Analytic evaluation of Hecke eigenvalues for Siegel modular forms of degree two

Owen Colman[*], Alexandru Ghitza[†]and Nathan C. Ryan[‡]

June 18, 2018

The standard approach to evaluate Hecke eigenvalues of a Siegel modular eigenform F is to determine a large number of Fourier coefficients of F and then compute the Hecke action on those coefficients. We present a new method based on the numerical evaluation of F at explicit points in the upper-half space and of its image under the Hecke operators. The approach is more efficient than the standard method and has the potential for further optimization by identifying good candidates for the points of evaluation, or finding ways of lowering the truncation bound. A limitation of the algorithm is that it returns floating point numbers for the eigenvalues; however, the working precision can be adjusted at will to yield as close an approximation as needed.

1 Introduction

The explicit computation of classical modular forms and their associated L-functions has been very useful to formulate and verify conjectures, to discover new phenomena and to prove theorems. There are a variety of ways to effectively compute the Fourier coefficients of classical modular forms and, therefore, their L-functions. Analogous work for Siegel modular forms of degree two is less well-developed for, perhaps, two main reasons:

1. the methods for computing Siegel modular forms are *ad hoc* and less efficient than those for computing classical modular forms;

2. computing Siegel modular forms does not immediately give you the associated L-functions since the Hecke eigenvalues of Siegel modular forms, unlike in the classical case, are not equal to the Fourier coefficients and because the Euler factors of the L-function involve knowing both the pth and the p^2th eigenvalues.

To give an idea of the difficulty of computing the L-function of a Siegel modular form, we consider an example. Let Υ_{20} be the unique normalized Siegel modular form of degree 2 and weight 20 that is a Hecke eigenform and not a Saito-Kurokawa lift. Skoruppa [14] gave an explicit formula for Υ_{20} in terms of the generators of the ring of Siegel modular forms of degree 2 and the largest calculation of Υ_{20} has been carried out by Kohnen and Kuss [8] (we point out that Kurokawa [10, 11] was the first to compute Υ_{20} but his computations

---

[*]owencolman@gmail.com
[†]aghitza@alum.mit.edu
[‡]nathan.ryan@bucknell.edu

were not very extensive). The computation that Kohnen and Kuss carried out was enough to find the pth eigenvalue for $p \leq 997$ and the p^2th eigenvalue for $p \leq 79$. They compute Fourier coefficients indexed by quadratic forms with discriminant up to 3000000 and then use them to determine the Hecke eigenvalues. An examination of the formulas on page 387 of [14] shows that to find the eigenvalue $\lambda(n)$ of T_n, for $n = p^2$, requires the Fourier coefficients indexed by quadratic forms of discriminant up to $n^2 = p^4$. This relation makes it infeasible to compute many more Fourier coefficients, and thus Hecke eigenvalues, using this approach. Instead, in this paper, we propose a different approach.

Our method does not compute *any* of the Fourier coefficients of the Siegel modular form being studied. Instead, we take suitable truncations of the Fourier expansions of the *Igusa generators* (whose coefficients are inexpensive to compute) and use these truncations to evaluate our modular form numerically at points in the upper half space. This approach is based on work of Bröker and Lauter [3] in which they use such techniques to evaluate Igusa functions. Using their method we find the eigenvalue $\lambda(p)$ of an eigenform F by doing the following:

- evaluate F at some point Z in the Siegel upper half-space;

- evaluate $F|T_p$ at the same point Z;

- take the ratio $(F|T_p)(Z)/F(Z)$.

The conceptual shift that we are proposing is that, instead of representing the Siegel modular form F as a list of Fourier coefficients, we represent F by its values at points in the Siegel upper half-space. The idea is simple but its importance can be seen by virtue of the results. We remark that in [2] we describe an implementation of the analogous method for classical modular forms and, in some cases, outperform the standard method using modular symbols.

The potential to parallelize our algorithm stems from the fact that we sum over the coset decomposition of the Hecke operators, and the computation of each summand is independent; these computations can therefore be performed in parallel. Such approaches have been used in the past, for instance in determining the Hecke eigenvalues of paramodular forms, see [13, 4]. We thank the referees for pointing this out, and note that the similarity ends at the level of the sum itself: Poor and Yuen specialize the paramodular eigenform to a modular curve, then compute the summands (which are power series in one variable) exactly. We work with the Siegel eigenform itself (as a power series in three variables) and compute good numerical approximations to the summands.

It is important to emphasize that our method takes as input the expression of a Siegel eigenform as a polynomial in the Igusa generators. Our objective is then to efficiently compute approximate values of the Hecke eigenvalues. We do not claim to obtain further information about the Fourier coefficients of the eigenform, nor that this is an efficient way of determining the exact value of the eigenvalues (unless the latter happen to be integers).

The paper is organized as follows. We begin by stating some numerical preliminaries used in our method. Then, we give the relevant background on Siegel modular forms and discuss Bröker and Lauter's work and how to compute $F|T_p$ both in theory and in practice. We conclude by presenting some results of our computations, together with details of the implementation and ideas for further improvement.

2 Numerical preliminaries

Before we describe our algorithm to compute Hecke eigenvalues of Siegel modular forms analytically, we begin by stating some results related to bounding the error introduced when we evaluate a given Siegel modular form and its image under the Hecke operators T_p and T_{p^2} at a point in the upper half-plane.

2.1 Error in quotient

We have a quantity defined as

$$z = \frac{x}{y} \quad \text{with } x, y \in \mathbb{C}.$$

The numerator and denominator can be approximated to x_A, resp. y_A; we define $z_A := \frac{x_A}{y_A}$. Given $\varepsilon > 0$, what values of ε_x and ε_y ensure that

$$\text{if } |x - x_A| < \varepsilon_x \text{ and } |y - y_A| < \varepsilon_y \text{ then } |z - z_A| < \varepsilon?$$

Lemma 1. *With the above notation, let $e_x = x - x_A$ and $e_y = y - y_A$. Then*

$$z - z_A = \frac{e_x - e_y z_A}{y_A + e_y}.$$

Proof. Straightforward calculation. $\qquad\qquad\square$

Proposition 2. *For any $h \in (0, 1)$, if*

$$\varepsilon_x < \frac{h\varepsilon|y_A|}{2} \quad and \quad \varepsilon_y < \min\left\{ \frac{(1-h)\varepsilon|y_A|}{2|z_A|}, \frac{|y_A|}{2} \right\},$$

then $|z - z_A| < \varepsilon$.

Proof. Under the hypotheses, we have $|y_A + e_y| > |y_A|/2$ so

$$|z - z_A| < \frac{2}{|y_A|}\left(|e_x| + |e_y z_A|\right) < h\varepsilon + (1-h)\varepsilon = \varepsilon.$$

$\qquad\qquad\square$

The value of the parameter h can be chosen in such a way that the calculations of x_A and of y_A are roughly of the same level of difficulty.

In order to use the results of Proposition 2 in practice, we need a lower bound on $|y_A|$ and an upper bound on $|z_A|$ (which can be obtained from the lower bound on $|y_A|$ and an upper bound on $|x_A|$). How do we bound $|x_A|$? We compute a very coarse estimate $\tilde{x}$ to x, with $\tilde{\varepsilon}_x$ just small enough that $|\tilde{x}| - 2\tilde{\varepsilon}_x > 0$. (We can start with $\tilde{\varepsilon}_x = 0.1$ and keep dividing by 10 until the condition holds.) Later we will make sure that ε_x is smaller than $\tilde{\varepsilon}_x$. Then we know that

$$|\tilde{x} - x| < \tilde{\varepsilon}_x \qquad \text{and} \qquad |x_A - x| < \varepsilon_x \leq \tilde{\varepsilon}_x,$$

so

$$\left||x_A| - |\tilde{x}|\right| \leq |x_A - \tilde{x}| < 2\tilde{\varepsilon}_x \qquad \Rightarrow \qquad 0 < |\tilde{x}| - 2\tilde{\varepsilon}_x < |x_A| < |\tilde{x}| + 2\tilde{\varepsilon}_x,$$

giving us lower and upper bounds on $|x_A|$. A similar argument works for $|y_A|$.

3 Siegel modular forms

Let the symplectic group of similitudes of genus 2 be defined by

$$\mathrm{GSp}(4) := \{G \in \mathrm{GL}(4) : {}^t G J G = \lambda(G) J, \lambda(G) \in \mathrm{GL}(1)\}$$

$$\text{where } J = \begin{bmatrix} & I_2 \\ -I_2 & \end{bmatrix}.$$

Let $\mathrm{Sp}(4)$ be the subgroup with $\lambda(G) = 1$. The group $\mathrm{GSp}^+(4, \mathbb{R}) := \{G \in \mathrm{GSp}(4, \mathbb{R}) : \lambda(G) > 0\}$ acts on the Siegel upper half space $\mathbb{H}_2 := \{Z \in M_2(\mathbb{C}) : {}^t Z = Z, \mathrm{Im}(Z) > 0\}$ by

$$(1) \quad G\langle Z \rangle := (AZ + B)(CZ + D)^{-1}, \quad \text{where } G = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathrm{GSp}^+(4, \mathbb{R}), Z \in \mathbb{H}_2.$$

Let $S_k^{(2)}$ be the space of holomorphic Siegel cusp forms of weight k, genus 2 with respect to $\Gamma^{(2)} := \mathrm{Sp}(4, \mathbb{Z})$. Then $F \in S_k^{(2)}$ satisfies

$$F(\gamma\langle Z \rangle) = \det(CZ + D)^k F(Z)$$

for all $\gamma = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \Gamma^{(2)}$ and $Z \in \mathbb{H}_2$. This also can be written in terms of the slash operator: for $M \in \mathrm{GSp}^+(4, \mathbb{R})$ let $(F|_k M)(Z) = \det(CZ + D)^{-k} F(M\langle Z \rangle)$. Then the functional equation satisfied by a Siegel modular form can be written as:

$$(F|_k M)(Z) = F(z)$$

for all $M \in \mathrm{Sp}(4, \mathbb{Z})$.

Now we describe the Hecke operators acting on $S_k^{(2)}$. For $M \in \mathrm{GSp}^+(4, \mathbb{R}) \cap M_4(\mathbb{Z})$, define the Hecke operator $T(\Gamma^{(2)} M \Gamma^{(2)})$ on $S_k^{(2)}$ as in [1, (1.3.3)]. For a positive integer m, we define the Hecke operator T_m by

$$(2) \quad T_m := \sum_{\lambda(M) = m} T(\Gamma^{(2)} M \Gamma^{(2)}).$$

See Section 5.1 for an explicit decomposition of the double cosets T_p and T_{p^2} into right cosets. Suppose

$$T_m = \sum \Gamma^{(2)} \alpha$$

is a right coset decomposition of the Hecke operator T_m. Then the operator T_m acts on a Siegel modular form F of weight k as

$$(F|_k T_m)(Z) = \sum (F|_k \alpha)(Z).$$

This action can be described in terms of the Fourier coefficients of the Siegel modular form F.

Any Siegel modular form F of degree 2 has a Fourier expansion of the form

$$F(Z) = \sum_N a_N(F) \exp(2\pi i \, \mathrm{Tr}(NZ)) \qquad a_N(F) \in \mathbb{C},$$

where the sum ranges over all positive semi-definite matrices $N = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$ with $a, b, c \in \mathbb{Z}$. The quadratic form N is often written $[a, b, c]$ using Gauss's notation. Using

the decompositions of the Hecke operators in Section 5.1 one can derive formulas for the action of T_p and T_{p^2} on a Siegel modular form F. When these formulas are written down as in [14, p. 387] one can see that to compute $\lambda_F(p)$, the Hecke eigenvalue of F with respect to the Hecke operator T_p, one needs Fourier coefficients up to discriminant of order p^2. To compute $\lambda_F(p^2)$, the Hecke eigenvalue of F with respect to the Hecke operator T_{p^2}, one needs Fourier coefficients up to discriminant p^4. With current methods, computing this number of coefficients of a Hecke eigenform that is not a Saito-Kurokawa lift has proven impossible.

A bottleneck to computing such a large number of coefficients is the fact that there is no known way to compute individual coefficients in parallel. The determination of a single Fourier coefficient requires knowledge of many other Fourier coefficients. Our method, described above, has approximately the same number of steps to compute a new Hecke eigenvalue but these steps, in our method, are easily done in parallel.

4 Evaluating Hecke eigenforms

4.1 Bounds on the coefficients of the Igusa generators

Proposition 3. *Let E_4, E_6, χ_{10} and χ_{12} denote the Igusa generators of the ring of even-weight Siegel modular forms of genus 2 with respect to $Sp(4, \mathbb{Z})$.*

We have the following bounds on the Fourier coefficients of these forms:

$$|a_N(E_4)| < 19\,230\,t^5,$$
$$|a_N(E_6)| < 12\,169\,t^9,$$
$$|a_N(\chi_{10})| < \frac{1}{236}\,A(\varepsilon, 9)\,t^{9+\varepsilon},$$
$$|a_N(\chi_{12})| < \frac{1}{311}\,A(\varepsilon, 11)\,t^{11+\varepsilon},$$

where the last two hold for any $\varepsilon > 0$, $t = \mathrm{Tr}(N)$, and the function $A(\varepsilon, s)$ is defined by

$$A(\varepsilon, s) = \frac{1}{(2\pi)^{1/4}}\,\exp\left(9\varepsilon^{-1}2^{3/\varepsilon}/\log(2)\right)\,\zeta(1+\varepsilon)\,\max\left\{1, \sqrt{\frac{\Gamma(s+1/2+\varepsilon)}{\Gamma(s-1/2-\varepsilon)}}\right\}.$$

Proof. It follows directly from [3, Corollary 3.6 and Remark 3.7] that

$$|a_N(E_4)| < 19\,230\left(4ac - b^2\right)^{5/2} \leq 19\,230\,\mathrm{Tr}(N)^5,$$
$$|a_n(E_6)| < 12\,169\left(4ac - b^2\right)^{9/2} \leq 12\,169\,\mathrm{Tr}(N)^9.$$

The second two inequalities follow from [3, Theorem 5.10] with $\gamma = \eta = \varepsilon/3$. $\quad\square$

Remark 4. The bounds for χ_{10} and χ_{12} in Proposition 3 allow for further optimization by choosing the parameter ε appropriately.

Considering χ_{10}, the factor $t^{9+\varepsilon}$ is of course dominant as $t \to \infty$, but choosing ε as small as possible is counterproductive for practical computations, as the factor $A(\varepsilon, 9)$ explodes for small ε.

In our computations, we use $\varepsilon = 2$, so the bounds can be summarized as:

$$|a_N(E_4)| < 19\,230\,t^5,$$
$$|a_N(E_6)| < 12\,169\,t^9,$$
$$|a_N(\chi_{10})| < 220\,439\,t^{11},$$
$$|a_N(\chi_{12})| < 287\,248\,t^{13},$$

where $t = \mathrm{Tr}(N)$.

4.2 The truncation error for Siegel modular forms

Let F be a Siegel modular form of degree 2, with Fourier expansion

$$F(Z) = \sum_N a_N(F) \exp\left(2\pi i \,\mathrm{Tr}(NZ)\right).$$

Given a positive integer T, we will truncate the Fourier expansion of F by considering only those indices N whose trace is at most T:

$$F_T(Z) = \sum_{\mathrm{Tr}(N) \le T} a_N(F) \exp\left(2\pi i \,\mathrm{Tr}(NZ)\right).$$

Lemma 5. *For any $t \in \mathbb{N}$, the number of Fourier indices of trace t satisfies*

$$\#\{N \mid \mathrm{Tr}(N) = t\} \le (t+1)(2t+1) = 2t^2 + 3t + 1 \le 6t^2.$$

Proof. We have

$$\#\{N \mid \mathrm{Tr}(N) = t\} = \sum_{a=0}^{t}\left(1 + 2\left\lfloor 2\sqrt{a(t-a)}\right\rfloor\right).$$

There are $t + 1$ terms in the sum, and the largest corresponds to $a = t/2$ (or $a = (t-1)/2$ if t is odd). In any case, every term in the sum is at most $1 + 2t$. $\qquad\square$

Suppose we have, like in Proposition 3, an upper bound on the Fourier coefficients of F:

$$(3) \qquad\qquad |a_N(F)| \le Ct^d \qquad \text{where } C \in \mathbb{R}_{>0},\ d \in \mathbb{N} \text{ and } t = \mathrm{Tr}(N).$$

We are interested in bounding the gap between the true value $F(Z)$ and its approximation $F_T(Z)$.

Proposition 6. *Suppose F is a Siegel modular form of degree two whose Fourier coefficients satisfy Equation (3), $Z \in \mathbb{H}_2$ and we wish to approximate the value $F(Z)$ with error at most 10^{-h}. It is then sufficient to use the truncation $F_T(Z)$ containing all terms of the Fourier expansion of F with indices of trace at most T, where*

$$T > \frac{d+2}{\alpha(Z)} \qquad and \qquad 6C\frac{d+3}{\alpha(Z)}\exp(-\alpha(Z)T)T^{d+2} < 10^{-h}.$$

Here

$$\delta(Z) = \sup\left\{\delta' \in \mathbb{R} \mid \mathrm{Im}(Z) - \delta' I \text{ is positive semi-definite}\right\}$$

and $\alpha(Z) = 2\pi\delta(Z)$.

Proof. Using [3, Lemma 6.1], we have

$$
\begin{aligned}
|F(Z) - F_T(Z)| &= \left| \sum_{\mathrm{Tr}(N) > T} a_N(F) \exp\left(2\pi i \,\mathrm{Tr}(NZ)\right) \right| \\
&\leq \sum_{\mathrm{Tr}(N) > T} |a_N(F)| \left| \exp\left(2\pi i \,\mathrm{Tr}(NZ)\right) \right| \\
&\leq \sum_{\mathrm{Tr}(N) > T} |a_N(F)| \exp\left(-\alpha(Z)\,\mathrm{Tr}(N)\right) \\
&< \sum_{t=T+1}^{\infty} \sum_{\mathrm{Tr}(N)=t} |a_N(F)| \exp\left(-\alpha(Z)t\right) \\
&\leq \sum_{t=T+1}^{\infty} 6C t^{d+2} \exp\left(-\alpha(Z)t\right) \\
&\leq 6C \int_T^{\infty} x^{d+2} \exp\left(-\alpha(Z)x\right)\, dx \\
&= 6C \exp(-\alpha(Z)T) \sum_{j=0}^{d+2} \frac{(d+2)!}{j!\alpha(Z)^{d-j+3}} T^j \\
&< \frac{6C(d+3)}{\alpha(Z)} \exp(-\alpha(Z)T) T^{d+2},
\end{aligned}
$$

where the last inequality holds if T is in the half-infinite interval on which the integrand is decreasing (i.e. $T > (d+2)/\alpha(Z)$). $\qquad\square$

Example 7. We determine T sufficient for computing $E_4(Z)$ within 10^{-20} at the point

$$
z = \begin{pmatrix} 5i & i \\ i & 6i \end{pmatrix}.
$$

We have

$$
\alpha(Z) = 27.5327
$$

so we are looking for T such that

$$
\exp(-\alpha(Z)T)T^7 < 2.983 \cdot 10^{-25},
$$

which is easily seen (numerically) to hold as soon as $T \geq 3$.

We proceed similarly to obtain the values in Table 1.

error	E_4	E_6	χ_{10}	χ_{12}
10^{-10}	2	2	2	2
10^{-20}	3	3	3	3
10^{-100}	10	10	10	11
10^{-1000}	86	86	87	87

Table 1: Truncation necessary for computing $F(Z)$ within specified error

5 Our method

As described above, our method is rather straightforward. We fix a $Z \in \mathbb{H}^2$ and evaluate $F(Z)$, using methods in Section 4. Consider the double coset $T_p = \sum \Gamma^{(2)} \alpha$ and its action on F:

$$\left(F|_k T_p\right)(Z) = \sum \left(F|_k \alpha\right)(Z).$$

What is left to do, then, is, for α in the decomposition, to compute $\left(F|_k \alpha\right)(Z)$. That is, to be able to write α as $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ and to be able to evaluate

$$\det(CZ + D)^{-k} F\left((AZ + B)(CZ + D)^{-1}\right).$$

In Section 5.1 we present the desired decompositions for the Hecke operators T_p and T_{p^2} and we use the methods of Section 4 to evaluate the Siegel modular form at the points $(AZ + B)(CZ + D)^{-1} \in \mathbb{H}^2$.

5.1 Hecke action

Hecke operators are defined in terms of double cosets $\Gamma M \Gamma$ and the action of such an operator is determined by the right cosets that appear in the decomposition of these double cosets. For a prime p we consider the double coset $T_p = \Gamma^{(2)} \operatorname{diag}(1, 1, p, p)\Gamma^{(2)}$. An explicit version of a formula, due to Andrianov, for the right cosets that appear in the decomposition of T_p, is given by Cléry and van der Geer:

Proposition 8. [1, 6] The double coset T_p admits the following left coset decomposition:

$$\Gamma^{(2)} \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \sum_{0 \le a,b,c \le p-1} \Gamma^{(2)} \begin{pmatrix} 1 & 0 & a & b \\ 0 & 1 & b & c \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix} +$$

$$\sum_{0 \le a \le p-1} \Gamma^{(2)} \begin{pmatrix} 0 & -p & 0 & 0 \\ 1 & 0 & a & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & p & 0 \end{pmatrix} + \sum_{0 \le a,m \le p-1} \Gamma^{(2)} \begin{pmatrix} p & 0 & 0 & 0 \\ -m & 1 & 0 & a \\ 0 & 0 & 1 & m \\ 0 & 0 & 0 & p \end{pmatrix}$$

and we have that the degree of T_p is $p^3 + p^2 + p + 1$.

Thus, in particular, in order to find λ_p, then, $p^3 + p^2 + p + 1$ independent evaluations of our Siegel modular form F at points in $\mathbb{H}^2$ are required. This is why our method is so amenable to parallelization.

Similarly, for a prime p define the operator T_{p^2} as a sum of double cosets:

$$T_{p^2} = \Gamma^{(2)} \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix} \Gamma^{(2)} + \Gamma^{(2)} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p^2 & 0 \\ 0 & 0 & 0 & p \end{pmatrix} \Gamma^{(2)} + \Gamma^{(2)} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p^2 & 0 \\ 0 & 0 & 0 & p^2 \end{pmatrix} \Gamma^{(2)}$$

Again, based on a result of Andrianov, Cléry and van der Geer give an explicit decomposition of the operator T_{p^2}:

Proposition 9. [1, 6] The Hecke operator T_{p^2} has degree $p^6 + p^5 + 2p^4 + 2p^3 + p^2 + p + 1$ and admits a known explicit left coset decomposition.

One can do better, however; we can reduce the number of summands at which we need to evaluate F to be $\mathcal{O}(p^4)$ instead of $\mathcal{O}(p^6)$ by using some standard facts about the Hecke

algebra for Siegel modular forms of degree 2. The Hecke operator T_{p^2} is itself a linear combination of three double cosets:

$$(4) \quad T_{p^2,0} = \Gamma^{(2)} \operatorname{diag}(p,p;p,p)\Gamma^{(2)}, \ T_{p^2,1} = \Gamma^{(2)} \operatorname{diag}(1,p;p^2,p)\Gamma^{(2)}, \ \text{and}$$
$$T_{p^2,2} = \Gamma^{(2)} \operatorname{diag}(1,1;p^2,p^2)\Gamma^{(2)}.$$

The decomposition in Proposition 9 is itself the (disjoint) sum of the decomposition of three double cosets $T_{p^2,0}$, $T_{p^2,1}$ and $T_{p^2,2}$.

The p-part of the Hecke algebra is generated by the operators T_p, $T_{p^2,0}$ and $T_{p^2,1}$ and, in fact, in [9, 17], it is shown that

$$(5) \quad (T_p)^2 = T_{p^2,0} + (p+1)T_{p^2,1} + (p^2+1)(p+1)T_{p^2,2}.$$

To determine the eigenvalue $\lambda_{p^2}(F)$ for $F \in S_k^{(2)}$ with respect to the Hecke operator T_{p^2}, using Proposition 8, we first find the eigenvalue $\lambda_p(F)$ for the operator T_p. Then, we find the eigenvalues $\lambda_{p^2,0}(F)$ (known to be p^{-2k} by the definitions in Section 3) and the eigenvalue $\lambda_{p^2,1}(F)$ for the operator $T_{p^2,1}$. Then using (5) we can find the eigenvalue $\lambda_{p^2,2}(F)$ for the operator $T_{p^2,2}$. Putting it all together, then, all we need is an explicit decomposition of $T_{p^2,1}$ into left cosets, in order to compute $\lambda_{p^2}(F)$.

Proposition 10 ([1]). *The Hecke operator $T_{p^2,1}$ admits the following left coset decomposition:*

$$\sum_{0 \le \alpha < p} \Gamma^{(2)} \begin{pmatrix} p^2 & 0 & 0 & 0 \\ -p\alpha & p & 0 & 0 \\ 0 & 0 & 1 & \alpha \\ 0 & 0 & 0 & p \end{pmatrix} \Gamma^{(2)} + \Gamma^{(2)} \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & p^2 & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Gamma^{(2)} + \sum_{\substack{0 \le a,b,c < p \\ ac-b^2 \equiv 0 \pmod{p} \\ and\ not\ all\ zero}} \Gamma^{(2)} \begin{pmatrix} p & 0 & a & b \\ 0 & p & b & c \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix} \Gamma^{(2)} +$$

$$\sum_{\substack{0 \le \alpha,\beta < p \\ 0 \le C < p^2}} \Gamma^{(2)} \begin{pmatrix} p & 0 & 0 & p\beta \\ -\alpha & 1 & \beta & \alpha\beta+C \\ 0 & 0 & p & p\alpha \\ 0 & 0 & 0 & p^2 \end{pmatrix} \Gamma^{(2)} + \sum_{\substack{0 \le \beta < p \\ 0 \le A < p^2}} \Gamma^{(2)} \begin{pmatrix} 1 & 0 & A & \beta \\ 0 & p & p\beta & 0 \\ 0 & 0 & p^2 & 0 \\ 0 & 0 & 0 & p \end{pmatrix} \Gamma^{(2)}.$$

Thus the degree of $T_{p^2,1}$ is $p^4 + p^3 + p^2 + p$.

Remark 11. In the Introduction, we discussed the difficulty of computing $\lambda_{p^2}(F)$ using the action of T_{p^2} on the coefficients of the eigenform F. One might ask whether we could more efficiently compute $\lambda_{p^2}(F)$ using the action of $T_{p^2,1}$ on F as described in Proposition 10 and (5). It turns out, though, that one still would require coefficients up to discriminant p^4 using $T_{p^2,1}$ and (5).

6 Some computations and implementation details

We describe some sample computations involving the eigenform of smallest weight that is not a lift from lower rank groups, namely the cusp form Υ_{20} mentioned in the introduction:

$$\Upsilon_{20} = -E_4^2\chi_{12} - E_4 E_6 \chi_{10} + 1785600\chi_{10}^2.$$

As a gauge of the performance of the algorithm, we compared the timings to those required by the implementation [16] of the standard method[1] by Sho Takemori.

---

[1] The only other publicly-available implementation we are aware of is [5]. We did not compare against it for two reasons: (a) at the moment, the computation of the Hecke image appears to be incorrect for primes that are congruent to 1 mod 4 and (b) it uses Cython for the most expensive part of the computation, namely the multiplication of the q-expansions. Since both our code and S. Takemori's are pure Python, we deemed this to be a more useful comparison of the two algorithms.

We implemented the method described in this paper in SageMath [15]; this implementation is available at [7]. The benchmarks described below were performed using a single core of a Linux machine with an i7-6700 CPU at 3.40GHz and 64GB of RAM, via the following helper functions:

```
def ups20_eigenvalue_numerical(p, prec, y11):
    CRING = _initialise_rings(prec, 2*p)
    Z = matrix(CRING, 2, 2, [y11*i, i, i, (y11+1)*i])
    R.<a, b, c, d> = QQ[]
    f = -a^2*d-a*b*c+1785600*c^2
    return _eigenvalue_T_fixed_trace(f, Z, p, 2*p)

def ups20_eigenvalue_standard(p):
    with degree2_number_of_procs(1):
        a = eisenstein_series_degree2(4, p)
        b = eisenstein_series_degree2(6, p)
        c = x10_with_prec(p)
        d = x12_with_prec(p)
        f = -a^2*d-a*b*c+1785600*c^2
        return f.hecke_eigenvalue(p)
```

p	y_{11}	precision (bits)	numerical (s)	standard (s)
2	2.7	37	0	0
3	4.3	62	0	0
5	6.1	101	0	0
7	7.5	130	1	1
11	9.5	172	3	7
13	10.3	190	6	15
17	10.9	208	16	55
19	11.9	226	25	90
23	12.3	240	54	230
29	13.5	267	140	735
31	13.9	275	186	1185
37	14.5	295	406	2876

Table 2: Benchmarks comparing the numerical and standard algorithms for computing the Hecke eigenvalues of Υ_{20}. The timings are rounded to the nearest second. The working precision was chosen so that the eigenvalue is the closest integer to the computed floating point number.

For the standard algorithm, the most expensive step appears to be the multiplication of the q-expansions of the Igusa generators. In the case of our numerical algorithm, the majority of the time is spent evaluating truncations of the q-expansions of the Igusa generators at various points in the Siegel upper half space. These functions are polynomials in the variables q_1, q_2, q_3 and q_3^{-1}, where

$$Z = \begin{pmatrix} z_1 & z_3 \\ z_3 & z_2 \end{pmatrix} \qquad \text{and} \qquad q_j = e^{2\pi i z_j}.$$

To evaluate such functions efficiently at a large number of points, we implemented an iterative version of Horner's method; to illustrate what is involved, here is how the truncation of the Igusa generator χ_{10} at trace up to 3 is evaluated:

$$q_1 \left(q_2 \left(q_3^{-1} - 2 + q_3 + q_2 \left(-2q_3^{-2} - 16q_3^{-1} + 36 - 16q_3 - 2q_3^2 \right) \right) \right.$$
$$\left. + q_1 \left(q_2 \left(-2q_3^{-1} - 16q_3^{-1} + 36 - 16q_3 - 2q_3^2 \right) \right) \right)$$

Many of the partial evaluations are repeated for different summands of the expression for the Hecke operators. We take advantage of this phenomenon by caching the results of evaluations of polynomials in q_3 and q_3^{-1}. All the operations are performed using interval arithmetic (via the `ComplexIntervalField` available in Sage). While this introduces a small overhead, it frees us from having to keep track of precision loss due to arithmetic operations (and evaluations of the complex exponential function). Sage gives the final approximation of the Hecke eigenvalue in the form

```
1.055528218470800414110149180000000000000000?e27 + 0.?e-13*I
```

from which we observe that the answer is most likely the integer

```
1055528218470800414110149180
```

which is indeed $\lambda_{29}(\Upsilon_{20})$. The question mark in the floating point number indicates that the last decimal may be incorrect due to rounding errors (but all preceding decimals are guaranteed to be correct).

There are certainly many variants of our choices that deserve further scrutiny and may lead to improved performance. Here are some of the more interesting ones:

- For computing the eigenvalue λ_p, we chose to focus on the initial evaluation point

$$Z = \begin{pmatrix} y_{11}i & i \\ i & (y_{11}+1)i \end{pmatrix},$$

  where the parameter y_{11} is (at the moment) determined by trial and error. The optimal values of y_{11} for Υ_{20} and small p are listed in the second column of Table 2. We note that the dependence of this optimal y_{11} on p appears to be linear in $\log(p)$.

  The choice of Z is significant for another reason: the fact that Z is a "purely imaginary matrix" gives an extra symmetry that allows to reduce the number of overall computations by almost a factor of 2. Note that the timings listed in Table 2 do not incorporate this optimization.

- Our experiments indicate that computing the value of λ_p accurately using the choice of point Z described above requires truncating the q-expansions of the Igusa generators at trace up to $2p$. It would be very interesting to see if this trace bound can be lowered; even a small improvement in the trace can reduce the computation time significantly. We have observed such phenomena in the case of classical modular forms (treated in [2]).

6.1 Summary of further computations

We performed similar numerical experiments with the following forms:

$$\Upsilon_{22} = 61E_4^3\chi_{10} - 30E_4E_6\chi_{12} + 5E_6^2\chi_{10} - 80870400\chi_{10}\chi_{12}$$

$$\Upsilon_{24a} = -67E_4^3\chi_{12} + 78E_4^2E_6\chi_{10} - 274492800E_4\chi_{10}^2 + 25E_6^2\chi_{12} + 71539200\chi_{12}^2$$

$$\Upsilon_{24b} = +70E_4^3\chi_{12} - 69E_4^2E_6\chi_{10} - 214341120E_4\chi_{10}^2 + 53E_6^2\chi_{12} - 137604096\chi_{12}^2$$

$$\Upsilon_{26a} = -22E_4^4\chi_{10} - 3E_4^2E_6\chi_{12} + 31E_4E_6^2\chi_{10} - 96609024E_4\chi_{10}\chi_{12} - 13806720E_6\chi_{10}^2$$

$$\Upsilon_{26b} = 973E_4^4\chi_{10} + 390E_4^2E_6\chi_{12} - 1255E_4E_6^2\chi_{10} + 3927813120E_4\chi_{10}\chi_{12} - 4438886400E_6\chi_{10}^2$$

These have in common that they are all "interesting" forms (Skoruppa's terminology and notation), not arising as lifts from lower rank groups. They also all have rational coefficients (and are very likely the only rational "interesting" forms in level one).

As we can see in Table 3, while the standard method slows down rapidly with the increase in the weight, the numerical method seems unaffected by the weight (in this range).

f	numerical (s)	standard (s)	$\lambda_{23}(f)$
Υ_{20}	57	240	$-7159245922546757692913520$
Υ_{22}	59	410	$1288399464282335021926848240$
Υ_{24a}	59	559	$-5704707743633516358011\,33259440$
Υ_{24b}	59	563	$-2612738224352475069296861434032$
Υ_{26a}	59	658	$1723965639346061287785316101052080$
Υ_{26b}	60	659	$-2455694249118004577637986236157520$

Table 3: Benchmarks comparing the numerical and standard algorithms for computing the Hecke eigenvalue λ_{23} of the rational "interesting" eigenforms. The timings are rounded to the nearest second.

As we increase the weight further, we encounter "interesting" eigenforms defined over number fields of increasing degree. Our implementation treats these in the same way as the rational eigenforms; the algebraic numbers appearing in the expression of an eigenform as a polynomial in the Igusa generators are first embedded into the `ComplexIntervalField` with the working precision, and the computations are then done exclusively with complex intervals.

We illustrate this with a number of examples from the L-functions and Modular Forms Database (LMFDB [12]): $\Upsilon_{28}, \Upsilon_{30}, \dots, \Upsilon_{56}$, contributed by Nils-Peter Skoruppa. These are representatives of the unique Galois orbit of "interesting" Siegel modular eigenforms of level one and weights given by the indices. We computed the integer closest to the eigenvalues $\lambda_2, \lambda_3, \dots, \lambda_{11}$ of these forms and verified the results against Sho Takemori's implementation.[2] The timings for λ_{11} appear in Table 4. We note once again that the change in weight has only a very minimal effect on the timings for the numerical approach. The degree of the number field over which each eigenform is defined varies from 3 for Υ_{28} to 29 for Υ_{56}.

---

[2]The LMFDB contains only λ_2, λ_3 and λ_5 for the forms $\Upsilon_{28}, \dots, \Upsilon_{48}$. We are not aware of the other eigenvalues we computed having been published anywhere.

f	numerical (s)	standard (s)	integer closest to $\lambda_{11}(f)$
Υ_{28}	5	42	$-575968117847737372167184 9774$
Υ_{30}	5	55	$25584027381199484130020567 5092$
Υ_{32}	5	72	$-6288907983750007346806149 6815555$
Υ_{34}	5	99	$43908608457248526492250997 0244600$
Υ_{36}	5	145	$-1085248116783567484088793 200996441965$
Υ_{38}	5	171	$99082752899176432104304580 529696472526$
Υ_{40}	6	316	$21639993149436935203941512 756710465353890$
Υ_{42}	6	405	$13264330942760158288281314 22320612505802642$
Υ_{44}	6	697	$-21625483413302053328965786 6886176910904279874$
Υ_{46}	6	1156	$30250103567979818612290216 8227017802 3420599162$
Υ_{48}	6	2147	$36236812596076837013528898 6324690125 1092385443364$
Υ_{50}	6	3558	$-50111326406849287661448298 5499331396 73192742821477$
Υ_{52}	6	7701	$-33891727074702812676183940 8879952198 01531644658145401$
Υ_{54}	6	12205	$-43243637347378158947714106 2825913385 1153783375885366874$
Υ_{56}	7	19290	$80732614396781887621126152 4740739769 8956319035 44298785221$

Table 4: Benchmarks comparing the numerical and standard algorithms for computing the Hecke eigenvalue λ_{11} of a representative of the unique Galois orbit of "interesting" eigenforms in each of the listed weights. The timings are rounded to the nearest second.

References

[1] A. N. Andrianov. *Quadratic forms and Hecke operators*, volume 286 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1987.

[2] D. Armendáriz, O. Colman, A. Ghitza, N. C. Ryan, and D. Terán. Analytic evaluation of Hecke eigenvalues for classical modular forms. arXiv:1806.01586 [math.NT], 2018.

[3] R. Bröker and K. Lauter. Evaluating Igusa functions. *Math. Comp.*, 83(290):2977–2999, 2014.

[4] A. Brumer, A. Pacetti, C. Poor, G. Tornaria, J. Voight, and D. S. Yuen. On the paramodularity of typical abelian surfaces. arXiv:1805.10873 [math.NT], 2018.

[5] C. Citro, A. Ghitza, M. Raum, N. Ryan, N.-P. Skoruppa, and G. Tornaria. Implement scalar-valued Siegel modular forms on $\mathrm{Sp}(4, \mathbb{Z})$. https://trac.sagemath.org/ticket/8701, 2010. [Online; last accessed 2 March 2018].

[6] F. Cléry and G. van der Geer. Constructing vector-valued Siegel modular forms from scalar-valued Siegel modular forms. *Pure and Applied Mathematics Quarterly*, 11(1):21–47, 2015.

[7] O. Colman, A. Ghitza, and N. C. Ryan. SageMath code for the analytic calculation of the Hecke eigenvalues of Siegel modular forms of degree two. https://bitbucket.org/aghitza/hecke-analytic-siegel, 2018. [Online; accessed 5 June 2018].

[8] W. Kohnen and M. Kuss. Some numerical computations concerning spinor zeta functions in genus 2 at the central point. *Math. Comp.*, 71(240):1597–1607, 2002.

[9] A. Krieg. Hecke algebras. *Mem. Amer. Math. Soc.*, 87(435):x+158, 1990.

[10] N. Kurokawa. Examples of eigenvalues of Hecke operators on Siegel cusp forms of degree two. *Invent. Math.*, 49(2):149–165, 1978.

[11] N. Kurokawa. Congruences between Siegel modular forms of degree two. *Proc. Japan Acad. Ser. A Math. Sci.*, 55(10):417–422, 1979.

[12] The LMFDB Collaboration. The L-functions and Modular Forms Database. http://www.lmfdb.org, 2018. [Online; accessed 30 May 2018].

[13] C. Poor and D. Yuen. Paramodular cusp forms. *Math. Comp.*, 84(293):1401–1438, 2015.

[14] N.-P. Skoruppa. Computations of Siegel modular forms of genus two. *Math. Comp.*, 58(197):381–398, 1992.

[15] W. A. Stein et al. *Sage Mathematics Software (Version 8.2)*. The Sage Development Team, 2018. http://www.sagemath.org.

[16] S. Takemori. A Sage package for computation of degree 2 Siegel modular forms. https://github.com/stakemori/degree2, 2018. [Online; last accessed 2 March 2018].

[17] G. van der Geer. Siegel modular forms and their applications. In *The 1-2-3 of modular forms*, Universitext, pages 181–245. Springer, Berlin, 2008.