is given by its Euclidean norm $||t_i||^2 = \sum_{j=1}^g T_{ji}^2$.

[2] Note, that this lattice Λ is not to be confused with the lattice generated by the matrix Y discussed in the present paper.

In the context of lattices there are two equivalent points of view. The first one, which we took above, is to consider the lattice as being generated by a basis T, and then define the matrix $Y = T^T T$ of inner products of the basis vectors. The complementary point of view is to take a lattice always as being represented as $\mathbb{Z}^g$ but equipped with a positive definite bilinear form Y defining the lengths and angles of the lattice vectors. The two view points are both useful and we can switch between them quite easily: given the matrix T of basis vectors the bilinear form is represented by $Y = T^T T$, and given a symmetric positive definite matrix Y we can obtain a matrix T representing a lattice basis by a Cholesky decomposition, which yields an upper triangular matrix R with $Y = R^T R$.

It is well known that in $\mathbb{R}^g$ it is always possible to introduce an orthonormal basis, and the Gram-Schmidt procedure allows to determine such a basis from a given general one. In the discrete case of a lattice, in general there will be no basis consisting of orthogonal vectors, and there is a lower bound on the length of vectors in a lattice. A basis is considered *reduced* if it satisfies certain conditions. In general, the goal is to find a basis of vectors of minimal length with a minimal deviation from orthogonality. Thus, important issues in lattice theory are the shortest vector problem (SVP), i.e., the determination of the shortest non-zero lattice vector, and the closest vector problem (CVP), i.e., the location of the lattice vector closest to a given point $\mathbf{x} \in \mathbb{R}^g$, see for instance [19] for a recent review.

The strongest known lattice reduction is *Minkowski reduction*: a basis for a lattice $\mathcal{L}$ generated by a matrix Y is Minkowski reduced if it consists of shortest lattice vectors which can be extended to a basis of $\mathcal{L}$.

In a more narrow sense, a symmetric and positive definite matrix is Minkowski reduced if it satisfies Minkowski's conditions,

$$||x_1 t_1 + \ldots + x_g t_g|| \geq ||t_i||, \tag{17}$$

for all $1 \leq i \leq g$ and for all integers $x_1, \ldots, x_g$ such that $\gcd(x_1, \ldots, x_g) = 1$. A minimal set of these conditions was given by Minkowski [32, 33] for $g \leq 4$. For $g = 5, 6, 7$ these conditions were presented in [41]. Note that the number of Minkowski reduction conditions grows rapidly with the dimension of the lattice, for $g = 7$ there are 90 000 conditions in [41]. The corresponding conditions do not appear to be known for $g > 7$.

The *simple Minkowski reduction* is achieved by the additional condition $Y_{i,i+1} > 0$ for $i = 1, \ldots, g-1$, which fixes the orientation of the vectors.

Remark 3.1. *For $g = 2$, the Minkowski fundamental domain (condition 2 of Def. 2.1), i.e., the fundamental domain of the unimodular group, is given by the simple Minkowski reduction. But for $g > 2$ the simple Minkowski reduction does not define the Minkowski fundamental domain, see [40]. The fundamental domain for $g = 3$ is given in [40], but the corresponding conditions in higher dimensions appear to be unknown.*

The condition that the set of shortest lattice vectors have to form a basis of the lattice is problematic from an algorithmic point of view, in addition the Minkowski conditions are not known for $g > 7$. Therefore the lattice reduction by Hermite [21], Khorkine and Zolotareff [27] (HKZ) is generally preferred in applications for $g \geq 7$: in this reduction, the shortest lattice vector is identified and a unimodular transformation is found such that this vector is used as the first basis vector. This means, it appears as $\tilde{T}_{i1}$, the first column in the transformed matrix $\tilde{T}$; next, the shortest vector of the $(g-1) \times (g-1)$ dimensional matrix $\tilde{T}_{i,j}$, $i, j = 2, \ldots, g$ is identified, a unimodular transformation to put this vector as the first of the transformed $(g-1) \times (g-1)$ dimensional matrix; then an SVP is solved for the $(g-2) \times (g-2)$ dimensional matrix obtained after taking off this vector and the first line of the matrix and so on.

Since both Minkowski and HKZ reduction require the solution of SVPs for which no algorithms in polynomial time are known, see [19], the same applies to these reductions. Therefore, the LLL algorithm [30] is often used since it converges in polynomial time. It essentially applies Gauss' algorithm [17, 29] in dimension 2 to pairs of vectors in higher dimensions, see below. The problem with the LLL algorithm is that it approaches the solution of the SVP with an error that grows exponentially with the dimension of the lattice, see the examples below and in the next section.

Note that all above mentioned reductions, Minkowski, HKZ and LLL lead to the same result in dimension $g = 2$ where they agree with the result of the Gauss algorithm: the two shortest lattice vectors are identified there.

3.2. **Algorithms.** An important step in most lattice reduction algorithms is the SVP or more generally, the CVP. One distinguishes exact algorithms to find the respective vectors, which typically find the vectors by more or less sophisticated enumeration of all possible vectors, approximative algorithms, or probabilistic algorithms of Monte-Carlo type, see [19] for a review. It turns out that for $g \leq 40$, enumerative algorithms can be more efficient than probabilistic ones. Since we are mainly interested in the case of small genus ($g \leq 20$), we concentrate here on the former. As before, we always put $Y = T^T T$ where without loss of generality we may assume that T is an upper triangular matrix formed by vectors $t_1, \dots, t_g$.

Gauss reduction. Gauss reduction provides an algorithm to find the Minkowski and HKZ reduced form of a two-dimensional lattice formed by two vectors t_1 and t_2. The algorithm identifies the two shortest vectors in this lattice. Size reductions motivated by Gram-Schmidt type formulae and swapping of vectors are alternated in this algorithm until it terminates. A basis is *size reduced* if condition (7) is satisfied.

- In the size reduction step, one puts

$$\tilde{t}_2 = t_2 - [\mu + 1/2] t_1, \quad \mu = \frac{\langle t_1, t_2 \rangle}{||t_1||^2}, \tag{18}$$

  i.e., a linear combination of the two vectors with the rounded Gram-Schmidt factor μ.
- If the resulting vector $\tilde{t}_2$ is shorter than t_1, the vectors are swapped, and a further size reduction step follows. The algorithm terminates when the vector $\tilde{t}_2$ of the size reduction step (18) is longer than t_1.

LLL reduction. The LLL algorithm essentially generalizes Gauss' algorithm to higher dimensions than 2. A parameter δ is chosen such that $1/4 < \delta \leq 1$ (the algorithm is not polynomial in time for $\delta = 1$). The Gram-Schmidt vectors t_k^* and matrix $\mu_{i,k}$, $i, k = 1, \dots, g$ given by

$$t_i^* = t_i - \sum_{j=1}^{i-1} \mu_{i,j} t_j^*, \quad \mu_{i,k} = \frac{\langle t_i, t_k^* \rangle}{||t_i^*||^2}$$

are computed. If the LLL condition

$$||t_k^*||^2 \geq (\delta - \mu_{k,k-1}^2) ||t_{k-1}^*||^2 \tag{19}$$

is not satisfied for some k (starting with $k = 2$), the reduction step of the Gauss algorithm and a possible swap of the vectors are applied to this pair of vectors. The Gram-Schmidt matrix is updated, and if t_{k-1}^* has changed, the algorithm continues with k replaced by $k - 1$. Otherwise one passes to the pair of vectors t_{k+1}^*, t_k^*. Reductions and swaps of pairs of vectors are continued until the LLL condition (19) is satisfied for all $k = 1, \dots, g$.

Sphere decoding. The basic idea of sphere decoding algorithms is that for the CVP for a general point $\mathrm{x} \in \mathbb{R}^g$, all lattice points z inside a sphere of radius ρ centered at x are enumerated (for treating the SVP, one takes $\mathrm{x} = 0$ excluding the lattice point $\mathrm{z} = 0$ as a possible solution of the problem).

- To obtain an estimate for the radius ρ, we choose the norm of the first vector. This is not the optimal choice for ρ, see for instance [1], but it guarantees that the algorithm will always find a vector with length ρ or shorter.
- The problem is then to find lattice vectors z satisfying $||T\mathrm{z} - \mathrm{x}|| < \rho$. Since T is upper triangular this inequality becomes

$$(T_{gg} z_g - x_g)^2 + (T_{g-1,g-1} z_{g-1} + T_{g-1,g} z_g - x_{g-1})^2 + \ldots < \rho^2. \tag{20}$$

  In particular this implies

$$|z_g| < |x_g \pm \rho| / T_{gg}. \tag{21}$$

Equation (20) suggests a recursive implementation of the algorithm, see [35]: the integers within the above limits are enumerated as possible candidates for z_g; for each such possible component z_g a CVP in dimension g is equivalent to a CVP in dimension $g-1$. Thus, a CVP in dimension g can be reduced to a finite number (corresponding to the possible choices for z_g) of CVPs in dimension $g-1$. The search process starts at level g and goes down recursively to level 1 to solve a one-dimensional problem.

- On each level, the enumeration of the integer candidates for the component z_g which are restricted by (21) uses the strategy by Schnorr and Euchner [37], i.e., one starts with $z_g = n_0 := [x_g/T_{gg}]$ and then in a zig-zag approach $z_g = n_0 \pm 1, n_0 \pm 2, \dots$ are explored until the limits of (21) are reached.
- All vectors with a length larger than ρ are rejected. If a vector with a length smaller than ρ is found, the radius ρ is updated with this smaller length and the procedure is continued with the new value. This is done until the shortest nontrivial vector t_{min} is identified.
- When used in the context of Minkowski reduction, only vectors with mutually prime entries are considered.
- A unimodular matrix $\mathcal{M}$ is constructed such that this vector is the first in the matrix $T\mathcal{M}$, i.e., $\mathcal{M}^{-1}t_{min} = e_1$, where $e_1 = (1, 0, \dots, 0)^T$.

Minkowski reduction. The idea of the algorithm [42] for a Minkowski reduction is to apply the above SVP algorithm successively to a lattice in order to find a set of shortest lattice vectors. In the first step, an SVP is solved and a unimodular matrix is identified such that this vector appears as the first vector of the transformed matrix as above. At the p^{th} step, we have a basis $B_p = \{\tilde{t}_1, \dots, \tilde{t}_{p-1}, t_p, \dots, t_g\}$. To extend this basis to a Minkowski reduced basis, the p^{th} reduced basis vector must satisfy:

$$||\tilde{t}_p|| = \min\{||T\mathrm{z}|| : \mathrm{z} \in \mathbb{Z}^g, \gcd(z_p, \dots, z_g) = 1\}.$$

The gcd condition is directly implemented in the SVP algorithm. To extend $\{\tilde{t}_1, \dots, \tilde{t}_p\}$ to a basis, one has to find a unimodular matrix Z such that $B_{p+1} = B_p Z$, i.e., a unimodular matrix which does not affect the first $p-1$ vectors, and which has z in pth position, where $\tilde{t}_p = B_p\mathrm{z}$.

For $1 \leq i \leq g$, the i-th Minkowski's successive minimum is defined as the radius of the smallest closed ball centered at the origin containing at least i linearly independent lattice vectors. Note that Minkowski showed that in dimensions $g > 4$, the vectors realizing Minkowski's successive minima may not form a lattice basis. Thus a Minkowski reduction algorithm based on SVPs only can fail. The construction of an appropriate unimodular matrix is thus crucial, see also [20]. Note that the corresponding reduced matrix will in general not satisfy the Minkowski reduction conditions which are not even explicitly known for $g > 7$. Since we are here mainly interested in the shortest lattice vector, we do not explore Minkowksi reductions for $g > 5$.

3.3. **Examples.** To illustrate the difference between an LLL and a Minkowski reduced basis, we consider a symmetric real 4×4 matrix (for the ease of representation, we only give 4 digits throughout the paper)

```
Y =

0.7563    0.4850    0.4806    0.3846
0.4850    1.3631    0.2669   -0.3084
0.4806    0.2669    0.7784   -0.4523
0.3846   -0.3084   -0.4523    1.7538.
```

This matrix was created using a 4×4 matrix L with random entries, and then setting $Y = L^T L$. As usual we put $Y = T^T T$ where the upper triangular matrix T is obtained from Y via a Cholesky decomposition.

To this matrix T, we apply the algorithm for Minkowski reductions discussed above based on a successive finding of shortest lattice vectors. The found matrix $\tilde{Y} = \tilde{T}^T \tilde{T}$ is then postprocessed to ensure the simple Minkowski reduction condition $\tilde{Y}_{i,i+1} \geq 0$, $i = 1, 2, 3$, i.e., a unimodular matrix $\tilde{Z}$ is constructed such that $\tilde{Z}^T \tilde{Y} \tilde{Z}$ has positive elements in the right parallel to the diagonal. This leads to the matrix

```
    0.5321    0.2058   -0.1639    0.0181
    0.2058    0.5735    0.0920    0.2634
   -0.1639    0.0920    0.5741    0.1364
    0.0181    0.2634    0.1364    0.6535.
```

In particular it can be seen that the squared length of the shortest lattice vector is 0.5321, the (11) element of the matrix.

An LLL reduction with $\delta = 3/4$ of the matrix Y leads to

```
    0.7563   -0.2757    0.3182   -0.1089
   -0.2757    0.5735    0.0920    0.2634
    0.3182    0.0920    0.5741    0.1364
   -0.1089    0.2634    0.1364    0.6535.
```

The length of the shortest vector identified by the LLL algorithm is in this example 0.5735, thus longer than the shortest vector of the lattice which is still 0.5321 since both the LLL and the Minkowski reduction of a lattice are obtained via unimodular transformations. The latter obviously do not change the length of the shortest vector. Thus, this example shows that the LLL algorithm can lead to a considerable overestimation of the length of the shortest vector even for small size of the matrix. The effect is known to grow exponentially with the size of the matrix.

Note that in the above example, the shortest vector appears as the second vector in contrast to a Minkowski ordered matrix where the shortest vector is always the first.

Remark 3.2. *An LLL reduced matrix is always ordered in accordance with the LLL condition (19). Thus there is no reason why the shortest vector should appear in the first position as in Minkowski reduced matrices. This is especially important in the context of the Siegel algorithm to be discussed in the following section, where the shortest vector is always assumed to be the first of the matrix.*

To illustrate this aspect even more, we consider another example of a random matrix,

```
    Y =

    1.7472    0.5191    1.0260    0.6713
    0.5191    1.3471    0.2216   -0.5122
    1.0260    0.2216    0.6801    0.4419
    0.6713   -0.5122    0.4419    0.7246.
```

The Minkowski reduction yields

```
    0.2205    0.0443    0.0342    0.0351
    0.0443    0.3636    0.1660   -0.0294
    0.0342    0.1660    0.3688    0.1516
    0.0351   -0.0294    0.1516    0.3753.
```

The corresponding LLL reduced matrix ($\delta = 3/4$) takes the form

```
    0.3753    0.0294   -0.1516    0.0351
    0.0294    0.3636    0.1660   -0.0443
   -0.1516    0.1660    0.3688   -0.0342
    0.0351   -0.0443   -0.0342    0.2205.
```

In this case the shortest vector is found by the LLL algorithm in contrast to the previous example, but it appears as the last vector. The first vector has with 0.3753 almost twice the length of the shortest vector, 0.2205.

4. Approximation to the Siegel fundamental domain

In this section, we review an algorithm due to Siegel [38] to approximate the Siegel fundamental domain, which has been implemented together with the LLL algorithm in [5]. This algorithm is used here together with an exact determination of the shortest lattice vector. As an example we consider the Fricke-Macbeath curve [16, 31], a curve of genus 7 with the maximal number of automorphisms.

4.1. **Siegel's algorithm.** Whereas Siegel's fundamental domain as defined in Def. 2.1 is an important theoretical concept in symplectic geometry, its practical relevance is limited since no constructive approach exists to actually identify the domain for $g > 2$: the first condition on the components of the matrix of the real part X of $\mathbb{B}$ is straight forward. But as discussed in the previous section, already the Minkowski fundamental domain appearing in the second condition of Def. 2.1 is only known for $g \leq 3$. The third condition of Def. 2.1 is, however, the least studied one. Siegel [38] showed that it is equivalent to a finite number of conditions, but these conditions, except for the classical case $g = 1$ in (6), are only known for $g = 2$ in the form (8) and (9) due to Gottschling [18].

However, Siegel [38] gave an algorithm to approximately reach the fundamental domain. He proved the following

Theorem 4.1. *Any Riemann matrix $\mathbb{B} = X + \mathrm{i}Y \in \mathbb{H}^g$ with real and imaginary part X respectively Y can be transformed by a symplectic transformation (3–5) to a Riemann matrix satisfying the following conditions:*

1. $|X_{nm}| \leq 1/2$, *for $n, m = 1...g$,*
2. *the squared length of the shortest lattice vector of the lattice generated by Y is greater than or equal to $\sqrt{3}/2$,*

The proof in [38], see also [5], is constructive and leads naturally to an algorithm:

Proof. As already mentioned in the previous section, the first condition can be always achieved by an appropriate choice of the matrix B in (3), $B = [X]$, i.e., each component of B is the integer part of the corresponding component of X.

For the second condition, we assume that the shortest vector of the lattice generated by T, where T is the Cholesky decomposition of $Y = T^T T$, is the first vector of T. It is discussed in the previous section that this can be always achieved. Siegel showed that the determinants of the imaginary part of two Riemann matrices $\tilde{\mathbb{B}} = \tilde{X} + \mathrm{i}\tilde{Y}$ and $\mathbb{B} = X + \mathrm{i}Y$ related by a symplectic transformation (5) satisfy

$$|\det(\tilde{Y})| = \frac{|\det(Y)|}{|\det(C\mathbb{B} + D)|^2}. \tag{22}$$

If one considers the *quasi-inversion*

$$A = \begin{pmatrix} 0 & \mathbf{0}_{g-1}^T \\ \mathbf{0}_{g-1} & \mathbf{1}_{g-1,g-1} \end{pmatrix}, \quad B = \begin{pmatrix} -1 & \mathbf{0}_{g-1}^T \\ \mathbf{0}_{g-1} & \mathbf{0}_{g-1,g-1} \end{pmatrix},$$
$$C = \begin{pmatrix} 1 & \mathbf{0}_{g-1}^T \\ \mathbf{0}_{g-1} & \mathbf{0}_{g-1,g-1} \end{pmatrix}, \quad D = \begin{pmatrix} 0 & \mathbf{0}_{g-1}^T \\ \mathbf{0}_{g-1} & \mathbf{1}_{g-1,g-1} \end{pmatrix}, \tag{23}$$

where $\mathbf{0}_{g-1}$ is the column vector of $g - 1$ zeros, equation (22) takes the form

$$|\det(\tilde{Y})| = \frac{|\det(Y)|}{|\mathbb{B}_{11}|^2}. \tag{24}$$

This leads to the following algorithm:

1. choose A in (3) such that the shortest lattice vector appears as the first vector of T;
2. choose B in (3) such that the real part of $\hat{\mathbb{B}} = A^T \mathbb{B} A$ has components $|\hat{X}_{nm}| \leq 1/2$, for $n, m = 1, \dots, g$;
3. if $|\hat{\mathbb{B}}_{11}| \geq 1$, terminate the algorithm; if not, apply the quasi-inversion (23) and continue with step 1 of the algorithm for the resulting $\mathbb{B}$.

Because of (24) the modulus of the determinant of the imaginary part of the transformed Riemann matrix increases with each application of step (3). Since Siegel [38] has shown that there exists only a finite number of symplectic transformations leading to increasing $|\det(Y)|$ and that this determinant will be eventually greater than or equal to 1, the algorithm terminates after a finite number of steps. Then Y_{11} is the squared length of the shortest lattice vector by construction. Since we have $|\mathbb{B}_{11}| \geq 1$, this implies $Y_{11}^2 + X_{11}^2 \geq 1$. Since $|X_{11}| \leq 1/2$, one has $Y_{11} \geq \sqrt{3}/2$. This proves the theorem. □

Remark 4.2. *The fact that the squared length of the shortest vector y_{min} of the lattice generated by Y is always greater than $\sqrt{3}/2$ implies that a general estimate can be given for the cutoff $\mathcal{N}_\epsilon$ in (13). For an $\epsilon = 2.2 \times 10^{-16}$, the smallest difference between two floating point numbers that Matlab can represent, we find $\mathcal{N}_\epsilon \approx 4.1$. This means that with an $\mathcal{N}_\epsilon = 4$ the neglected terms in the theta series (1) will be of the order of 10^{-14}, the order of the rounding errors, and smaller. For longer shortest vectors, even smaller values of $\mathcal{N}_\epsilon$ are possible.*

Note that the algorithm [5] implemented in Maple uses the LLL algorithm on Y instead of an exact determination of the shortest lattice vector. As discussed in the previous section and illustrated below, this is considerably more rapid than an exact determination of the vector, but can lead to exponentially (in the dimension g) growing errors in this context. Since the convergence of the theta series is directly related to the shortest vector, we opt here for an exactly determined shortest vector. If the LLL algorithm is applied, the length of the shortest vector is only approximately identified (with an error growing exponentially with the dimension). Thus the the cutoff $\mathcal{N}_\epsilon$ (13) has to be based on an estimate of the length of the shortest vector which is not provided by the algorithm.

Remark 4.3. *In this article, we use the cutoff (13) for all lattice vectors appearing in the theta sum (1), because the summation over a g-dimensional sphere can be more easily parallelized. If a summation over an ellipsoid as in [5] is applied, a different cutoff can be used for each lattice vector. In this case a full Minkowski reduction will be beneficial, whereas in our case, the exact determination of the shortest lattice vector is sufficient.*

4.2. **Example.** As an example we want to study the Riemann matrix of the Fricke-Macbeath surface [16,31], a surface of genus $g = 7$ with the maximal number $84(g-1) = 504$ of automorphisms. It can be defined via the algebraic curve

$$f(x,y) := 1 + 7yx + 21y^2x^2 + 35x^3y^3 + 28x^4y^4 + 2x^7 + 2y^7 = 0. \tag{25}$$

The code [15] produces for this curve the following Riemann matrix[3]

```
RieMat =

  Columns 1 through 4

   1.0409 + 1.3005i   0.0530 + 0.3624i   0.3484 + 0.0000i   0.2077 + 0.6759i
   0.0530 + 0.3624i  -0.5636 + 1.0753i   0.0187 - 0.5975i   0.6749 + 0.3001i
   0.3484 + 0.0000i   0.0187 - 0.5975i   1.0544 + 1.7911i   0.3220 - 1.0297i
   0.2077 + 0.6759i   0.6749 + 0.3001i   0.3220 - 1.0297i  -0.0978 + 1.7041i
  -0.2091 - 0.2873i   0.1220 - 0.5274i   0.3029 + 0.8379i  -0.7329 - 0.8055i
  -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.2297 - 0.3668i  -0.0714 - 0.1766i
   0.3590 + 0.5023i   0.1990 - 0.1118i   0.3495 - 0.0499i  -0.0415 + 0.5448i

  Columns 5 through 7

  -0.2091 - 0.2873i  -0.1064 - 0.4257i   0.3590 + 0.5023i
   0.1220 - 0.5274i   0.1205 - 0.1783i   0.1990 - 0.1118i
   0.3029 + 0.8379i  -0.2297 - 0.3668i   0.3495 - 0.0499i
  -0.7329 - 0.8055i  -0.0714 - 0.1766i  -0.0415 + 0.5448i
   1.1824 + 1.0163i   0.4425 + 0.2592i   0.0835 - 0.2430i
   0.4425 + 0.2592i   0.2815 + 0.7791i  -0.6316 - 0.0369i
   0.0835 - 0.2430i  -0.6316 - 0.0369i   0.2315 + 0.6895i.
```

Remark 4.4. *Since we work with finite precision, rounding is an issue also in the context of lattice reductions. The code [15] generally produces results with a tolerance Tol between 10^{-10} and 10^{-14}, which appears for instance in the form of an asymmetry of the computed Riemann matrix of the*

[3] For the ease of the reader, we present only 4 digits though the Riemann matrix is computed with an error of the order of 10^{-10}.

order of Tol. Since in lattice reductions the components of the Riemann matrix are multiplied with integers, these errors will be amplified. Thus a rounding of an order of magnitude larger than Tol is necessary in practice.

After LLL reduction the first basis vector of the lattice is found to have squared norm 1.3005 i.e., the (11) component of the imaginary part of the above Riemann matrix. Note that the lattice basis is almost LLL reduced, there are only minor effects of the LLL algorithm applied to this matrix. Since the norm of the shortest vector is greater than $\sqrt{3}/2$, no quasi-inversion is applied. An ensuing shift of the real part leads to the matrix

```
W =

  Columns 1 through 4

   0.0409 + 1.3005i   0.0530 + 0.3624i  -0.4849 - 0.6245i  -0.1064 - 0.4257i
   0.0530 + 0.3624i   0.4364 + 1.0753i  -0.3594 - 0.6598i   0.1205 - 0.1783i
  -0.4849 - 0.6245i  -0.3594 - 0.6598i  -0.4706 + 1.3844i  -0.1946 - 0.1178i
  -0.1064 - 0.4257i   0.1205 - 0.1783i  -0.1946 - 0.1178i   0.2815 + 0.7791i
   0.3590 + 0.5023i   0.1990 - 0.1118i  -0.0510 - 0.0073i   0.3684 - 0.0369i
  -0.4511 + 0.1383i  -0.0171 + 0.2485i  -0.0543 - 0.3239i   0.3907 - 0.1531i
   0.2684 - 0.2975i  -0.4161 + 0.2521i   0.0481 + 0.3949i  -0.2437 - 0.3094i

  Columns 5 through 7

   0.3590 + 0.5023i  -0.4511 + 0.1383i   0.2684 - 0.2975i
   0.1990 - 0.1118i  -0.0171 + 0.2485i  -0.4161 + 0.2521i
  -0.0510 - 0.0073i  -0.0543 - 0.3239i   0.0481 + 0.3949i
   0.3684 - 0.0369i   0.3907 - 0.1531i  -0.2437 - 0.3094i
   0.2315 + 0.6895i   0.3656 - 0.1563i  -0.2134 - 0.1308i
   0.3656 - 0.1563i  -0.4318 + 0.6585i  -0.1541 + 0.0260i
  -0.2134 - 0.1308i  -0.1541 + 0.0260i  -0.4997 + 1.0021i.
```

However, the square of the norm of the shortest lattice vector of the imaginary part of the matrix W is 0.6585, well below the threshold $\sqrt{3}/2$. This shows once more the limitations of the LLL algorithm since the convergence of the theta series we are interested in is controlled by the length of the shortest lattice vector. Note that the LLL reduced $\tilde{Y}$ above has the shortest vector in the 6th column (with squared norm 0.6585). One could construct a unimodular matrix Z such that $T * Z$ has this vector appearing in the first column (the resulting matrix might not satisfy the LLL condition (19)). This would be more suited to the application of Siegel's algorithm, but will be still approximate since in general LLL does not identify the shortest lattice vector correctly.

If the same algorithm is applied with an exact determination of the shortest vector, the picture changes considerably: in the first step of the iteration, the shortest lattice vector is correctly identified having the square of the norm 0.6585. Thus after a shift of the real part, a quasi-inversion is applied. The subsequent identification of the shortest vector of the resulting matrix leads to a vector of squared norm 0.7259. After a shift of the real part, another quasi-inversion is applied. This time the square of the norm of the shortest vector is 1.0211 and thus greater than $\sqrt{3}/2$. After a shift of the real part we finally obtain

```
W =

  Columns 1 through 4

   0.3967 + 1.0211i   0.0615 - 0.1322i  -0.0000 + 0.0000i  -0.4609 - 0.2609i
   0.0615 - 0.1322i   0.3967 + 1.0211i   0.3553 - 0.5828i  -0.3386 + 0.1933i
  -0.0000 + 0.0000i   0.3553 - 0.5828i   0.2894 + 1.1656i   0.0905 + 0.2450i
  -0.4609 - 0.2609i  -0.3386 + 0.1933i   0.0905 + 0.2450i   0.3967 + 1.0211i
   0.3553 - 0.5828i   0.4776 - 0.1287i  -0.4776 + 0.1287i  -0.4776 + 0.1287i
   0.1838 + 0.3219i   0.2743 + 0.5669i   0.3871 - 0.3736i   0.0167 - 0.3895i
  -0.3386 + 0.1933i  -0.3386 + 0.1933i  -0.1223 - 0.4541i   0.0615 - 0.1322i

  Columns 5 through 7

   0.3553 - 0.5828i   0.1838 + 0.3219i  -0.3386 + 0.1933i
   0.4776 - 0.1287i   0.2743 + 0.5669i  -0.3386 + 0.1933i
  -0.4776 + 0.1287i   0.3871 - 0.3736i  -0.1223 - 0.4541i
  -0.4776 + 0.1287i   0.0167 - 0.3895i   0.0615 - 0.1322i
   0.2894 + 1.1656i  -0.1671 - 0.7115i   0.0905 + 0.2450i
  -0.1671 - 0.7115i   0.4414 + 1.2784i  -0.3386 + 0.1933i
   0.0905 + 0.2450i  -0.3386 + 0.1933i   0.3967 + 1.0211i.
```

In contrast to the algorithm incorporating LLL reductions, the squared length of the shortest vector of the imaginary part is here given by the (11) component of the matrix W. Note that the approximate character of the LLL algorithm is unsatisfactory for our purposes for two reasons: First the overestimation of the length of the shortest vector leads to a premature end of the algorithm and a much shorter shortest vector than necessary. But secondly the potentially crude approximation of its length implies that an estimate of the truncation parameter $\mathcal{N}_\epsilon$ in (13) based on the LLL result could be misleading with the consequence of a loss of accuracy in the approximation of the theta function.

Matlab timings have to be taken with a grain of salt since they depend crucially on the coding, in particular on how many precompiled commands could be used. Still in applications it is important to know how long a certain task takes on a given computer. For the above example, the LLL code is not very efficient, but converges in roughly 1 ms. The SVP code takes in this case 4-5 times longer, which is still completely negligible compared to what can be gained by applying the above algorithm in the computation of a theta function associated to this surface.

The above example is in fact typical. If we consider an example of even higher genus, the curve

$$f(x,y) := y^9 + 2x^2y^6 + 2x^4y^3 + x^6 + y^2 = 0 \tag{26}$$

of genus 16, we find a similar behavior. Using Siegel's algorithm on the Riemann matrix for this curve computed with the code [15], we find that the variant with the LLL algorithm converges within three iterations. The LLL algorithm takes 1-2ms in each step. The algorithm produces $\mathbb{B}_{11} = 0.3314+1.0188i$, a value clearly larger than 1. The length of the shortest vector generated by the imaginary part of this Riemann matrix as found via SVP is 0.4437, well below the theoretical minimum of $\sqrt{3}/2 \approx 0.866$. On the other hand Siegel's algorithm with an exact solution of the SVP in each step requires 14 iterations where each SVP takes around 10ms. Finally we get $\mathbb{B}_{11} = 0.4748 + 0.8956i$, i.e., a shortest vector almost twice as long as what has been found with the LLL algorithm.

5. Outlook

In this paper, we have shown that for Siegel's algorithm (theorem 4.1) can be used to efficiently compute multi-dimensional theta functions. For a genus $g > 2$, an exact determination of the shortest vector of the lattice generated by the imaginary part of the Riemann matrix is recommended. The approximative LLL algorithm is for $g < 20$ only an order of magnitude faster than the SVP algorithm, but finds the shortest vector merely with an error growing exponentially with g.

From the point of view of symplectic geometry, it would be interesting to find an algorithm to approach Siegel's fundamental domain (see definition 2.1) in a better way. This would allow to decide within numerical precision whether two different algebraic curves in fact define the same Riemann surface: a possible way to decide this would be to construct for both Riemann surfaces the symplectic transformations to the Siegel fundamental domain. If both Riemann matrices map to the same point in the fundamental domain, they correspond to the same surface. A problem in this context is that the Minkowski fundamental domain is only known for $g \leq 3$. Thus the case

$g = 3$ is the most promising to study in this context. Even less is known about the third condition in definition 2.1. It would be interesting to explore the matrices C and D there as Gottschling [18] did in genus 2 to compute $|\det(C\mathbb{B} + D)|$ at least for an interesting set of these matrices. The goal would be to approximate this maximal height condition better than with the quasi-inversion (23). This will be the subject of further work.

Acknowledgement:
We thanks D. Stehlé for helpful discussions and hints. JF thanks for the hospitality at the University of Burgundy as a visiting professor, where part of this work has been completed.

References

[1] Babai, L.: On Lovász Lattice reduction and the nearest lattice point problem, Combinatorica **6**, 1-13 (1986)

[2] Belokolos, E.D., Bobenko, A.I., Enolskii, V.Z., Its, A.R., Matveev, V.B.: Algebro-geometric approach to nonlinear integrable equations. Springer, Berlin (1994)

[3] Bobenko, A.I., Klein, C. (ed.), *Computational Approach to Riemann Surfaces*, Lect. Notes Math. **2013** (2011).

[4] Deconinck, B., van Hoeij, M.: Computing Riemann matrices of algebraic curves. Physica D **28**, 152–153 (2001)

[5] Deconinck, B., Heil, M., Bobenko, A., van Hoeij, M., Schmies, M.: Computing Riemann theta functions. Mathematics of Computation, **73**, 1417–1442 (2004)

[6] Deconinck, B., and Patterson, M., in Bobenko, A.I., Klein, C. (ed.), *Computational Approach to Riemann Surfaces*, Lect. Notes Math. **2013** (2011).

[7] Dubrovin, B.A.: Theta functions and non-linear equations, Usp. Mat. Nauk **36**, No. 2, 11–80 (1981) (English translation: Russ. Math. Surv. **36**, No. 2, 11–92 (1981)).

[8] Dupond, R.: Fast evaluation of modular functions using Newton iterations and the AGM, Math. Comp. **80**, 1823-1847 (2011).

[9] Fay, J.D.: Theta functions on Riemann surfaces. Lect. Notes in Math., **352**, Springer (1973)

[10] Frauendiener, J., Klein, C.: On the exact treatment of stationary counter-rotating dust disks: Physical Properties. Phys. Rev. D, **63**, 84025 (2001)

[11] Frauendiener, J., Klein, C.: Hyperelliptic theta-functions and spectral methods. J. Comp. Appl. Math., **167**, 193 (2004)

[12] Frauendiener, J., Klein, C.: Hyperelliptic theta-functions and spectral methods: KdV and KP solutions, Lett. Math. Phys., **76**, 249-267 (2006)

[13] Frauendiener, J., Klein, C.: *Algebraic curves and Riemann surfaces in Matlab*, in A. Bobenko and C. Klein (ed.), *Computational Approach to Riemann Surfaces*, Lecture Notes in Mathematics Vol. **2013** (Springer) (2011).

[14] Frauendiener, J., Klein, C.: *Computational approach to hyperelliptic Riemann surfaces*, Lett. Math. Phys. 105(3), 379-400, DOI 10.1007/s11005-015-0743-4 (2015).

[15] Frauendiener, J., Klein, C.: *Computational approach to compact Riemann surfaces*, Nonlinearity 30(1), 138 (2016).

[16] Fricke, R., Über eine einfache Gruppe von 504 Operationen, Mathematische Annalen, 52 (23): 321–339, (1899)

[17] Gauss, C.F.: Disquisitiones Arithmetics, (Springer-Verlag, 1801)

[18] Gottschling, E.: Explizite Bestimmung der Randflächen des Fundamentalbereiches der Modulgruppe zweiten Grades. Math. Ann. **138** 103-124 (1959)

[19] Hanrot, G., Pujol X., and Stehlé, D.: Algorithms for the Shortest and Closest Lattice Vector Problems, Invited contribution for IWCC'11. (Springer, 2011)

[20] Helfrich, B.: Algorithms to construct Minkowski reduced and Hermite reduced lattice bases, Theory comput. Sci., **41**, no. 2-3, 125-139 (1985)

[21] Hermite, C.: Extraits de lettres de M. Hermite á M. Jacobi sur différents objets de la théorie des nombres, J. Reine. Angew. Math., Vol. 40, 279-290 (1850)

[22] Kalla, C., Klein, C.: On the numerical evaluation of algebro-geometric solutions to integrable equations, Nonlinearity **25** 569-596 (2012).

[23] Kalla, C., Klein, C.: Computation of the topological type of a real Riemann surface, Math. Comp. 83, 1823-1846, dx.doi.org/10.1090/S0025-5718-2014-02817-2 (2014).

[24] Klein, C., Kokotov, A., Korotkin, D.: Extremal properties of the determinant of the Laplacian in the Bergman metric on the moduli space of genus two Riemann surfaces. Math. Zeitschr. **261**(1), 73–108 (2009)

[25] Klein, C., Richter, O.: Ernst Equation and Riemann Surfaces. Lecture Notes in Physics **685**, Springer, Berlin (2005)

[26] Klingen, H.: Introductory Lectures on Siegel Modular Forms, (Cambridge University Press, Cambridge, 1990).

[27] Korkine, A., Zolotareff, G.: Sur les formes quadratiques, Math. Ann., Vol. 6, 366-389 (1873)

[28] Labrande, H.: Explicit computation of the Abel-Jacobi map and its inverse, PhD thesis (University of Lorraine, 2016)

[29] Lagrange, J.-L.: Recherches d'arithmétique, Nouveaux Mémoires de l'Académie de Berlin (1773)

[30] Lenstra, A.K., Lenstra, H.W., Jr., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann, **261**, 515–534 (1982).

[31] Macbeath, A., On a curve of genus 7, Proceedings of the London Mathematical Society **15**, 527–542, (1965)

[32] Minkowski, H.: Über die positiven quadratischen Formen und über kettenbruchähnliche Algorithmen, J. Reine und Angewandte Math., 107, 278-297.

[33] Minkowski, H.: Gesammelte Abhandlungen 1, pp. 145-148, 153-156, 217-218 (Leipzig-Berlin: Teubner 1911).

[34] Mumford, D.: Tata Lectures on Theta. I and II., Progress in Mathematics, 28 and 43, respectively. Birkhäuser Boston, Inc., Boston, MA, (1983 and 1984).

[35] Pohst, M.: On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications, ACM SIGSAM Bulletin, Vol. 15, 37-44, (1981)

[36] Quine, J.R., Sarnak, P. (ed): Extremal Riemann surfaces. Contemporary Mathematics, **201** AMS (1997)

[37] Schnorr, C.P. and Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems Mathematical Programming, **66**, 181-191 (1994)

[38] Siegel, C.L.: Topics in complex function theory. Vol. III. John Wiley & Sons, Inc., New York, 1989.

[39] Swierczewski, C., Deconinck, B.: Computing Riemann theta functions in Sage with applications, Mathematics and Computers in Simulation, ISSN 0378-4754, http://dx.doi.org/10.1016/j.matcom.2013.04.018 (2013).

[40] Tammela, P.P.: On the theory of reduction of positive quadratic forms, Dokl. AN SSSR **209** no 6, 1299-1302 (1973).

[41] Tammela, P.P.: The Minkowski's fundamental reduction domain of positive quadratic forms in seven variables. Zapiski nauc. Sem. Leningrad. Otd. mat. Inst. Steklov **67**, 108-143 (1977).

[42] Zhang, W., Qiao, S., Wei, Y.: HKZ and Minkowski Reduction Algorithms for Lattice-Reduction-Aided MIMO Detection, IEEE Trans- actions on SIGNAL Processing, **60**(11) (2012)

(J. Frauendiener) Department of Mathematics and Statistics, University of Otago, P.O. Box 56, Dunedin 9054, New Zealand
Email address: `joergf@maths.otago.ac.nz`

(C. Jaber) Institut de Mathématiques de Bourgogne, Université de Bourgogne, 9 avenue Alain Savary, BP 47970, 21078 Dijon Cedex, France
Email address: `carine.jaber@u-bourgogne.fr`

(C. Klein) Institut de Mathématiques de Bourgogne, Université de Bourgogne, 9 avenue Alain Savary, BP 47970, 21078 Dijon Cedex, France
Email address: `christian.klein@u-bourgogne.fr`