BASIC USAGE OF `extremal_rays` IN SAGEMATH

IGNACIO BARROS, LAURE FLAPAN, AND RICCARDO ZUFFETTI

SUMMARY

In [BFZ25] we illustrated two different methods to check whether the rays generated by primitive Heegner divisors are extremal in the pseudoeffective cone of an orthogonal modular variety. Both methods are implemented in the SageMath code **`extremal_rays`**, available on the authors' webpages. The purpose of the present note is to explain how these methods have been implemented and double-checked.

Last update: April 24, 2025.

CONTENTS

1. INTRODUCTION

Throughout this note, we use the same notation as in [BFZ25]. Let $\mathcal{F}_\Lambda$ be the orthogonal Shimura variety arising from an even lattice Λ of signature $(2,n)$. Let $\mu \in D(\Lambda)$ and let $m \in \mathbb{Z} - q(\mu)$. To check whether the primitive Heegner divisor $P_{-m,\mu}$ is extremal in the pseudoeffective cone $\overline{\mathrm{Eff}}(\mathcal{F}_\Lambda)$, we showed it is enough to check that

$$\deg(\varphi^* P_{-m,\mu}) < 0 \qquad \text{or equivalently} \qquad \mathrm{vol}(P^2_{-m,\mu}) < 0. \tag{1.1}$$

In the above, $\varphi\colon \mathcal{F}_L \to \mathcal{F}_\Lambda$ is the finite map of orthogonal modular varieties induced by the inclusion of the sublattice $L := \rho^\perp$ in Λ, where ρ is a primitive representative for $(-m,\mu)$. For details, see [BFZ25, Corollary 3.3 and Lemma 3.12].

Both quantities $\deg(\varphi^* P_{-m,\mu})$ and $\mathrm{vol}(P^2_{-m,\mu})$ may be explicitly computed (up to the volume of some modular variety) in terms of Fourier coefficients of modular forms. The former can be bounded as $\deg(\varphi^* P_{-m,\mu}) \le \deg(\varphi^* H_{-m,\mu})$, where

$$\frac{\deg(\varphi^* H_{-m,\mu})}{\mathrm{vol}(\mathcal{F}_L)} = - \sum_{\substack{\alpha \in \Lambda / L \oplus K \\ t \in \mathbb{Z} - q(\alpha_L) \\ t \ge 0}} \theta\Big(\frac{m}{r^2} - t, \alpha_K\Big) c_{t,\alpha_L}\Big(E_{\frac{n+1}{2},L}\Big), \tag{1.2}$$

see [BFZ25, (3.8)] for a precise formula for $\deg(\varphi^* P_{-m,\mu})/\mathrm{vol}(\mathcal{F}_L)$, while the latter as

$$\frac{\mathrm{vol}(P^2_{-m,\mu})}{\mathrm{vol}(\mathcal{F}_\Lambda)} = g\big(m,\mu,E^k_{1,\Lambda}\big)^2 + 2 \sum_{\substack{r \in \mathbb{Z}_{>0} \\ r^2 \mid m}} \mu(r) \sum_{\substack{\alpha \in D(\Lambda) \\ r\alpha = \mu}} \sum_{\substack{w_1 \in \mathbb{Z}_{>0} \\ w_1^2 \mid (m/r^2)}} \sum_{\substack{\beta \in D(\Lambda) \\ w_1 \beta = \alpha}} c_{m/r^2 w_1^2, \beta}(E^k_{1,\Lambda}) \times \sum_{\substack{w_2 \in \mathbb{Z}_{>0} \\ \gcd(w_1,w_2)=1}} g\big(m,\mu,\mathcal{P}^\Lambda_{m w_2^2 / r^2 w_1^2, w_2 \beta}\big), \tag{1.3}$$

see [BFZ25, Proposition 3.13].

We implemented (1.2) and (1.3) in `extremal_rays`. Before describing the implementation, we outline pros and cons of both formulas.

Remark 1.1. We implemented (1.2) for Λ *unimodular* and for *moduli spaces of K3 surfaces*. Since the sum on the right-hand side of (1.2) is *finite*, this program certifies whether the degree of $\varphi^* H_{-m,\mu}$ (and hence of $\varphi^* P_{-m,\mu}$) is negative, and therefore whether the ray generated by $P_{-m,\mu}$ is extremal in $\overline{\mathrm{Eff}}(X)$. This certification led to the table of extremal rays for $\mathcal{F}_{2d}$ in [BFZ25, Table 1]. This method computes the exact value of $\deg(\varphi^* H_{-m,\mu})/\mathrm{vol}(\mathcal{F}_L)$ thereby giving a rigorous certification of extremality, but this computation depends on the choice of a sublattice $L \subset \Lambda$, which can be difficult to implement in Sage as it requires more specifics about the lattice Λ (hence why we only implemented it for Λ unimodular or the lattice associated to a K3 surface).

Remark 1.2. By contrast, the formula (1.3) does not depend on any choice of a sublattice $L \subset \Lambda$. In fact, our implementation in Sage works for any lattice Λ. Since (1.3) is given in terms of a sum of *infinitely many* Fourier coefficients of Poincaré series, the Sage computation of $\mathrm{vol}(P^2_{m,\mu})/\mathcal{F}_\Lambda$ is up to a fixed precision, which tells the program at which point to truncate the infinite sum. Nevertheless, it may be used to obtain empirical evidence of the negativity of $\mathrm{vol}(P^2_{m,\mu})$, and hence of the extremality of rays in the pseudoeffective cone of e.g. moduli spaces of projective hyperkähler varieties that are not covered by the first implemented method.

2. Basic usage

The program `extremal_rays` has been written using *SageMath Version 10.5*, released on December 4, 2024. To use the program, it is necessary to install the SageMath package `WeilRep` [Wil], which enables us to work with vector-valued modular forms with respect to the dual Weil representation ρ^*_Λ.

The first step to use the program is to launch it as `load("extremal_rays.sage")`.

2.1. **Pulling back Heegner divisors: The unimodular case.** We illustrate here the implementation of (1.2) in the case of Λ unimodular. Due to the classification of even unimodular lattices of signature $(2,n)$, Λ is of the form $\Lambda = U \oplus U \oplus E_8(-1)^{\oplus r}$ for some $r \geq 0$.

Let m be a positive integer, and $\rho \in U^{\oplus 2} \oplus E_8(-1)^{\oplus r}$ a primitive representative of $(-m, 0)$, that is, a primitive element such that $q(\rho) = -m$. Up to $O^+(\Lambda)$, we can assume that $\rho = e - mf$, where $\{e, f\}$ is the standard basis of the fist copy of U. Then $L = \rho^\perp = U \oplus A_1(m) \oplus E_8(-1)^{\oplus r}$ and $K = \mathbb{Z}\rho = A_1(-m)$. In this case, $\Lambda / L \oplus K$ has order $2m$, is cyclic and generated by

$$f = \left(\frac{e + mf}{2m}\right) - \left(\frac{e - mf}{2m}\right) = (\alpha, -\beta) \in D(L) \oplus D(K).$$

We call α and β the standard generators of $D(L) \oplus D(K)$. Let $k = \mathrm{rk}(\Lambda)$. For simplicity, we write P_{-m} and H_{-m} in place of respectively $P^\Lambda_{-m,0}$ and $H^\Lambda_{-m,0}$. Then

$$\varphi^* H_{-m} = \sum_{i=0}^{2m-1} \sum_{t \in \mathbb{Z} - q(i\alpha)} \theta(m - t, -i\beta) H^L_{-t, i\alpha}$$

and

$$\frac{\deg(\varphi^* P_{-m})}{\mathrm{vol}(\mathcal{F}_L)} \leq \frac{\deg(\varphi^* H_{-m})}{\mathrm{vol}(\mathcal{F}_L)} = -\theta(m, 0) + \sum_{i=0}^{2m-1} \sum_{\substack{t \in \mathbb{Z} - q(i\alpha) \\ t \neq 0}} \theta(m - t, -i\beta) \left| c_{t, i\alpha}(E_{L,k}) \right| \tag{2.1}$$

The command to compute (1.2) for Λ unimodular is

`deg_H(m,r)`.

The inputs are:

- `r`, a positive integer.
- `m`, a positive integer.

The output is the value of $\deg(\varphi^* H_{-\mathtt{m},0})/\mathrm{vol}(\mathcal{F}_L)$, with $\Lambda = U \oplus U \oplus E_8(-1)^{\oplus \mathtt{r}}$ and L as above, computed implementing the right-hand side of (2.1).

Example 2.1. Let $r = 4$ and $m = 1$, so that $\Lambda = U^{\oplus 2} \oplus E_8(-1)^{\oplus 4}$ and $L = U \oplus A_1(1) \oplus E_8(-1)^{\oplus 4}$. The command

```
deg_H(1,4)
```

gives `-149608934335/151628697551`, which is the value of $\deg(\varphi^* P_{-1,0})/\mathrm{vol}(\mathcal{F}_L)$. Since it is negative, the ray generated by $P_{-1,0}$ in $\overline{\mathrm{Eff}}(\mathcal{F}_\Lambda)$ is extremal. The command

```
deg_H(5,4)
```

gives `7375890180567886038104506603698965/192805859312945992490 71`, which is the value of $\deg(\varphi^* P_{-5,0})/\mathrm{vol}(\mathcal{F}_L)$. It is positive, and in fact it is easy to check (see also [BBFW25]) that the cone of Heegner divisors is generated by $P_{-1,0}$ and $P_{-2,0}$, while $P_{-5,0}$ lies in the interior. Therefore, $P_{-5,0}$ does not generate an extremal ray of (the cone of primitive Heegner divisors and hence of) the pseudoeffective cone of $\mathcal{F}_\Lambda$.

2.2. **Pulling back Heegner divisors: The K3 case.** We illustrate here the implementation of (1.2) in the case of moduli spaces of quasi-polarized K3 surfaces of degree $2d$, where $\Lambda = \Lambda_{2d}$. Recall that $\Lambda_{2d} := U^{\oplus 2} \oplus E_8(-1)^{\oplus 2} \oplus \mathbb{Z}\ell$, with $q(\ell) = -d$ and $\ell_* = \ell/2d$ the standard generator of $D(\Lambda_{2d})$.

We start with an algorithm to choose $\rho \in \Lambda_{2d}$ a primitive representative for $(-m, b\ell_*)$. After this we can compute $L = \rho^\perp$, $K = \mathbb{Z}\rho$ and decompose the classes $\alpha + b\ell_*$ in $D(L) \oplus D(K)$. This is done in [BBBF23, Section 3] for a class $h \in U^{\oplus 3} \oplus E_8(-1)^{\oplus 2} \oplus \langle -2(n-1) \rangle$ of fixed square and divisibility. Since there are enough copies of U, the argument is the same for Λ_{2d} replacing $n-1$ with d.

We keep the same notation as in [BBBF23, Section 3]. For a primitive element $\rho \in \Lambda_{2d}$, γ is its divisibility, which is the same as the order of ρ_* (the primitive multiple of ρ in $\Lambda_{2d}^\vee$) in $D(\Lambda_{2d})$. First note that

$$\gamma = \mathrm{ord}(b\ell_*) = \frac{2d}{\gcd(b, 2d)}$$

and if ρ is a primitive representative, then $q(\rho) = -m\gamma^2$. We have to find $0 \leq a < \gamma$ coprime with γ, and t an integer such that $\gamma(e + tf) - a\ell$ is in the same $\widetilde{\mathrm{O}}^+(L)$-orbit as ρ, in particular, we need

$$q(\gamma(e + tf) - a\ell) = -\gamma^2 m \quad \text{and} \quad \frac{a \cdot 2d}{\gamma} = b. \tag{2.2}$$

If $P_{-m,b\ell_*}$ is non-empty, that is, if there is a primitive representative ρ for $(-m, b\ell_*)$, then there is always a and t such that (2.2) holds. This is a restatement of [BBBF23, Lemma 3.4].

The first condition in (2.2) becomes

$$\gamma^2 m = a^2 d - \gamma^2 t. \tag{2.3}$$

This means that given (m, b, d) we can find integers γ, a and t as above if and only if $P_{-m,b\ell_*}$ is non-empty. There is a standard check in the SAGE program, when t is not an integer, then $P_{-m,b\ell_*}$ will be empty and the program returns "Bad entries: m is not in ZZ-q". From now on we assume $\rho = \gamma(e + tf) - a\ell$.

The orthogonal complement of ρ in $U \oplus \mathbb{Z}\ell$ is generated by

$$z_1 = \frac{2d \cdot a}{\gamma} f - \ell = bf - \ell \quad \text{and} \quad z_2 = e - tf.$$

Then $L = \rho^\perp = U \oplus E_8(-1)^{\oplus 2} \oplus Q$, where

$$Q = \begin{pmatrix} -2d & \frac{2d \cdot a}{\gamma} \\ \frac{2d \cdot a}{\gamma} & -2t \end{pmatrix} = \begin{pmatrix} -2d & b \\ b & -2t \end{pmatrix}.$$

This means that given (m, b, d) we can find L and natural generators for $D(L)$. Recall that $D_L = 4md$. Now for $\alpha \in \Lambda_{2d}/L \oplus K$ we have to decompose $\alpha + b\ell_*$ in $D(L) \oplus D(K)$. The rest of the algorithm involves computing $\theta(x, (\alpha + b\ell_*)_K)$ and $c_{t,(\alpha+b\ell_*)_L}$, both implemented in WeilRep [Wil].

Recall that $\Lambda_{2d}/L \oplus K$ is cyclic of order $2m\gamma$ generated by any $\alpha \in \Lambda_{2d}$ such that $\gamma = |\langle \alpha, \rho \rangle|$. In particular we can take $\alpha = f$ and sum with index $i = 0, \dots, 2\gamma m - 1$ over the classes $(if + b\ell_*) = \left(if + \frac{\rho}{\gamma}\right)$. Then as in the proof of [BFZ25, Theorem 3.7]

$$f = \underbrace{\left(f + \frac{1}{2\gamma m}\rho\right)}_{f_L} + \underbrace{\left(-\frac{1}{2\gamma m}\rho\right)}_{f_K}$$

and

$$if + b\ell_* = \left(if_L, if_K + \frac{\rho}{\gamma}\right) = \underbrace{\left(if + \frac{i}{2\gamma m}\rho\right)}_{(if+b\ell_*)_L} + \underbrace{\left(\frac{1}{\gamma} - \frac{i}{2\gamma m}\right)\rho}_{(if+b\ell_*)_K}.$$

Now, $g = \frac{\rho}{2\gamma^2 m}$ is a generator of $D(K)$ and

$$(z_1)_* = \frac{z_1}{\mathrm{div}_L(z_1)} = \frac{z_1}{2d/\gamma}, \quad (z_2)_* = \frac{z_2}{\mathrm{div}_L(z_2)} = \frac{z_2}{\gcd(2t, b)}$$

are generators for $D(L)$. These are the generators induced by the column vectors of the Gram matrix, so they are recover via `w.ds()`, where `w` is the Weil representation attached to L computed with WeilRep. Elements in `w.ds()` should be tuples of the form $(x/(2d/\gamma), y/\gcd(2t, b))$ corresponding to the element $x(z_1)_* + y(z_2)_* \in D(L)$. Then

$$\begin{aligned}(if + b\ell_*) &= \left(\frac{i \cdot a \cdot d}{\gamma^2 m}(z_1)_* + \frac{i \cdot \gcd(2t, b)}{2m}(z_2)_*, \gamma(2m - i)g\right) \\ &= \left(\frac{i \cdot a}{2\gamma m}z_1 + \frac{i}{2m}z_2, \frac{2m - i}{2\gamma m}\rho\right) \in D(L) \oplus D(K).\end{aligned}$$

The formula (1.2) boils then down to

(2.4)
$$\frac{\deg\left(\varphi^* H_{-m,b\ell_*}\right)}{\mathrm{vol}(\mathcal{F}_L)} = -\theta(m, b\ell_*) + \sum_{i=0}^{2\gamma m-1} \sum_{\substack{t \in \mathbb{Z} - q((if+b\ell_*)_L) \\ t \neq 0}} \theta(m - t, \gamma(2m - i)g) \left|c_{t,(if+b\ell_*)_L}\left(E_{k,L}\right)\right|.$$

The command to compute (1.2) in the case of moduli spaces of K3 surfaces is

`deg_Hb(m,b,d)`.

The inputs are:

- `b`, a positive integer.
- `m`, an element of $\mathbb{Z} - q(\mathtt{b}\ell_*)$.
- `d`, a positive integer.

The output is the value of $\deg(\varphi^* H_{-\mathtt{m},\mathtt{b}\ell_*})/\mathrm{vol}(\mathcal{F}_L)$, where $\Lambda = \Lambda_{2\mathtt{d}}$, computed implementing (2.4).

Example 2.2. Let $d = 3$, so that the Heegner divisors are on $\mathcal{F}_{\Lambda_6}$.

We compute $\deg(\varphi^* H_{-1/12,\ell_*})/\mathrm{vol}(\mathcal{F}_L)$ as

`deg_Hb(1/12,1,3)`.

The output is `-1`. Since it is negative and $\deg(\varphi^* P_{-1/12,\ell_*}) \leq \deg(\varphi^* H_{-1/12,\ell_*})$, then $P_{-1/12,\ell_*}$ generates an extremal ray in $\overline{\mathrm{Eff}}(\mathcal{F}_{\Lambda_6})$.

We compute $\deg(\varphi^* P_{-1/3,2\ell_*})/\mathrm{vol}(\mathcal{F}_L)$ as

`deg_Hb(1/3,2,3)`.

The output is `-1983/1984`. Since it is negative, $P_{-1/3,2\ell_*}$ generates an extremal ray in $\overline{\mathrm{Eff}}(\mathcal{F}_{\Lambda_6})$.

It has been certified in [BBFW25, Table 1] that $P_{-4/3,2\ell_*}$ does not generate an extremal ray of $\overline{\mathrm{Eff}}(\mathcal{F}_{\Lambda_6})$. In fact, if we compute $\deg(\varphi^* P_{-4/3,2\ell_*})/\mathrm{vol}(\mathcal{F}_L)$ as

`deg_Hb(4/3,2,3)`,

then the output is `692302366439/520093696`, which is positive.

2.3. **Pulling back Siegel Eisenstein series.** We illustrate here how to use the implementation of (1.3), then we describe a doublecheck of it in terms of the Fourier coefficients of $E^k_{2,\Lambda}$.

The command to compute (1.3) is
`vol_selfint_prim_Heegner(m,delta,S,up_to=20,nterms=50)`.
The inputs are:

- `S`, the Gram matrix of the lattice Λ.
- `delta`, an element of $D(\Lambda)$. It must be given as an element of the discriminant group computed with WeilRep, hence as an entry of the list `WeilRep(S).ws()`.
- `m`, a positive rational number in $\mathbb{Z} - q($`delta`$)$.
- `up_to`, a positive integer. In the implementation, the series $\sum_{w_2}$ appearing on the right-hand side of (1.3) is truncated up to such value. If this input is not given, the program runs with `up_to`$= 20$ by default.
- `nterms`, a positive integer. It is the value at which we truncate the series in the formula for the Fourier coefficients of the Poincaré series appearing in (1.3). It is the same as for the command `poincare_series` of WeilRep, see [Wil, README, Section 1.4.5].

The output is a floating-point number which approximates the value of $\mathrm{vol}(P^2_{-\mathtt{m},\mathtt{delta}})/\mathrm{vol}(\mathcal{F}_\Lambda)$. The greater the values of `up_to` and `nterms`, the more precise the output.

Example 2.3. Denote by `U` and `E8` the Gram matrix of the lattices U and E_8 respectively. The command
`S=block_diagonal_matrix(U, U, -E8, -E8, matrix([[-6]]), subdivide=false)`
gives the Gram matrix `S` of the K3 lattice Λ_6. The command
`D=WeilRep(S).ds()`
gives the list `D` of the elements of the discriminant group of Λ_6. The standard generator ℓ_* of $D(\Lambda_6)$ is
`D[len(D)-1]`.
The command
`vol_selfint_prim_Heegner(1/12,D[len(D)-1],S)`
computes an approximation of $\mathrm{vol}\big(P^2_{-1/12,\ell_*}\big)/\mathrm{vol}(\mathcal{F}_{\Lambda_6})$ with the default precision. The output is the floating number `-4.84929386352914e-9`, which is negative. This provide evidence that $P_{-1/12,\ell_*}$ generates an extremal ray of $\overline{\mathrm{Eff}}(\mathcal{F}_{\Lambda_6})$.
Similarly, the command
`vol_selfint_prim_Heegner(1/3,D[4],S)`
computes an approximation of $\mathrm{vol}\big(P^2_{-1/3,2\ell_*}\big)/\mathrm{vol}(\mathcal{F}_{\Lambda_6})$. The output is `-0.00253866837593326`, which is negative, hence provides evidence that $P_{-1/3,2\ell_*}$ is extremal in $\overline{\mathrm{Eff}}(\mathcal{F}_{\Lambda_6})$.
The extremality of the previous two divisors has been actually certified in Example 2.2.

Remark 2.4. All the outputs we wrote in [BFZ25, Tables 1 and 2] were verified with the method of Section 2.2 and the one of the present section. For a fixed primitive Heegner divisor, we checked that both outputs obtained with `deg_Hb` and `vol_selfint_prim_Heegner` have the same sign.

2.4. **A double-check.** We illustrate in this section a double-check of the implemented formula (1.3) to compute $\mathrm{vol}(P^2_{-m,\mu})$. This check depends on the computation of the Fourier coefficients of the Siegel Eisenstein series $E^k_{2,\Lambda}$, with $k := \mathrm{rk}\Lambda/2$. Currently, a formula for those coefficients is available in the literature only for Λ unimodular. Therefore, we assume in this section that Λ is unimodular. For simplicity, we write E^k_2, P_{-m} and H_{-m} in place of respectively $E^k_{2,\Lambda}$, $P_{-m,0}$ and $H_{-m,0}$.

We described in the proof of [BFZ25, Proposition 3.13] a formula for the volume of the intersection of two Heegner divisors on $\mathcal{F}_\Lambda$. Up to a factor, this volume is some Fourier coefficient of the restricted Siegel Eisenstein series $E_2^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)$, more precisely

$$\frac{\mathrm{vol}(H_{-m_1}\cdot H_{-m_2})}{\mathrm{vol}(\mathcal{F}_\Lambda)} = c_{(m_1,m_2)}\big(E_2^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big).$$

This can be computed in terms of elliptic Eisenstein and Poincaré series. If m is square-free, then $H_{-m} = P_{-m}$, and the formula for $c_{(m,m)}\big(E_2^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big)$ is the same as (1.3).

To verify that the implemented formula for $c_{(m,m)}\big(E_2^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big)$ is correct, we compute such Fourier coefficient in a different way as follows. Let

$$E_2^k(\tau) = \sum_{\substack{T\in\mathcal{S}_2\\ T\geq 0}} c_T(E_2^k) e(\mathrm{tr} T\tau)$$

denote the Fourier expansion of E_2^k. Then

$$c_{(m,m)}\big(E_1^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big) = \sum_{\substack{r\in\mathbb{Z}\\ |r|\leq 2m}} c_{\begin{pmatrix} m & r/2\\ r/2 & m\end{pmatrix}}(E_2^k). \tag{2.5}$$

The Fourier coefficients $c_T(E_2^k)$ can be computed using the package `degree2` of Takemori [Tak17] with the command `eisenstein_series_degree2`.

Example 2.5. Let $k = 18$ and $m = 1$. Then

$$c_{(1,1)}\big(E_1^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big) = c_{\left(\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right)}(E_2^k) + 2\cdot c_{\begin{pmatrix} 1 & 1/2\\ 1/2 & 1\end{pmatrix}}(E_2^k) + 2\cdot c_{\left(\begin{smallmatrix}1 & 1\\ 1 & 1\end{smallmatrix}\right)}(E_2^k). \tag{2.6}$$

The right-hand side of (2.6) can be computed in Sage with [Tak17], obtaining the rational output `-5311478523411648/6651496075469717`. We transform it in a floating-point number by multiplying it by `1.0` as

`-5311478523411648/6651496075469717 * 1.0`.

The output is `-0.798538924648852`.

We now compute the left-hand side of (2.6) with the command

`vol_selfint_prim_Heegner(1,WeilRep(S).ds()[0],S)`,

where `S` is the Gram matrix of the lattice $U^{\oplus 2}\oplus E_8(-1)^{\oplus 4}$. The output is `-0.798538924648852`. This verifies that the output of `vol_selfint_prim_Heegner(1,WeilRep(S).ds()[0],S)` is a good approximation of $c_{(1,1)}\big(E_1^k\left(\begin{smallmatrix}\tau_1 & 0\\ 0 & \tau_2\end{smallmatrix}\right)\big)$.

References

[BBBF23] I. Barros, P. Beri, E. Brakkee, and L. Flapan. *Kodaira dimension of moduli spaces of hyperkähler varieties.* https://arxiv.org/abs/2212.12586. 2023.

[BBFW25] I. Barros, P. Beri, L. Flapan, and B. Williams. *Cones of Noether-Lefschetz divisors and moduli spaces of hyperkähler manifolds.* https://arxiv.org/abs/2407.07622. 2025.

[BFZ25] I. Barros, L. Flapan, and R. Zuffetti. *Extremal divisors on moduli spaces of K3 surfaces.* 2025.

[Tak17] S. Takemori. *A Sage package for computation of degree 2 Siegel modular forms.* https://github.com/stakemori/degree2/. 2017.

[Wil] B. Williams. *WeilRep.* Available at https://github.com/btw-47/weilrep.