Computing With Modular Forms

William Stein

November 28, 2005

Contents

Preface **7**

1 Modular Forms **9**
1.1 Basic Definitions 9
1.2 Modular Forms of Level 1 10
1.3 Modular Forms of Any Level 12
1.4 Exercises 14

2 Modular Forms of Level One **15**
2.1 Examples of Modular Forms of Level 1 15
2.1.1 The Cusp Form Δ 16
2.1.2 Fourier Expansions of Eisenstein Series 16
2.2 Structure Theorem For Level 1 Modular Forms 18
2.3 The Victor Miller Basis 21
2.4 Hecke Operators 22
2.5 Computing Hecke Operators 25
2.5.1 A Conjecture about Complexity 26
2.6 Exercises 27

3 Modular Symbols of Weight Two **29**
3.1 Review of modular forms and Hecke operators 30
3.2 Modular symbols 32
3.2.1 Manin's trick 34
3.2.2 Manin symbols 36
3.2.3 Hecke operators on modular symbols 40
3.3 Computing the boundary map 43
3.4 Computing a basis for $S_2(\Gamma_0(N))$ 45
3.4.1 Examples 47
3.5 Computing $S_2(\Gamma_0(N))$ using eigenvectors 49
3.5.1 Summary 50
3.6 Exercises 50

4 Dirichlet Characters **53**
4.1 The Definition 54
4.2 Dirichlet Characters in SAGE 54
4.3 Representing Dirichlet Characters 56
4.4 Evaluation of Dirichlet Characters 58
4.5 Conductors of Dirichlet Characters 62
4.6 Restriction, Extension, and Galois Orbits 64
4.7 Alternative Representations of Characters 66
4.8 Exercises 67

5 Eisenstein Series **69**
5.1 Generalized Bernoulli Numbers 69
5.2 Explicit Basis for the Eisenstein Subspace 71
5.3 Exercises 73

6 Dimensions Formulas **75**
6.1 Modular Forms for $\Gamma_0(N)$ 76
6.1.1 New and Old Subspaces 77
6.2 Modular Forms for $\Gamma_1(N)$ 80
6.3 Modular Forms with Character 81
6.4 Exercises 84

7 Linear Algebra **85**
7.1 Echelon Forms of Matrices 85
7.2 Echelon Forms over $\mathbb{Q}$ 88
7.3 Polynomials 93
7.4 Decomposing Spaces 93
7.4.1 Wiedemann's Minimal Polynomial Algorithm 94
7.4.2 Polynomial Factorization 98
7.4.3 Decomposition Using Kernels 98
7.4.4 Multi-Modular Decomposition Algorithm 98

8 Modular Symbols of any Weight and Level **101**
8.1 Modular Symbols 102
8.2 Manin Symbols 104
8.2.1 Coset Representatives and Manin Symbols 107
8.2.2 Modular Symbols With Character 108
8.3 Hecke Operators 108
8.3.1 General Definition of Hecke Operators 109
8.3.2 Hecke Operators on Manin Symbols 111
8.3.3 Remarks on Complexity 113
8.4 Cuspidal Modular Symbols 114
8.5 The Pairing Between Modular Symbols and Modular Forms 115
8.6 Explicitly Computing $\mathbb{M}_k(\Gamma_0(N)$ 119
8.6.1 Computing $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ 119
8.6.2 Examples of Computation of $\mathbb{M}_k(\Gamma_0(N))$ 122

8.6.3 Refined Algorithm For Computing Presentation 130
8.7 Applications 133
8.7.1 Later in this Book 133
8.7.2 Discussion of the Literature and Research 133
8.8 Exercises 134

9 Computing Spaces of Modular Forms 137
9.1 Decomposing Modular Forms Using Dirichlet Characters 137
9.2 Atkin-Lehner-Li Theory 139
9.3 Computing Cuspforms Using Modular Symbols 141
9.4 Computing Systems of Eigenvalues 142
9.4.1 Computing Projection Onto a Subspace 143
9.4.2 Systems of Eigenvalues 143

10 Special Values of L-functions and Periods 149
10.1 The Period Mapping and Complex Torus Attached to a Newform 149
10.2 Extended Modular Symbols 151
10.3 Numerically Approximating Period Integrals 151
10.4 Speeding Convergence Using the Atkin-Lehner Operator 154
10.4.1 Another Atkin-Lehner Trick 156
10.5 Computing the Period Mapping 157
10.6 Computing Elliptic Curves of Given Conductor 158
10.6.1 Using Modular Symbols 158
10.6.2 Finding Curves by Finding S-Integral Points 159
10.7 Examples 160
10.7.1 Jacobians of genus-two curves 160
10.7.2 Level one cusp forms 161
10.7.3 CM elliptic curves of weight greater than two 162
10.8 Exercises 162

11 Congruences 165
11.1 Congruences Between Modular Forms 165
11.1.1 The j-invariant 165
11.1.2 Congruences for Modular Forms 166
11.1.3 Congruence for Newforms 169
11.2 Generating the Hecke Algebra as a $\mathbb{Z}$-module 170

12 Miscellaneous 171
12.1 Computing Widths of Cusps 171

Preface

This is a book about algorithms for computing with modular forms that started as a series of notes for a graduate course at Harvard University in 2004. This book is meant to answer the question "How do *you* compute spaces of modular forms", by both providing a clear description of the specific algorithms that are used and explaining how to apply them using SAGE [SJ05].

I have spent many years trying to find good practical ways to compute with classical modular forms for congruence subgroups of $\mathrm{SL}_2(\mathbb{Z})$, and have implemented most of these algorithms several times, first in C++ [Ste99], then in MAGMA [BCP97], and most recently as part of SAGE. Much of this work has involved turning formulas and constructions burried in obscure research papers into precise computational recipes, then testing these in many cases and eliminating subtle inaccuracies (published theorems sometimes contain small mistakes that appear magnified when implemented and run on a computer). The goal of this book is to explain some of what I have learned along the way.

The author is aware of no other books on computing with modular forms, the closest work being Cremona's book [Cre97a], which is about computing with elliptic curves, and Cohen's book [Coh93] about algebraic number theory. The field is not yet mature, and there are missing details and potential improvements to many of the algorithms, which you the reader might fill in, and which would be greatly appreciated by other mathematicians.

This book focuses on how best to compute the spaces $M_k(N, \varepsilon)$ of modular forms, where $k \geq 2$ is an integer and ε is a Dirichlet character modulo N. I will spend the most effort explaining the algorithms that appear so far to be the best (in practice!) for such computations. I will not discuss computing half-integral weight forms, weight one forms, forms for non-congruence subgroups or groups other than GL_2, Hilbert and Siegel modular forms, trace formulas, p-adic modular forms, and modular abelian varieties, all of which are topics for another book.

The reader is not assumed to have prior exposure to modular forms, but should be familiar with abstract algebra, basic algebraic number theory, Riemann surfaces, and complex analysis.

Acknowledgement. Kevin Buzzard made many helpful remarks which were helpful in finding the algorithms in Chapter 4. Noam Elkies made many remarks about chapters 1 and 2. The students in the Harvard course made helpful remark; in particular, Abhinav Kumar made observations about computing widths of cusps, Thomas James Barnet-Lamb about how to represent Dirichlet characters, and Tseno V. Tselkov, Jennifer Balakrishnan and Jesse Kass made other remarks.

Parts of Chapter 1 follow [Ser73, Ch. VII] closely, though we adjust the notation, definitions, and order of presentation to be consistent with the rest of this book. (For example, Serre writes $2k$ for the weight instead of k.)

Mark Watkins and Lynn Walling made many helpful comments on Chapter 3.

Grant Information. This material is based upon work supported by the National Science Foundation under Grant No. 0400386.

Chapter 1

Modular Forms

This chapter introduces modular forms, which are the central object of study in this book. We first introduce the upper half plane and the group $\mathrm{SL}_2(\mathbb{Z})$, then recall some definitions from complex analysis. Next we define modular forms of level 1 followed by modular forms of general level.

This chapter assumes familiarity with basic number theory, group theory, and complex analysis.

1.1 Basic Definitions

Modular forms are certain types of functions on the *complex upper half plane*

$$\mathfrak{h} = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}.$$

The objects we will consider arise from the modular group $\mathrm{SL}_2(\mathbb{Z})$ of two-by-two integer matrices with determinant equal to one. This group

$$\mathrm{SL}_2(\mathbb{R}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} : ad - bc = 1, \text{ and } a, b, c, d \in \mathbb{R} \right\}$$

acts on $\mathfrak{h}$ via linear fractional transformations, as follows. If $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{R})$ and $z \in \mathfrak{h}$, then (see Exercise 1.1)

$$\gamma(z) = \frac{az + b}{cz + d} \in \mathfrak{h}. \tag{1.1.1}$$

Definition 1.1.1 (Modular Group). The *modular group* is the subgroup $\mathrm{SL}_2(\mathbb{Z})$ of $\mathrm{SL}_2(\mathbb{R})$ of matrices with integer entries. Thus $\mathrm{SL}_2(\mathbb{Z})$ is the group of all matrices $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $a, b, c, d \in \mathbb{Z}$ and $ad - bc = 1$.

For example, the matrices

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{1.1.2}$$

are both elements of $\mathrm{SL}_2(\mathbb{Z})$; the matrix S defines the function $z \mapsto -1/z$, and T the function $z \mapsto z+1$.

Theorem 1.1.2. *The group* $\mathrm{SL}_2(\mathbb{Z})$ *is generated by* S *and* T*.*

Proof. See e.g. [Ser73, §VII.1], which uses the fundamental domain $\mathcal{F}$ consisting of all elements of $\mathfrak{h}$ that satisfy $|z| \geq 1$ and $\mathrm{Re}(z) \leq 1/2$. □

In SAGE we compute the group $\mathrm{SL}_2(\mathbb{Z})$ and its generators as follows:

```
sage: G = SL(2,Z)
sage: print G
The modular group SL(2,Z)
sage: S, T = G.gens()
sage: S
[ 0 -1]
[ 1  0]
sage: T
[1 1]
[0 1]
```

Definition 1.1.3 (Holomorphic and Meromorphic). Let R be an open subset of $\mathbb{C}$. A function $f : R \to \mathbb{C}$ is *holomorphic* if f is complex differentiable at every point $z \in R$, i.e., for each $z \in R$ the limit $\lim_{h\to 0}(f(z+h) - f(z))/h$ exists, where h may approach 0 along any path. The function $f : R \to \mathbb{C} \cup \{\infty\}$ is *meromorphic* if it is holomorphic except (possibly) at a discrete set S of points in R, and at each $\alpha \in S$ there is a positive integer n such that $(z-\alpha)^n f(z)$ is holomorphic at α.

The function $f(z) = e^z$ is a holomorphic function on $\mathfrak{h}$ (in fact on all of $\mathbb{C}$). The function $1/(z-i)$ is meromorphic on $\mathfrak{h}$, and fails to be analytic at i.

Modular forms are holomorphic functions on $\mathfrak{h}$ that transform in a particular way under a subgroup of $\mathrm{SL}_2(\mathbb{Z})$. Before definining general modular forms, we define modular forms of level 1.

1.2 Modular Forms of Level 1

Definition 1.2.1 (Weakly Modular Function). A *weakly modular function* of weight $k \in \mathbb{Z}$ is a meromorphic function f on $\mathfrak{h}$ such that for all $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z})$ and all $z \in \mathfrak{h}$ we have

$$f(z) = (cz+d)^{-k} f(\gamma(z)). \tag{1.2.1}$$

The constant functions are weakly modular of weight 0. There are no nonzero weakly modular functions of odd weight (see Exercise 1.4), and it is by no means obvious that there are any weakly modular functions of even weight $k \geq 2$. The product of two weakly modular functions of weights k_1 and k_2 is a weakly modular function of weight $k_1 + k_2$ (see Exercise 1.3), so once we find some nonconstant weakly modular functions, we'll find many of them.

When k is even (1.2.1) has a possibly more conceptual interpretation; namely (1.2.1) is the same as

$$f(\gamma(z))d(\gamma(z))^{k/2} = f(z)dz^{k/2}.$$

Thus (1.2.1) simply says that the weight k "differential form" $f(z)dz^{k/2}$ is fixed under the action of every element of $\mathrm{SL}_2(\mathbb{Z})$.

Since $\mathrm{SL}_2(\mathbb{Z})$ is generated by the matrices S and T of (1.1.2), to show that a meromorphic function f on $\mathfrak{h}$ is a weakly modular function all we have to do is show that for all $z \in \mathfrak{h}$ we have

$$f(z+1) = f(z) \qquad \text{and} \qquad f(-1/z) = z^k f(z). \tag{1.2.2}$$

Suppose that f is a weakly modular function of some weight k. Then f might have a *Fourier expansion*, which we try to obtain as follows. Let $q = q(z) = e^{2\pi i z}$, which we view as a holomorphic function $\mathbb{C} \cup \infty \to D$, where D is the closed unit disk. Let D' be the punctured unit disk, i.e., D with the origin removed, and note that $q : \mathbb{C} \to D'$. By (1.2.2) we have $f(z+1) = f(z)$, so there is a set-theoretic map $F : D' \to \mathbb{C}$ such that for every $z \in \mathfrak{h}$ we have $F(q(z)) = f(z)$. This function F is thus a complex-valued function on the open unit disk. It may or may not be well behaved at 0.

Suppose that F is well-behaved at 0, namely that for some $m \in \mathbb{Z}$ and all q in a neighborhood of 0 we have the equality

$$F(q) = \sum_{n=m}^{\infty} a_n q^n.$$

If this is the case, we say that f is *meromorphic at* ∞. If, moreover, $m \geq 0$, then we say that f is *holomorphic at* ∞.

Definition 1.2.2 (Modular Function). A *modular function* of weight k is a weakly modular function of weight k that is meromorphic at ∞.

Definition 1.2.3 (Modular Form). A *modular form* of weight k (and level 1) is a modular function of weight k that is holomorphic on $\mathfrak{h}$ and at ∞.

If f is a modular form, then there are complex numbers a_n such that for all $z \in \mathfrak{h}$,

$$f(z) = \sum_{n=0}^{\infty} a_n q^n = \sum_{n=0}^{\infty} a_n e^{2\pi i n z}. \tag{1.2.3}$$

Proposition 1.2.4. *The above series converges for all $z \in \mathfrak{h}$.*

Proof. n The function $f(q)$ is holomorphic on D, so its Taylor series converges absolutely in D. See also [Ser73, §VII.4] for an explicit bound on the $|a_n|$. □

We set $f(\infty) = a_0$, since $q^{2\pi i z} \to 0$ as $z \to i\infty$, and the value of f at ∞ should be the value of F at 0, which is a_0 from the power series.

Definition 1.2.5 (Cusp Form). A *cusp form* of weight k (and level 1) is a modular form of weight k such that $f(\infty) = 0$, i.e., $a_0 = 0$.

1.3 Modular Forms of Any Level

We next define spaces of modular forms of arbitrary level. For example, when $k = 2$ these are closely related to elliptic curves and abelian varieties.

A *congruence subgroup* of $\mathrm{SL}_2(\mathbb{Z})$ is any subgroup that contains

$$\Gamma(N) = \ker(\mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z}))$$

for some N. The smallest such N is the *level* of Γ. For example,

$$\Gamma_1(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \pmod{N} \right\}$$

and

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \pmod{N} \right\}$$

are congruence subgroups of level N.

Let k be an integer. Define the *weight k right action* of $\mathrm{GL}_2(\mathbb{Q})$ on the set of functions $f : \mathfrak{h} \to \mathbb{C}$ as follows. If $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Q})$, let

$$f|[\gamma]_k = \det(\gamma)^{k-1}(cz + d)^{-k} f(\gamma(z)). \tag{1.3.1}$$

Proposition 1.3.1. *The action $f \mapsto f|[\gamma]_k$ is a right action of $\mathrm{GL}_2(\mathbb{Z})$ on the set of all functions $f : \mathfrak{h} \to \mathbb{C}$; in particular,*

$$f|[\gamma_1\gamma_2]_k = (f|[\gamma_1]_k)|[\gamma_2]_k,$$

Proof. See Exercise 1.7. □

Definition 1.3.2 (Weakly Modular Function). A *weakly modular function* of weight k for a congruence subgroup Γ is a meromorphic function $f : \mathfrak{h} \to \mathbb{C}$ such that $f|[\gamma]_k = f$ for all $\gamma \in \Gamma$.

A central object in the theory of modular forms (and modular symbols) is the set of all *cusps*

$$\mathbb{P}^1(\mathbb{Q}) = \mathbb{Q} \cup \{\infty\}.$$

The set of cusps for a congruence subgroup Γ is the set $C(\Gamma)$ of orbits of $\mathbb{P}^1(\mathbb{Q})$ under the action of Γ. (We will often identify elements of $C(\Gamma)$ with a representative element from the orbit.) For example, if $\Gamma = \mathrm{SL}_2(\mathbb{Z})$ there is exactly one orbit.

Lemma 1.3.3. *For any cusps $\alpha, \beta \in \mathbb{P}^1(\mathbb{Q})$ there exists $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\gamma(\alpha) = \beta$.*

Proof. This is Exercise 1.8. □

Proposition 1.3.4. *For any congruence subgroup Γ, the set $C(\Gamma)$ of cusps is finite.*

In order to define modular forms for general congruence subgroups we need to make sense of what it means for a function to be holomorphic on the *extended upper halfplane*

$$\mathfrak{h}^* = \mathfrak{h} \cup \mathbb{P}^1(\mathbb{Q}).$$

See [Shi94, §1.3–1.5] for a detailed description of the right topology to consider on $\mathfrak{h}^*$, so that $\mathfrak{h}^*$ is a compactification of $\mathfrak{h}$. In particular, a basis of neighborhoods for $\alpha \in \mathbb{Q}$ is given by the sets $\{\alpha\} \cup D$, where D is a disc in $\mathfrak{h}$ that is tangent to the real line at α.

Recall from Section 1.2 that a weakly modular function f on $\mathrm{SL}_2(\mathbb{Z})$ is holomorphic at ∞ if its q-expansion is of the form $\sum_{n=0}^{\infty} a_n q^n$.

In order to make sense of holomorphicity of a weakly modular function f for Γ at any $\alpha \in \mathbb{Q}$, we first prove a lemma.

Lemma 1.3.5. *If $f : \mathfrak{h} \to \mathbb{C}$ is a weakly modular function of weight k for a congruence subgroup Γ, and $\delta \in \mathrm{SL}_2(\mathbb{Z})$, then $f|[\delta]_k$ is a weakly modular function for $\delta^{-1}\Gamma\delta$.*

Proof. If $s = \delta^{-1}\gamma\delta \in \delta^{-1}\Gamma\delta$, then

$$(f|[\delta]_k)|[s]_k = f|[\delta s]_k = f|[\delta\delta^{-1}\gamma\delta]_k = f|[\gamma\delta]_k = f|[\delta]_k.$$

$\square$

Fix a weight k weakly modular function f for some congruence subgroup Γ, and suppose $\alpha \in \mathbb{Q}$. In Section 1.2 we constructed the q-expansion of f by using that $f(z) = f(z+1)$, which held since $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$. Unfortunately, there are congruence subgroups Γ such that $T \not\in \Gamma$. Moreover, even if we are interested only in modular forms for $\Gamma_1(N)$, where we have $T \in \Gamma_1(N)$ for all N, we have to consider q-expansions at infinity for modular forms on groups $\delta^{-1}\Gamma_1(N)\delta$, and these need not contain T. Fortunately, $T^N = \begin{pmatrix} 1 & N \\ 0 & 1 \end{pmatrix} \in \Gamma(N)$, so a congruence subgroup of level N contains T^N. Thus for our f we have $f(z+h) = f(z)$ for some positive integer h, and again when f is meromorphic at infinity we obtain a Fourier expansion

$$f(z) = \sum_{n=m}^{\infty} a_n q^{n/N}, \tag{1.3.2}$$

but instead it is in powers of the function $q^{1/N} = e^{2\pi i z/N}$. We say that f is holomorphic at ∞ if in (1.3.2) we have $m \geq 0$.

What about the other cusps $\alpha \in \mathbb{P}^1(\mathbb{Q})$? By Lemma 1.3.3 there is a $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\gamma(\infty) = \alpha$. We declare f to be *holomorphic at the cusp* α if the weakly modular function $f|[\gamma]_k$ is holomorphic at ∞.

Definition 1.3.6 (Modular Form). A *modular form* of integer weight k for a congruence subgroup Γ is a weakly modular function $f : \mathfrak{h}^* \to \mathbb{C}$ that is holomorphic on $\mathfrak{h}^*$.

Proposition 1.3.7. *A weakly modular function f of weight k for Γ is holomorphic at every element of $\mathbb{P}^1(\mathbb{Q})$ if is holomorphic at a set of representative elements for $C(\Gamma)$.*

Proof. Let $c_1, \ldots, c_n \in \mathbb{P}^1(\mathbb{Q})$ be representatives for the set of cusps for Γ. If $\alpha \in \mathbb{P}^1(\mathbb{Q})$ then there is $\gamma \in \Gamma$ such that $\alpha = \gamma(c_i)$ for some i. By hypothesis f is holomorphic at c_i, so if $\delta \in \mathrm{SL}_2(\mathbb{Z})$ is such that $\delta(\infty) = c_i$, then $f|[\delta]_k$ is holomorphic at ∞. Since f is a weakly modular function,

$$f|[\delta]_k = (f|[\gamma]_k)|[\delta]_k = f|[\gamma\delta]_k. \tag{1.3.3}$$

But $\gamma(\delta(\infty)) = \gamma(c_i) = \alpha$, so (1.3.3) implies that f is holomorphic at α. $\square$

1.4 Exercises

1.1 Suppose $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ is a matrix with real entries and positive determinant. Prove that if $z \in \mathbb{C}$ is a complex number with positive imaginary part, then the imaginary part of $\gamma(z) = (az+b)/(cz+d)$ is also positive.

1.2 (a) Prove that a polynomial is an analytic function on $\mathbb{C}$.

(b) Prove that a rational function (quotient of two polynomials) is a meromorphic function on $\mathbb{C}$.

1.3 Suppose f and g are weakly modular functions with $f \neq 0$.

(a) Prove that the product fg is a weakly modular function.

(b) Prove that $1/f$ is a weakly modular function.

(c) If f and g are modular functions, show that fg is a modular function.

(d) If f and g are modular forms, show that fg is a modular form.

1.4 Suppose f is a weakly modular function of odd weight k. Show that $f = 0$.

1.5 Prove that $\mathrm{SL}_2(\mathbb{Z}) = \Gamma_0(1) = \Gamma_1(1) = \Gamma(1)$.

1.6 (a) Prove that $\Gamma_1(N)$ is a group.

(b) Prove that $\Gamma_1(N)$ has finite index in $\mathrm{SL}_2(\mathbb{Z})$ (Hint: it contains the kernel of the homomorphism $\mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$.)

(c) Prove that $\Gamma_0(N)$ has finite index in $\mathrm{SL}_2(\mathbb{Z})$.

1.7 Let k be an integer, and for any function $f : \mathfrak{h}^* \to \mathbb{C}$ and $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Q})$, set $f|[\gamma]_k(z) = (cz+d)^{-k} f(\gamma(z))$. Prove that if $\gamma_1, \gamma_2 \in \mathrm{SL}_2(\mathbb{Z})$, then for all $z \in \mathfrak{h}^*$ we have

$$f|[\gamma_1\gamma_2]_k(z) = ((f|[\gamma_1]_k)|[\gamma_2]_k)(z).$$

1.8 Prove that for any $\alpha, \beta \in \mathbb{P}^1(\mathbb{Q})$, there exists $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\gamma(\alpha) = \beta$.

Chapter 2

Modular Forms of Level One

In this chapter we study in more detail the structure of level 1 modular forms.

We assume that you know some complex analysis (e.g., the residue theorem) and linear algebra.

2.1 Examples of Modular Forms of Level 1

In this section you will finally see some examples of modular forms of level 1! We will first introduce the Eisenstein series, one of each weight, then define Δ, which is a cusp form of weight 12. In Section 2.2 we will prove the structure theorem, which says that using addition and multiplication of these forms, we can generate all modular forms of level 1.

For an even integer $k \geq 4$, the *non-normalized weight k Eisenstein series* is

$$G_k(z) = \sum_{m,n\in\mathbb{Z}}^{*} \frac{1}{(mz+n)^k},$$

where for a given z, the sum is over all $m, n \in \mathbb{Z}$ such that $mz + n \neq 0$.

Proposition 2.1.1. *The function $G_k(z)$ is a modular form of weight k, i.e., $G_k \in M_k(\mathrm{SL}_2(\mathbb{Z}))$.*

Proof. See [Ser73, § VII.2.3] for a proof that $G_k(z)$ defines a holomorphic function on $\mathfrak{h}^*$. To see that G_k is modular, observe that

$$G_k(z+1) = \sum^{*} \frac{1}{(m(z+1)+n)^k} = \sum^{*} \frac{1}{(mz+(n+m))^k} = \sum^{*} \frac{1}{(mz+n)^k},$$

and

$$G_k(-1/z) = \sum^{*} \frac{1}{(-m/z+n)^k} = \sum^{*} \frac{z^k}{(-m+nz)^k} = z^k \sum^{*} \frac{1}{(mz+n)^k} = z^k G_k(z).$$

□

Proposition 2.1.2. *$G_k(\infty) = 2\zeta(k)$, where ζ is the Riemann zeta function.*

Proof. Taking the limit as $z \to i\infty$ in the definition of $G_k(z)$, we obtain $\sum_{n\in\mathbb{Z}}^{*} \frac{1}{n^k}$, since the terms involving z all go to 0 as $z \mapsto i\infty$. This sum is twice $\zeta(k) = \sum_{n\geq 1} \frac{1}{n^k}$. □

For example,

$$G_4(\infty) = 2\zeta(4) = \frac{1}{3^2 \cdot 5}\pi^4$$

and

$$G_6(\infty) = 2\zeta(6) = \frac{2}{3^3 \cdot 5 \cdot 7}\pi^6.$$

2.1.1 The Cusp Form Δ

Suppose $E = \mathbb{C}/\Lambda$ is an elliptic curve over $\mathbb{C}$, viewed as a quotient of $\mathbb{C}$ by a lattice $\Lambda = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$, with $\omega_1/\omega_2 \in \mathfrak{h}$. Then

$$\wp_\Lambda(u) = \frac{1}{u^2} + \sum_{k=4,\ (k \text{ even})}^{\infty} (k-1)G_k(\omega_1/\omega_2)u^{k-2},$$

and

$$(\wp')^2 = 4\wp^3 - 60G_4(\omega_1/\omega_2)\wp - 140G_6(\omega_1/\omega_2).$$

The discriminant of the cubic $4x^3 - 60G_4(\omega_1/\omega_2)x - 140G_6(\omega_1/\omega_2)$ is $16\Delta(\omega_1/\omega_2)$, where

$$\Delta = (60G_4)^3 - 27(140G_6)^2$$

is a cusp form of weight 12.

Lemma 2.1.3. *The cusp form Δ has a 0 only at ∞.*

Proof. Let ω_1, ω_2 be as above. Since E is an elliptic curve, $\Delta(\omega_1/\omega_2) \neq 0$. □

2.1.2 Fourier Expansions of Eisenstein Series

Recall from (1.2.3) that elements f of $M_k(\mathrm{SL}_2(\mathbb{Z}))$ can be expressed as formal power series in terms of $q(z) = e^{2\pi i z}$, and that this expansion is called the Fourier expansion of f. The following proposition gives the Fourier expansion of the Eisenstein series $G_k(z)$.

Definition 2.1.4 (Sigma). For any integer $t \geq 0$ and any positive integer n, let

$$\sigma_t(n) = \sum_{1\leq d \mid n} d^t$$

be the sum of the tth powers of the positive divisors of n. Also, let $\sigma(n) = \sigma_0(n)$. For example, if p is prime then $\sigma_t(p) = 1 + p^t$.

Proposition 2.1.5. *For every even integer $k \geq 4$, we have*

$$G_k(z) = 2\zeta(k) + 2 \cdot \frac{(2\pi i)^k}{(k-1)!} \cdot \sum_{n=1}^{\infty} \sigma_{k-1}(n) q^n.$$

Proof. See [Ser73, §VII.4], which uses a series of clever manipulations of series, starting with the identity

$$\pi \cot(\pi z) = \frac{1}{z} + \sum_{m=1}^{\infty} \left(\frac{1}{z+m} + \frac{1}{z-m} \right).$$

□

From a computational point of view, the q-expansion for G_k from Proposition 2.1.5 is unsatisfactory, because it involves transcendental numbers. To understand more clearly what is going on, we introduce the *Bernoulli numbers* B_n for $n \geq 0$ *defined* by the following equality of formal power series:

$$\frac{x}{e^x - 1} = \sum_{n=0}^{\infty} B_n \frac{x^n}{n!}. \tag{2.1.1}$$

Expanding the power series on the left we have

$$\frac{x}{e^x - 1} = 1 - \frac{x}{2} + \frac{x^2}{12} - \frac{x^4}{720} + \frac{x^6}{30240} - \frac{x^8}{1209600} + \cdots$$

As this expansion suggests, the Bernoulli numbers B_n with $n > 1$ odd are 0 (see Exercise 1.6). Expanding the series further, we obtain the following table:

$$B_0 = 1, \quad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \quad B_8 = -\frac{1}{30},$$

$$B_{10} = \frac{5}{66}, \quad B_{12} = -\frac{691}{2730}, \quad B_{14} = \frac{7}{6}, \quad B_{16} = -\frac{3617}{510}, \quad B_{18} = \frac{43867}{798},$$

$$B_{20} = -\frac{174611}{330}, \quad B_{22} = \frac{854513}{138}, \quad B_{24} = -\frac{236364091}{2730}, \quad B_{26} = \frac{8553103}{6}.$$

Use the `bernoulli` command to compute Bernoulli numbers in SAGE.

```
sage: bernoulli(12)
-691/2730
sage: bernoulli(50)
495057205241079648212477525/66
```

For us, the significance of the Bernoulli numbers is their connection with values of ζ at positive even integers.

Proposition 2.1.6. *If $k \geq 2$ is an even integer, then*

$$\zeta(k) = -\frac{(2\pi i)^k}{2 \cdot k!} \cdot B_k.$$

Proof. The proof in [Ser73, §VII.4] involves manipulating a power series expansion for $z\cot(z)$. □

Definition 2.1.7 (Normalized Eisenstein Series). The *normalized Eisenstein series* of even weight $k \geq 4$ is

$$E_k = \frac{(k-1)!}{2\cdot(2\pi i)^k}\cdot G_k$$

Combining Propositions 2.1.5 and 2.1.6 we see that

$$E_k = -\frac{B_k}{2k} + q + \sum_{n=2}^{\infty}\sigma_{k-1}(n)q^n. \tag{2.1.2}$$

It is thus now simple to explicitly write down Eisenstein series (see Exercise 2.1).

Warning 2.1.8. Our series E_k is normalized so that the coefficient of q is 1, but often in the literature E_k is normalized so that the constant coefficient is 1. We use the normalization with the coefficient of q equal to 1, because then the eigenvalue of the nth Hecke operator (see Section 2.4) is the coefficient of q^n. Our normalization is also convenient when considering congruences between cusp forms and Eisenstein series.

2.2 Structure Theorem For Level 1 Modular Forms

In this section we describe a structure theorem for modular forms of level 1. If f is a nonzero meromorphic function on $\mathfrak{h}$ and $w \in \mathfrak{h}$, let $\mathrm{ord}_w(f)$ be the largest integer n such that $f/(w-z)^n$ is holomorphic at w. If $f = \sum_{n=m}^{\infty} a_n q^n$ with $a_m \neq 0$, let $\mathrm{ord}_\infty(f) = m$. We will use the following theorem to give a presentation for the vector space of modular forms of weight k; this presentation will allow us to obtain an algorithm to compute a basis for this space.

Let $\mathcal{F}$ be the subset of $\mathfrak{h}$ of numbers z with $|z| \geq 1$ and $\mathrm{Re}(z) \leq 1/2$. This is the standard fundamental domain for $\mathrm{SL}_2(\mathbb{Z})$.

Theorem 2.2.1 (Valence Formula). *Suppose $f \in M_k(\mathrm{SL}_2(\mathbb{Z}))$ is nonzero. Then*

$$\mathrm{ord}_\infty(f) + \frac{1}{2}\mathrm{ord}_i(f) + \frac{1}{3}\mathrm{ord}_\rho(f) + \sideset{}{^*}\sum_{w\in D}\mathrm{ord}_w(f) = \frac{k}{12},$$

where $\sideset{}{^*}\sum_{w\in D}$ *is the sum over elements of $\mathcal{F}$ other than i or ρ.*

Proof. Serre proves this theorem in [Ser73, §VII.3] using the residue theorem from complex analysis. □

Let $M_k = M_k(\mathrm{SL}_2(\mathbb{Z}))$ denote the complex vector space of modular forms of weight k for $\mathrm{SL}_2(\mathbb{Z})$, and let $S_k = S_k(\mathrm{SL}_2(\mathbb{Z}))$ denote the subspace of weight k cusp forms for $\mathrm{SL}_2(\mathbb{Z})$. We have an exact sequence

$$0 \to S_k \to M_k \to \mathbb{C}$$

that sends $f \in M_k$ to $f(\infty)$. When $k \geq 4$ is even, the space M_k contains the Eisenstein series G_k and $G_k(\infty) = 2\zeta(k) \neq 0$, so the map $M_k \to \mathbb{C}$ is surjective, so the following sequence is exact:

$$0 \to S_k \to M_k \to \mathbb{C} \to 0 \qquad (\text{when } k \geq 4 \text{ is even})$$

Thus when $k \geq 4$ is even $\dim(S_k) = \dim(M_k) - 1$ hence

$$M_k = S_k \oplus \mathbb{C}G_k.$$

Proposition 2.2.2. *For $k < 0$ and $k = 2$, we have $M_k = 0$.*

Proof. Suppose $f \in M_k$ is nonzero yet $k = 2$ or $k < 0$. By Theorem 2.2.1,

$$\mathrm{ord}_\infty(f) + \frac{1}{2}\,\mathrm{ord}_i(f) + \frac{1}{3}\,\mathrm{ord}_\rho(f) + \sum_{w\in D}^{*} \mathrm{ord}_w(f) = \frac{k}{12} \leq 1/6.$$

This is impossible because each quantity on the left-hand side is nonnegative so whatever the sum is, it is too big (or 0, in which $k = 0$). $\square$

Theorem 2.2.3. *Multiplication by Δ defines an isomorphism $M_{k-12} \to S_k$.*

Proof. (We follow [Ser73, §VII.3.2] closely.) We apply Theorem 2.2.1 to G_4 and G_6. If $f = G_4$, then

$$\mathrm{ord}_\infty(f) + \frac{1}{2}\,\mathrm{ord}_i(f) + \frac{1}{3}\,\mathrm{ord}_\rho(f) + \sum_{w\in D}^{*} \mathrm{ord}_w(f) = \frac{4}{12} = \frac{1}{3},$$

with the ords all nonnegative, so $\mathrm{ord}_\rho(G_4) = 1$ and $\mathrm{ord}_w(G_4) = 0$ for all $w \neq \rho$. Likewise $\mathrm{ord}_i(G_6) = 1$ and $\mathrm{ord}_w(G_6) = 0$ for all $w \neq i$. Thus $\Delta(i) \neq 0$, so Δ is not identically 0 (this also follows from Lemma 2.1.3 above). Since Δ has weight 12 and $\mathrm{ord}_\infty(\Delta) \geq 1$, Theorem 2.2.1 implies that Δ has a simple zero at ∞ and does not vanish on $\mathfrak{h}$. Thus if $f \in S_k$ and we let $g = f/\Delta$, then g is holomorphic and satisfies the appropriate transformation formula, so g is a modular form of weight $k - 12$. $\square$

Corollary 2.2.4. *For $k = 0, 4, 6, 8, 10, 14$, the vector space M_k has dimension 1, with basis 1, G_4, G_6, E_8, E_{10}, and E_{14}, respectively, and $S_k = 0$.*

Proof. Combining Proposition 2.2.2 with Theorem 2.2.3 we see that the spaces M_k for $k \leq 10$ can not have dimension bigger than 1, since then $M_{k'} \neq 0$ for some $k' < 0$. Also M_{14} has dimension at most 1, since M_2 has dimension 0. Each of the indicated spaces of weight ≥ 4 contains the indicated Eisenstein series, so has dimension 1, as claimed. $\square$

Corollary 2.2.5. $\dim M_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ \lfloor k/12 \rfloor & \text{if } k \equiv 2 \pmod{12}, \\ \lfloor k/12 \rfloor + 1 & \text{if } k \not\equiv 2 \pmod{12}, \end{cases}$ *where* $\lfloor x \rfloor$ *is the biggest integer* $\leq x$.

Proof. As we have seen above, the formula is true when $k \leq 12$. By Theorem 2.2.3, the dimension increases by 1 when k is replaced by $k + 12$. □

Theorem 2.2.6. *The space* M_k *has as basis the modular forms* $G_4^a G_6^b$*, where* a, b *are all pairs of nonnegative integers such that* $4a + 6b = k$.

Proof. We first prove by induction that the modular forms $G_4^a G_6^b$ generate M_k, the cases $k \leq 12$ being clear (e.g., when $k = 0$ we have $a = b = 0$ and basis 1). Choose some pair of integers a, b such that $4a + 6b = k$ (it is an elementary exercise to show these exist). The form $g = G_4^a G_6^b$ is not a cusp form, since it is nonzero at ∞. Now suppose $f \in M_k$ is arbitrary. Since $M_k = S_k \oplus \mathbb{C}G_k$, there is $\alpha \in \mathbb{C}$ such that $f - \alpha g \in S_k$. Then by Theorem 2.2.3, there is $h \in M_{k-12}$ such that $f - \alpha g = \Delta h$. By induction, h is a polynomial in G_4 and G_6 of the required type, and so is Δ, so f is as well.

Suppose there is a nontrivial linear relation between the $G_4^a G_6^b$ for a given k. By multiplying the linear relation by a suitable power of G_4 and G_6, we may assume that that we have such a nontrivial relation with $k \equiv 0 \pmod{12}$. Now divide the linear relation by $G_6^{k/12}$ to see that G_4^3/G_6^2 satisfies a polynomial with coefficients in $\mathbb{C}$. Hence G_4^3/G_6^2 is a root of a polynomial, hence a constant, which is a contradiction since the q-expansion of G_4^3/G_6^2 is not constant. □

Algorithm 2.2.7 (Basis for M_k). *Given integers* n *and* k*, this algorithm computes a basis of* q*-expansions for the complex vector space* M_k *mod* q^n*. The* q*-expansions output by this algorithm have coefficients in* $\mathbb{Q}$.

1. [Simple Case] If $k = 0$ output the basis with just 1 in it, and terminate; otherwise if $k < 4$ or k is odd, output the empty basis and terminate.

2. [Power Series] Compute E_4 and E_6 mod q^n using the formula from (2.1.2) and the definition (2.1.1) of Bernoulli numbers.

3. [Initialize] Set $b = 0$.

4. [Enumerate Basis] For each integer b between 0 and $\lfloor k/6 \rfloor$, compute $a = (k - 6b)/4$. If a is an integer, compute and output the basis element $E_4^a E_6^b$ mod q^n. When we compute, e.g., E_4^a, do the computation by finding $E_4^m \pmod{q^n}$ for each $m \leq a$, and save these intermediate powers, so they can be reused later, and likewise for powers of E_6.

Proof. This is simply a translation of Theorem 2.2.6 into an algorithm, since E_k is a nonzero scalar multiple of G_k. That the q-expansions have coefficients in $\mathbb{Q}$ follows from (2.1.2). □

Example 2.2.8. We compute a basis for M_{24}, which is the space with smallest weight whose dimension is bigger than 1. It has as basis E_4^6, $E_4^3E_6^2$, and E_6^4, whose explicit expansions are

$$\begin{aligned}
E_4^6 &= \frac{1}{191102976000000} + \frac{1}{132710400000}q + \frac{203}{44236800000}q^2 + \cdots \\
E_4^3E_6^2 &= \frac{1}{3511517184000} - \frac{1}{12192768000}q - \frac{377}{4064256000}q^2 + \cdots \\
E_6^4 &= \frac{1}{64524128256} - \frac{1}{32006016}q + \frac{241}{10668672}q^2 + \cdots
\end{aligned}$$

In Section 2.3, we will discuss properties of the reduced row echelon form of any basis for M_k, which have better properties than the above basis.

2.3 The Victor Miller Basis

Lemma 2.3.1 (Victor Miller). *The space S_k has a basis $f_1, \ldots, f_d$ such that if $a_i(f_j)$ is the ith coefficient of f_j, then $a_i(f_j) = \delta_{i,j}$ for $i = 1, \ldots, d$. Moreover the f_j all lie in $\mathbb{Z}[[q]]$.*

This is a straightforward construction involving E_4, E_6 and Δ. The following proof very closely follows [Lan95, Ch. X, Thm. 4.4], which is in turn follows the first lemma of Victor Miller's thesis.

Proof. Let $d = \dim S_k$. Since $B_4 = -1/30$ and $B_6 = 1/42$, we note that

$$F_4 = -\frac{8}{B_4} \cdot E_4 = 1 + 240q + 2160q^2 + 6720q^3 + 17520q^4 + \cdots$$

and

$$F_6 = -\frac{12}{B_6} \cdot E_6 = 1 - 504q - 16632q^2 - 122976q^3 - 532728q^4 + \cdots$$

have q-expansions in $\mathbb{Z}[[q]]$ with leading coefficient 1. Choose integers $a, b \geq 0$ such that

$$4a + 6b \leq 14 \qquad \text{and} \qquad 4a + 6b \equiv k \pmod{12},$$

with $a = b = 0$ when $k \equiv 0 \pmod{12}$, and let

$$g_j = \Delta^j F_6^{2(d-j)+b} F_4^a, \qquad \text{for } j = 1, \ldots, d.$$

Then it is elementary to check that g_j has weight k, and

$$a_j(g_j) = 1, \qquad \text{and} \qquad a_i(g_j) = 0 \qquad \text{when} \qquad i < j.$$

Hence the g_j are linearly independent over $\mathbb{C}$, and thus form a basis for S_k. Since F_4, F_6, and Δ are all in $\mathbb{Z}[[q]]$, so are the g_j. The f_i may then be constructed from the g_j by Gauss elimination. The coefficients of the resulting power series lie in $\mathbb{Z}$ because each time we clear a column we use the power series g_j whose leading coefficient is 1 (so no denominators are introduced). $\square$

Remark 2.3.2. The basis coming from Victor Miller's lemma is canonical, since it is just the reduced row echelon form of any basis. Also the *integral* linear combinations are precisely the modular forms of level 1 with integral q-expansion.

We extend the Victor Miller basis to all M_k by taking a multiple of G_k with constant term 1, and subtracting off the f_i from the Victor Miller basis so that the coefficients of $q, q^2, \ldots q^d$ of the resulting expansion are 0. We call the extra basis element f_0.

Example 2.3.3. If $k = 24$, then $d = 2$. Choose $a = b = 0$, since $k \equiv 0 \pmod{12}$. Then

$$g_1 = \Delta F_6^2 = q - 1032q^2 + 245196q^3 + 10965568q^4 + 60177390q^5 - \cdots$$

and

$$g_2 = \Delta^2 = q^2 - 48q^3 + 1080q^4 - 15040q^5 + \cdots$$

We let $f_2 = g_2$ and

$$f_1 = g_1 + 1032g_2 = q + 195660q^3 + 12080128q^4 + 44656110q^5 - \cdots .$$

Example 2.3.4. When $k = 36$, the Victor Miller basis, including f_0, is

$$\begin{aligned}
f_0 &= 1 + & 6218175600q^4 &+ 15281788354560q^5 + \cdots \\
f_1 &= \quad q + & 57093088q^4 &+ \quad 37927345230q^5 + \cdots \\
f_2 &= \quad\quad q^2 + & 194184q^4 &+ \quad\quad 7442432q^5 + \cdots \\
f_3 &= \quad\quad\quad q^3 - & 72q^4 &+ \quad\quad\quad 2484q^5 + \cdots
\end{aligned}$$

Remark 2.3.5. If you wish to write $f \in M_k$ as a polynomial in E_4 and E_6, then it is wasteful to compute the Victor Miller basis. Instead, use the upper triangular basis $\Delta^j F_6^{2(d-j)+a} F_4^b$, and match coefficients from q^0 to q^d.

2.4 Hecke Operators

For any positive integer n, let

$$S_n = \left\{ \begin{pmatrix} a & b \\ 0 & d \end{pmatrix} \in M_2(\mathbb{Z}) \ : \ a \geq 1, \ ad = n, \text{ and } 0 \leq b < d \right\}.$$

Note that the set S_n is in bijection with the set of sublattices of $\mathbb{Z}^2$ of index n, where $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ corresponds to $L = \mathbb{Z} \cdot (a, b) + \mathbb{Z} \cdot (0, d)$, as one can see, e.g., by using Hermite normal form (the analogue of reduced row echelon form over $\mathbb{Z}$).

Recall from (1.3.1) that if $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Q})$, then

$$f|[\gamma]_k = \det(\gamma)^{k-1}(cz + d)^{-k} f(\gamma(z)).$$

Definition 2.4.1 (Hecke Operator $T_{n,k}$). The nth Hecke operator $T_{n,k}$ of weight k is the operator on functions on $\mathfrak{h}$ defined by

$$T_{n,k}(f) = \sum_{\gamma \in S_n} f|[\gamma]_k.$$

Remark 2.4.2. It would make more sense to write $T_{n,k}$ on the right, e.g., $f|T_{n,k}$, since $T_{n,k}$ is defined using a right group action. However, if n, m are integers, then $T_{n,k}$ and $T_{m,k}$ commute (by Proposition 2.4.4 below), so it does not matter whether we consider the Hecke operators as acting on the right or left.

Proposition 2.4.3. *If f is a weakly modular function of weight k, then so is $T_{n,k}(f)$, and if f is also a modular function, then so is $T_{n,k}(f)$.*

Proof. Suppose $\gamma \in \mathrm{SL}_2(\mathbb{Z})$. Since γ induces an automorphism of $\mathbb{Z}^2$, the set

$$S_n \cdot \gamma = \{\delta\gamma : \delta \in S_n\}$$

is also in bijection with the sublattices of $\mathbb{Z}^2$ of index n. Then for each element $\delta\gamma \in S_n \cdot \gamma$, there is $\sigma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\sigma\delta\gamma \in S_n$ (the element σ transforms $\delta\gamma$ to Hermite normal form), and the set of elements $\sigma\delta\gamma$ is equal to S_n. Thus

$$T_{n,k}(f) = \sum_{\sigma\delta\gamma \in S_n} f|[\sigma\delta\gamma]_k = \sum_{\delta \in S_n} f|[\delta\gamma]_k = T_{n,k}(f)|[\gamma]_k.$$

Since f is holomorphic on $\mathfrak{h}$, each $f|[\gamma]_k$ is holomorphic on $\mathfrak{h}$. A finite sum of holomorphic functions is holomorphic, so $T_{n,k}(f)$ is holomorphic.

□

We will frequently drop k from the notation in $T_{n,k}$, since the weight k is implicit in the modular function to which we apply the Hecke operator. Thus we henceforth make the convention that if we write $T_n(f)$ and f is modular, then we mean $T_{n,k}(f)$, where k is the weight of f.

Proposition 2.4.4. *On weight k modular functions we have*

$$T_{mn} = T_n T_m \qquad \textit{if } (n,m) = 1, \tag{2.4.1}$$

and

$$T_{p^n} = T_{p^{n-1}} T_p - p^{k-1} T_{p^{n-2}}, \qquad \textit{if } p \textit{ is prime.} \tag{2.4.2}$$

Proof. Let L be a lattice of index mn. The quotient $\mathbb{Z}^2/L$ is an abelian group of order mn, and $(m,n) = 1$, so $\mathbb{Z}^2/L$ decomposes uniquely as a direct sum of a subgroup order m with a subgroup of order n. Thus there exists a unique lattice L' such that $L \subset L' \subset \mathbb{Z}^2$, and L' has index m in $\mathbb{Z}^2$. The lattice L' corresponds to an element of S_m, and the index n subgroup $L \subset L'$ corresponds to multiplying that element on the right by some uniquely determined element of S_n. We thus have

$$\mathrm{SL}_2(\mathbb{Z}) \cdot S_m \cdot S_n = \mathrm{SL}_2(\mathbb{Z}) \cdot S_{mn},$$

i.e., the set products of elements in S_m with elements of S_n equal the elements of S_{mn}, up to $\mathrm{SL}_2(\mathbb{Z})$-equivalence. It then follows from the definitions that for any f, we have $T_{mn}(f) = T_n(T_m(f))$.

We will show that $T_{p^n} + p^{k-1}T_{p^{n-2}} = T_pT_{p^{n-1}}$. Suppose f is a weight k weakly modular function. Using that $f|[p]_k = (p^2)^{k-1}p^{-k}f = p^{k-2}f$, we have

$$\sum_{x\in S_{p^n}} f|[x]_k \ + \ p^{k-1}\sum_{x\in S_{p^{n-2}}} f|[x]_k = \sum_{x\in S_{p^n}} f|[x]_k \ + p\sum_{x\in pS_{p^{n-2}}} f|[x]_k.$$

Also

$$T_pT_{p^{n-1}}(f) = \sum_{y\in S_p}\sum_{x\in S_{p^{n-1}}} f|[x]_k|[y]_k = \sum_{x\in S_{p^{n-1}}\cdot S_p} f|[x]_k.$$

Thus it suffices to show that S_{p^n} union p copies of $pS_{p^{n-2}}$ is equal to $S_{p^{n-1}}\cdot S_p$, where we consider elements up to left $\mathrm{SL}_2(\mathbb{Z})$-equivalence (i.e., the left action of $\mathrm{SL}_2(\mathbb{Z})$).

Suppose L is a sublattice of $\mathbb{Z}^2$ of index p^n, so L corresponds to an element of S_{p^n}. First suppose L is not contained in $p\mathbb{Z}^2$. Then the image of L in $\mathbb{Z}^2/p\mathbb{Z}^2 = (\mathbb{Z}/p\mathbb{Z})^2$ is of order p, so if $L' = p\mathbb{Z}^2 + L$, then $[\mathbb{Z}^2 : L'] = p$ and $[L : L'] = p^{n-1}$, and L' is the only lattice with this property. Second suppose that $L \subset p\mathbb{Z}^2$ if of index p^n, and that $x \in S_{p^n}$ corresponds to L. Then every one of the $p+1$ lattices $L' \subset \mathbb{Z}^2$ of index p contains L. Thus there are $p+1$ chains $L \subset L' \subset \mathbb{Z}^2$ with $[\mathbb{Z}^2 : L'] = p$.

The chains $L \subset L' \subset \mathbb{Z}^2$ with $[\mathbb{Z}^2 : L'] = p$ and $[\mathbb{Z}^2 : L] = p^{n-1}$ are in bijection with the elements of $S_{p^{n-1}}\cdot S_p$. On the other hand the union of S_{p^n} with p copies of $pS_{p^{n-2}}$ corresponds to the lattices L of index p^n, but with those that contain $p\mathbb{Z}^2$ counted $p+1$ times. The structure of the set of chains $L \subset L' \subset \mathbb{Z}^2$ that we derived in the previous paragraph gives the result. □

Corollary 2.4.5. *The Hecke operator T_{p^n}, for prime p, is a polynomial in T_p. If n, m are any integers then $T_nT_m = T_mT_n$.*

Proof. The first statement is clear from (2.4.2), and this gives commutativity when m and n are both powers of p. Combining this with (2.4.1) gives the second statement in general. □

Proposition 2.4.6. *Suppose $f = \sum_{n\in\mathbb{Z}} a_nq^n$ is a modular function of weight k. Then*

$$T_n(f) = \sum_{m\in\mathbb{Z}}\left(\sum_{1\leq d\,|\,\gcd(n,m)} d^{k-1}a_{mn/d^2}\right)q^m.$$

In particular, if $n = p$ is prime, then

$$T_p(f) = \sum_{m\in\mathbb{Z}}\left(a_{mp} + p^{k-1}a_{m/p}\right)q^m,$$

where $a_{m/p} = 0$ if $m/p \notin \mathbb{Z}$.

The proposition is not that difficult to prove (or at least the proof is easy to follow), and is proved in [Ser73, §VII.5.3] by writing out $T_n(f)$ explicitly and using that $\sum_{0\leq b<d} e^{2\pi ibm/d}$ is d if $d \mid m$ and 0 otherwise. A corollary of Proposition 2.4.6 is that T_n preserves M_k and S_k.

Corollary 2.4.7. *The Hecke operators preserve M_k and S_k.*

Remark 2.4.8. We knew this already—for M_k it's Proposition 2.4.3, and for S_k it's easy to show directly that if $f(i\infty) = 0$ then $T_n f$ also vanishes at $i\infty$.

Example 2.4.9. Recall that

$$E_4 = \frac{1}{240} + q + 9q^2 + 28q^3 + 73q^4 + 126q^5 + 252q^6 + 344q^7 + \cdots .$$

Using the formula of Proposition 2.4.6, we see that

$$T_2(E_4) = (1/240 + 2^3 \cdot (1/240)) + 9q + (73 + 2^3 \cdot 1)q^2 + \cdots = 9E_4.$$

Since M_k has dimension 1, and we have proved that T_2 preserves M_k, we know that T_2 acts as a scalar. Thus we know just from the constant coefficient of $T_2(E_4)$ that $T_2(E_4) = 9E_4$. More generally, $T_p(E_4) = (1 + p^3)E_4$. In fact for any k one has that

$$T_n(E_k) = \sigma_{k-1}(n)E_k,$$

for any integer $n \geq 1$ and even weight $k \geq 4$.

Example 2.4.10. By Corollary 2.4.7, the Hecke operators T_n also preserve the subspace S_k of M_k. Since S_{12} has dimension 1 (spanned by Δ), we see that Δ is an eigenvector for every T_n. Since the coefficient of q in the q-expansion of Δ is 1, the eigenvalue of T_n on Δ is the nth coefficient of Δ. Moreover the function $\tau(n)$ that gives the nth coefficient of Δ is a multiplicative function. Likewise, one can show that the series E_k are eigenvectors for all T_n, and because in this book we normalize E_k so that the coefficient of q is 1, the eigenvalue of T_n on E_k is the coefficient $\sigma_{k-1}(n)$ of q^n.

2.5 Computing Hecke Operators

In this section we describe a simple algorithm for computing matrices of Hecke operators on M_k.

Algorithm 2.5.1 (Hecke Operator). *This algorithm computes a matrix for the Hecke operator T_n on the Victor Miller basis for M_k.*

1. [Compute dimension] Set $d = \dim(S_k)$, which we compute using Corollary 2.2.5.
2. [Compute basis] Using the algorithm implicit in Lemma 2.3.1, compute a basis $f_0, \ldots, f_d$ for M_k modulo q^{dn+1}.

3. [Compute Hecke operator] Using the formula from Proposition 2.4.6, compute $T_n(f_i) \pmod{q^{d+1}}$ for each i.
4. [Write in terms of basis] The elements $T_n(f_i) \pmod{q^{d+1}}$ uniquely determine linear combinations of $f_0, f_1, \ldots, f_d \pmod{q^d}$. These linear combinations are trivial to find once we have computed $T_n(f_i) \pmod{q^{d+1}}$, since the basis of f_i are in reduced row echelon form, so the combinations are just the first few coefficients of the power series $T_n(f_i)$.
5. [Write down matrix] The matrix of T_n acting from the right is the matrix whose columsn are the linear combinations found in the previous step, i.e., whose columns are the coefficients of $T_n(f_i)$.

Proof. First note that we compute a modular form f modulo q^{dn+1} in order to compute $T_n(f)$ modulo q^{d+1}. This follows from Proposition 2.4.6, since in the formula the dth coefficient of $T_n(f)$ involves only a_{dn}, and smaller-indexed coefficients of f. The uniqueness assertion of Step 4 follows from Lemma 2.3.1 above. □

Example 2.5.2. This is the Hecke operator T_2 on M_{36} is:

$$\begin{pmatrix} 34359738369 & 0 & 6218175600 & 9026867482214400 \\ 0 & 0 & 34416831456 & 5681332472832 \\ 0 & 1 & 194184 & -197264484 \\ 0 & 0 & -72 & -54528 \end{pmatrix}$$

It has characteristic polynomial

$$(x - 34359738369) \cdot (x^3 - 139656x^2 - 59208339456x - 1467625047588864),$$

where the cubic factor is irreducible.

Conjecture 2.5.3 (Maeda). *The characteristic polynomial of T_2 on S_k is irreducible for any k.*

Kevin Buzzard even observed that in many specific cases the Galois group of the characteristic polynomial of T_2 is the full symmetric group (see [Buz96]). See also [FJ02] for more evidence for Maeda's conjecture.

2.5.1 A Conjecture about Complexity

Let

$$\begin{aligned} \Delta &= \sum_{n=1}^{\infty} \tau(n) q^n \\ &= q - 24q^2 + 252q^3 - 1472q^4 + 4830q^5 - 6048q^6 - 16744q^7 \\ &\quad + 84480q^8 - 113643q^9 - 115920q^{10} + 534612q^{11} - \\ &\quad 370944q^{12} - 577738q^{13} + 401856q^{14} + 1217160q^{15} + \\ &\quad 987136q^{16} - 6905934q^{17} + 2727432q^{18} + 10661420q^{19} + \cdots \end{aligned}$$

be the Δ-function.

Conjecture 2.5.4 (Edixhoven). *There is an algorithm to compute $\tau(p)$, for prime p, that is polynomial-time in $\log(p)$. More generally, suppose $f = \sum a_n q^n$ is an eigenform in some space $M_k(N, \varepsilon)$, where $k \geq 2$. Then there is an algorithm to compute a_p, for p prime, in time polynomial in $\log(p)$.*

Bas Edixhoven, Jean-Marc Couveignes and Robin de Jong have mostly proved that $\tau(p)$ can be computed in polynomial time; their approach involves sophisticated techniques from arithmetic geometry (e.g., étale cohomology, motives, Arakelov theory). This is work in progress and has not been written up completely yet. The ideas Edixhoven uses are inspired by the ones introduced by Schoof, Elkies and Atkin for quickly counting points on elliptic curves over finite fields (see [Sch95]).

More precisely, Edixhoven describes his strategy as follows:

1. We compute the mod ℓ Galois representations associated to Δ. In particular, we produce a polynomial f such that $\mathbb{Q}[x]/(f)$ is the corresponding field. This is then used to obtain $\tau(p) \pmod{\ell}$ and do a Schoof like algorithm for computing $\tau(p)$.

2. We compute the field of definition of suitable points of order ℓ on $J_1(\ell)$ to do part 1.

3. The method is to approximate the polynomial f in some sense (e.g., over the complex numbers, or modulo many small primes r), and use an estimate from Arakelov theory to determine a precision that will suffice.

The rest of this book is about methods for computing subspaces of $M_k(\Gamma_1(N))$ for general N and k. These general methods are much more complicated than the methods presented in this chapter, since there are many more forms of small weight, and it can be difficult to obtain them. These forms of higher level have subtle and deep connections with arithmetic geometric objects such as elliptic curves, abelian varieties, and motives.

2.6 Exercises

2.1 By hand use (2.1.2) write down the coefficients of 1, q, q^2, and q^3 of the Eisenstein series E_8.

2.2 Explicitly compute the Victor Miller basis for $M_{28}(\mathrm{SL}_2(\mathbb{Z}))$ to precision $O(q^8)$. Your answer will look like Example 2.3.4.

2.3 Consider the cuspform $f = q^2 + 192q^3 - 8280q^4 \cdots$ in $S_{28}(\mathrm{SL}_2(\mathbb{Z}))$. Write f as a polynomial in E_4 and E_6 (see Remark 2.3.5).

Chapter 3

Modular Symbols of Weight Two

We saw in Chapter 2 (especially Section 2.2) that we can compute each space $M_k(\mathrm{SL}_2(\mathbb{Z}))$ explicitly. This involved computing Eisenstein series E_4 and E_6 to some precision, then forming the basis $\{E_4^a E_6^b : 4a + 6b = k, 0 \leq a, b \in \mathbb{Z}\}$ for $M_k(\mathrm{SL}_2(\mathbb{Z}))$. In this chapter we instead consider the problem of computing $M_2(\Gamma_0(N))$, for positive integers N. Again we have a decomposition

$$M_2(\Gamma_0(N)) = S_2(\Gamma_0(N)) \oplus \mathrm{Eis}_2(\Gamma_0(N)),$$

where $\mathrm{Eis}_2(\Gamma_0(N))$ is a space spanned by explicit generalized Eisenstein series and $S_2(\Gamma_0(N))$ is the space of cusp forms, i.e., elements of $M_2(\Gamma_0(N))$ that vanish at all cusps.

The space $\mathrm{Eis}_2(\Gamma_0(N))$ can be computed explicitly much like $M_k(\mathrm{SL}_2(\mathbb{Z}))$, as we will see in Chapter 5. On the other hand, general elements of $S_2(\Gamma_0(p))$ can not be written as sums or products of generalized Eisenstein series. In fact, the structure of $M_2(\Gamma_0(N))$ is drastically different than that of $M_k(\mathrm{SL}_2(\mathbb{Z}))$. For example, when p is a prime $\mathrm{Eis}_2(\Gamma_0(p))$ has dimension 1, whereas $S_2(\Gamma_0(p))$ has dimension about $p/12$.

Fortunately an idea of Birch called "modular symbols" provides a powerful method for computing $S_2(\Gamma_0(N))$, and indeed much more. In this chapter, we explain how $S_2(\Gamma_0(N))$ is related to modular symbols, and how to use this relationship to explicitly compute a basis for $S_2(\Gamma_0(N))$. We will discuss much more general modular symbols in Chapter 8, where we will explain how to use them to compute $S_k(\Gamma_1(N))$ for any integers $k \geq 2$ and N.

Section 3.1 contains a brief summary of basic facts about modular forms, Hecke operators, and integral homology. Section 3.2 introduces modular symbols, and describes how to compute with them. Section 3.5 outlines an algorithm for constructing cusp forms using modular symbols in conjunction with Atkin-Lehner theory.

This chapter assumes some familiarity with algebraic curves, Riemann surfaces, and homology groups of compact Riemann surfaces.

3.1 Review of modular forms and Hecke operators

The group $\Gamma_0(N)$ acts on $\mathfrak{h}^*$ by linear fractional transformations, and the quotient $\Gamma_0(N)\backslash\mathfrak{h}^*$ is a Riemann surface, which we denote by $X_0(N)$. Shimura showed in [Shi94, §6.7] that $X_0(N)$ has a canonical structure of algebraic curve over $\mathbb{Q}$.

Recall from Section 1.3 that a cusp form of weight 2 for $\Gamma_0(N)$ is a function f on $\mathfrak{h}$ such that $f(z)dz$ defines a holomorphic differential on $X_0(N)$. Equivalently, a cusp form is a holomorphic function f on $\mathfrak{h}$ such that

(a) the expression $f(z)dz$ is invariant under replacing z by $\gamma(z)$ for each $\gamma \in \Gamma_0(N)$, and

(b) $f(z)$ vanishes at every cusp for $\Gamma_0(N)$.

The space $S_2(\Gamma_0(N))$ of weight 2 cusp forms on $\Gamma_0(N)$ is a finite dimensional complex vector space, of dimension equal to the genus g of $X_0(N)$. Viewed topologically, as a 2-dimensional real manifold, $X_0(N)(\mathbb{C})$ is a g-holed torus (see Figure 3.1.1 on page 32).

Condition (b) in the definition of $f(z)$ means that $f(z)$ has a Fourier expansion about each element of $\mathbb{P}^1(\mathbb{Q})$. Thus, at ∞ we have

$$\begin{aligned} f(z) &= a_1 e^{2\pi i z} + a_2 e^{2\pi i 2z} + a_3 e^{2\pi i 3z} + \cdots \\ &= a_1 q + a_2 q^2 + a_3 q^3 + \cdots, \end{aligned}$$

where, for brevity, we write $q = q(z) = e^{2\pi i z}$.

Example 3.1.1. Let E be the elliptic curve defined by the equation $y^2 + xy = x^3 + x^2 - 4x - 5$. Let $a_p = p + 1 - \#\tilde{E}(\mathbb{F}_p)$, where $\tilde{E}$ is the reduction of E mod p (note that for the bad primes we have $a_3 = -1$, $a_{13} = 1$). For n composite, define a_n using the relations at the end of Section 3.5. Then

$$\begin{aligned} f &= q + a_2 q^2 + a_3 q^3 + a_4 q^4 + a_5 q^5 + \cdots \\ &= q + q^2 - q^3 - q^4 + 2q^5 + \cdots \end{aligned}$$

is the q-expansion of a modular form on $\Gamma_0(39)$. The Shimura-Taniyama conjecture, which is now a theorem (see [BCDT01]) asserts that any q-expansion constructed as above from an elliptic curve over $\mathbb{Q}$ is a modular form.

Just as is the case for level 1 modular forms (see Section 2.4) there is a family of commuting Hecke operators that act on $S_2(\Gamma_0(N))$. To define them conceptually, we introduce an interpretation of $X_0(N)$ as a space whose points *parameterize* elliptic curves with extra structure.

Proposition 3.1.2. *The complex points of the open subcurve $Y_0(N) = \Gamma_0(N)\backslash\mathfrak{h}$ are in natural bijection with pairs (E, C), where E is an elliptic curve over $\mathbb{C}$ and C is a cyclic subgroup of $E(\mathbb{C})$ of order N.*

Suppose n and N are coprime positive integers. Keeping in mind Proposition 3.1.2, we see that there are two natural maps π_1 and π_2 from $Y_0(n \cdot N)$ to $Y_0(N)$; the first, π_1, sends a pair (E, C) to (E, C'), where C' is the unique cyclic subgroup of C of order N, and the second, π_2, sends a point $(E, C) \in Y_0(N)(\mathbb{C})$ to $(E/D, C/D)$, where D is the unique cyclic subgroup of C of order n. These maps extend in a unique way to algebraic maps from $X_0(n \cdot N)$ to $X_0(N)$:

$$\begin{tikzcd} & X_0(n\cdot N) \arrow[dl, "\pi_2"'] \arrow[dr, "\pi_1"] & \\ X_0(N) & & X_0(N). \end{tikzcd}$$

The nth *Hecke operator* T_n is $(\pi_1)_* \circ (\pi_2)^*$, where π_2^* and $(\pi_1)_*$ denote pullback and pushforward of differentials respectively. (There is a similar definition of T_n when $\gcd(n, N) \neq 1$.) Using our interpretation of $S_2(\Gamma_0(N))$ as differentials on $X_0(N)$ this gives an action of Hecke operators on $S_2(\Gamma_0(N))$. One can show that these induce the maps of Proposition 2.4.6 on q-expansions.

Example 3.1.3. There is a basis of $S_2(39)$ so that

$$T_2 = \begin{pmatrix} 1 & 1 & 0 \\ -2 & -3 & -2 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad T_5 = \begin{pmatrix} -4 & -2 & -6 \\ 4 & 4 & 4 \\ 0 & 0 & 2 \end{pmatrix}.$$

Notice that these matrices commute, and that 1 is an eigenvalue of T_2, and 2 is an eigenvalue of T_5.

The first homology group $\mathrm{H}_1(X_0(N), \mathbb{Z})$ is the group of closed 1-cycles modulo boundaries of 2 cycles (formal sums of images of 2-simplexes). Recall that topologically $X_0(N)$ is a g-holed torus, where g is the genus of $X_0(N)$. The group $\mathrm{H}_1(X_0(N), \mathbb{Z})$ is thus a free abelian group of rank $2g$ (see, e.g., [GH81, Ex. 19.30]), with two generators corresponding to each hole, as illustrated in the case $N = 39$ in Figure 3.1.1.

Homology is closely connected to modular forms, since the Hecke operators T_n also act on $\mathrm{H}_1(X_0(N), \mathbb{Z})$. The action is by pullback of homology classes by π_2 followed by taking the image under π_1. Moreover, integration defines a pairing

$$\langle\, ,\, \rangle : S_2(\Gamma_0(N)) \times H_1(X_0(N), \mathbb{Z}) \to \mathbb{C}. \tag{3.1.1}$$

Explicitly, for a path x,

$$\langle f, x\rangle = 2\pi i \int_x f(z)dz,$$

where the integral is locally a complex line integral along preimages of intervals of x in the upper half plane.

$$\mathrm{H}_1(X_0(39), \mathbb{Z}) \cong \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$$

Figure 3.1.1: The homology of $X_0(39)$.

Theorem 3.1.4. *The pairing (3.1.1) is nondegenerate and Hecke equivariant in the sense that for every Hecke operator T_n, we have $\langle fT_n, x\rangle = \langle f, T_n x\rangle$.*

As we will see, modular symbols allow us to make explicit the action of the Hecke operators on $\mathrm{H}_1(X_0(N), \mathbb{Z})$; the above pairing then translates this into a wealth of information about cusp forms.

3.2 Modular symbols

The modular symbols formalism provides a presentation of $\mathrm{H}_1(X_0(N), \mathbb{Z})$ in terms of paths between elements of $\mathbb{P}^1(\mathbb{Q})$. Furthermore, a trick due to Manin gives an explicit finite list of generators and relations for the space of modular symbols.

The *modular symbol* defined by a pair $\alpha, \beta \in \mathbb{P}^1(\mathbb{Q})$ is denoted $\{\alpha, \beta\}$. As illustrated in Figure 3.2.1, we view this modular symbol as the homology class, relative to the cusps, of a (geodesic) path from α to β in $\mathfrak{h}^*$. The homology group relative to the cusps is a slight enlargement of the usual homology group, in that we allow paths with endpoints in the cusps instead of restricting to closed loops.

Note that modular symbols satisfy the following homology relations: if $\alpha, \beta, \gamma \in \mathbb{Q} \cup \{\infty\}$, then

$$\{\alpha, \beta\} + \{\beta, \gamma\} + \{\gamma, \alpha\} = 0.$$

Furthermore, the space of modular symbols is torsion free, so, e.g., $\{\alpha, \alpha\} = 0$ and $\{\alpha, \beta\} = -\{\beta, \alpha\}$.

Denote by $\mathbb{M}_2$ the free abelian group with basis the set of symbols $\{\alpha, \beta\}$ modulo the three-term homology relations above and modulo any torsion. There is a left action of $\mathrm{GL}_2(\mathbb{Q})$ on $\mathbb{M}_2$, whereby a matrix g acts by

$$g\{\alpha, \beta\} = \{g(\alpha), g(\beta)\},$$

and g acts on α and β by a linear fractional transformation. The space $\mathbb{M}_2(\Gamma_0(N))$ of *modular symbols for* $\Gamma_0(N)$ is the quotient of $\mathbb{M}_2$ by the submodule generated by the infinitely many elements of the form $x - g(x)$, for x in $\mathbb{M}_2$ and g in

Figure 3.2.1: The modular symbols $\{\alpha, \beta\}$ and $\{0, \infty\}$.

$\Gamma_0(N)$, and modulo any torsion. A *modular symbol for* $\Gamma_0(N)$ is an element of this space. We frequently denote the equivalence class that defines a modular symbol by giving a representative element.

Example 3.2.1. Some modular symbols are 0 no matter what the level N is! For example, since $\gamma = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma_0(N)$, we have

$$\{\infty, 0\} = \{\gamma(\infty), \gamma(0)\} = \{\infty, 1\},$$

so

$$0 = \{\infty, 1\} - \{\infty, 0\} = \{\infty, 1\} + \{0, \infty\} = \{0, \infty\} + \{\infty, 1\} = \{0, 1\}.$$

There is a natural homomorphism

$$\varphi : \mathbb{M}_2(\Gamma_0(N)) \to \mathrm{H}_1(X_0(N), \{\text{cusps}\}, \mathbb{Z}) \tag{3.2.1}$$

that sends a formal linear combination of geodesic paths in the upper half plane to their image as paths on $X_0(N)$. In [Man72] Manin proved that (3.2.1) is an isomorphism. He also identified the subspace of $\mathbb{M}_2(\Gamma_0(N))$ that is sent isomorphically onto $\mathrm{H}_1(X_0(N), \mathbb{Z})$. This subspace is constructed as follows. Let $\mathbb{B}_2(\Gamma_0(N))$ denote the free abelian group whose basis is the finite set $C(\Gamma_0(N)) = \Gamma_0(N)\backslash\mathbb{P}^1(\mathbb{Q})$ of cusps for $\Gamma_0(N)$. The *boundary map*

$$\delta : \mathbb{M}_2(\Gamma_0(N)) \to \mathbb{B}_2(\Gamma_0(N))$$

sends $\{\alpha, \beta\}$ to $\{\beta\} - \{\alpha\}$, where $\{\beta\}$ denotes the basis element of $\mathbb{B}_2(\Gamma_0(N))$ corresponding to $\beta \in \mathbb{P}^1(\mathbb{Q})$. The kernel $\mathbb{S}_2(\Gamma_0(N))$ of δ is the subspace of *cuspidal* modular symbols. An element of $\mathbb{S}_2(\Gamma_0(N))$ can be thought of as a linear combination of paths in $\mathfrak{h}^*$ whose endpoints are cusps, and whose images in $X_0(N)$ are a linear combination of loops.

Theorem 3.2.2 (Manin). *The map φ given above induces a canonical isomorphism*

$$\mathbb{S}_2(\Gamma_0(N)) \cong \mathrm{H}_1(X_0(N), \mathbb{Z}).$$

Example 3.2.3. We illustrate modular symbols in the case when $N = 11$. Using SAGE we find that $\mathbb{M}_2(11)$ has basis $\{\infty, 0\}$, $\{-1/8, 0\}$, $\{-1/9, 0\}$:

```
sage: M = ModularSymbols(11, 2)
sage: print [b.modular_symbol_rep() for b in M.basis()]
[{Infinity,0}, {-1/8,0}, {-1/9,0}]
```

The integral homology $\mathrm{H}_1(X_0(11), \mathbb{Z})$ corresponds to the abelian subgroup generated by $\{-1/7, 0\}$ and $\{-1/5, 0\}$.

3.2.1 Manin's trick

In this section, we describe a trick of Manin that shows that the space of modular symbols can be computed.

The group $\Gamma_0(N)$ has finite index in $\mathrm{SL}_2(\mathbb{Z})$ (see Exercise 1.6). Let $r_0, r_1, \ldots, r_m$ be distinct right coset representatives for $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$, so that

$$\mathrm{SL}_2(\mathbb{Z}) = \Gamma_0(N)r_0 \cup \Gamma_0(N)r_1 \cup \cdots \cup \Gamma_0(N)r_m,$$

where the union is disjoint. For example, when N is prime, a list of coset representatives is

$$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}, \begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}, \begin{pmatrix}1 & 0\\2 & 1\end{pmatrix}, \begin{pmatrix}1 & 0\\3 & 1\end{pmatrix}, \ldots, \begin{pmatrix}1 & 0\\N-1 & 1\end{pmatrix}, \begin{pmatrix}0 & -1\\1 & 0\end{pmatrix}.$$

Let

$$\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z}) = \{(a:b) \,:\, a, b \in \mathbb{Z}/N\mathbb{Z},\ \gcd(a,b,N) = 1\}/\sim$$

where $(a:b) \sim (a':b')$ if there is $u \in (\mathbb{Z}/N\mathbb{Z})^*$ such that $a = ua', b = ub'$.

Proposition 3.2.4. *There is a bijection between $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ and the right cosets of $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$, which sends a coset representative $\left(\begin{smallmatrix}a & b\\c & d\end{smallmatrix}\right)$ to the class of $(c:d)$ in $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.*

Proof. See Exercise 3.1. □

We now describe an observation of Manin (see [Man72, §1.5]) that is crucial to making $\mathbb{M}_2(\Gamma_0(N))$ computable. It allows us to write any modular symbol $\{\alpha, \beta\}$ as a $\mathbb{Z}$-linear combination of symbols of the form $r_i\{0, \infty\}$, where the $r_i \in \mathrm{SL}_2(\mathbb{Z})$ are coset representatives as above. In particular, the finitely many symbols $r_0\{0, \infty\}, \ldots r_m\{0, \infty\}$ generate $\mathbb{M}_2(\Gamma_0(N))$.

Proposition 3.2.5 (Manin). *Let N be a positive integer and $r_0, \ldots, r_m$ a set of right coset representatives for $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$. Every $\{\alpha, \beta\} \in \mathbb{M}_2(\Gamma_0(N))$ is a $\mathbb{Z}$-linear combination of $r_0\{0,\infty\}, \ldots r_m\{0,\infty\}$.*

We give two proofs of the proposition. The first is useful for actual computation (see [Cre97a, §2.1.6]); the second seems less useful for computation but is easy to understand conceptually (see [MTT86, §2]).

Continued Fractions Proof of Proposition 3.2.5. Because of the relation $\{\alpha, \beta\} = \{0, \beta\} - \{0, \alpha\}$, it suffices to consider modular symbols of the form $\{0, b/a\}$, where the rational number b/a is in lowest terms. Expand b/a as a continued fraction and consider the successive convergents in lowest terms:

$$\frac{b_{-2}}{a_{-2}} = \frac{0}{1}, \quad \frac{b_{-1}}{a_{-1}} = \frac{1}{0}, \quad \frac{b_0}{a_0} = \frac{b_0}{1}, \ldots, \quad \frac{b_{n-1}}{a_{n-1}}, \quad \frac{b_n}{a_n} = \frac{b}{a}$$

where the first two are added formally. Then

$$b_k a_{k-1} - b_{k-1} a_k = (-1)^{k-1},$$

so that

$$g_k = \begin{pmatrix} b_k & (-1)^{k-1} b_{k-1} \\ a_k & (-1)^{k-1} a_{k-1} \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}).$$

Hence

$$\left\{ \frac{b_{k-1}}{a_{k-1}}, \frac{b_k}{a_k} \right\} = g_k\{0, \infty\} = r_i\{0, \infty\},$$

for some i, is of the required special form. Since

$$\{0, b/a\} = \{0, \infty\} + \{\infty, b_0\} + \left\{ \frac{b_0}{1}, \frac{b_1}{a_1} \right\} + \cdots + \left\{ \frac{b_{n-1}}{a_{n-1}}, \frac{b_n}{a_n} \right\},$$

this completes the proof. □

Inductive Proof of Proposition 3.2.5. As in the first proof it suffices to prove the proposition for any symbol $\{0, b/a\}$, where b/a is in lowest terms. We will induct on $a \in \mathbb{Z}_{\geq 0}$. If $a = 0$ then the symbol is $\{0, \infty\}$, which corresponds to the identity coset, so assume that $a > 0$. Find $a' \in \mathbb{Z}$ such that

$$ba' \equiv 1 \pmod{a},$$

and set $b' = (ba' - 1)/a$. Then the matrix

$$\delta = \begin{pmatrix} b & b' \\ a & a' \end{pmatrix}$$

is an element of $\mathrm{SL}_2(\mathbb{Z})$, so $\delta = \gamma \cdot r_j$ for some right coset representative r_j and $\gamma \in \Gamma_0(N)$. Then

$$\{0, b/a\} - \{0, b'/a'\} = \{b'/a', b/a\} = \begin{pmatrix} b & b' \\ a & a' \end{pmatrix} \cdot \{0, \infty\} = r_j\{0, \infty\}.$$

By induction $\{0, b'/a'\}$ is a linear combination of symbols of the form $r_k\{0, \infty\}$, which completes the proof. □

Example 3.2.6. Let $N = 11$, and consider the modular symbol $\{0, 4/7\}$. We have

$$\frac{4}{7} = 0 + \frac{1}{1 + \frac{1}{1+\frac{1}{3}}},$$

so the partial convergents are

$$\frac{b_{-2}}{a_{-2}} = \frac{0}{1}, \quad \frac{b_{-1}}{a_{-1}} = \frac{1}{0}, \quad \frac{b_0}{a_0} = \frac{0}{1}, \quad \frac{b_1}{a_1} = \frac{1}{1}, \quad \frac{b_2}{a_2} = \frac{1}{2}, \quad \frac{b_3}{a_3} = \frac{4}{7}.$$

Thus, noting as in Example 3.2.1 that $\{0, 1\} = 0$, we have

$$\begin{aligned}
\{0, 4/7\} &= \{0, \infty\} + \{\infty, 0\} + \{0, 1\} + \{1, 1/2\} + \{1/2, 4/7\} \\
&= \begin{pmatrix} 1 & -1 \\ 2 & -1 \end{pmatrix} \{0, \infty\} + \begin{pmatrix} 4 & 1 \\ 7 & 2 \end{pmatrix} \{0, \infty\} \\
&= \begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0, \infty\} + \begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0, \infty\} \\
&= 2 \cdot \left[\begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0, \infty\} \right]
\end{aligned}$$

3.2.2 Manin symbols

As above, fix coset representatives $r_0, \ldots, r_m$ for $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$. Consider formal symbols $[r_i]'$ for $i = 0, \ldots, m$. Let $[r_i]$ be the modular symbol $r_i\{0, \infty\} = \{r_i(0), r_i(\infty)\}$. We equip the symbols $[r_0]', \ldots, [r_m]'$ with a right action of $\mathrm{SL}_2(\mathbb{Z})$, which is given by $[r_i]'.g = [r_j]'$, where $\Gamma_0(N)r_j = \Gamma_0(N)r_i g$. We extend the notation by writing $[\gamma]' = [\Gamma_0(N)\gamma]' = [r_i]'$, where $\gamma \in \Gamma_0(N)r_i$; then the action is simply $[\gamma]'.g = [\gamma g]'$.

Theorem 1.1.2 implies that $\mathrm{SL}_2(\mathbb{Z})$ is generated by the two matrices $\sigma = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\tau = \left(\begin{smallmatrix} 1 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. Note that $\sigma = S$ from that theorem and $\tau = TS$, so $T = \tau\sigma \in \langle \sigma, \tau \rangle$.

The following theorem provides us with a finite presentation for the space of modular symbols.

Theorem 3.2.7 (Manin). *Consider the quotient M of the free abelian group on Manin symbols $[r_0]', \ldots, [r_m]'$ modulo the subgroup generated by the elements (for all i):*

$$[r_i]' + [r_i]'\sigma \qquad \textit{and} \qquad [r_i]' + [r_i]'\tau + [r_i]'\tau^2,$$

and modulo any torsion. Then there is an isomorphism $\Psi : M \xrightarrow{\sim} \mathbb{M}_2(\Gamma_0(N))$ given by $[r_i]' \mapsto [r_i] = r_i\{0, \infty\}$.

Proof. Proposition 3.2.5 implies that Ψ is surjective, assuming that Ψ is well defined.

We next verify that Ψ is well defined, i.e. that the listed two and three term relations hold in the image. To see that the the first relation holds, note that

$$\begin{aligned}[r_i] + [r_i]\sigma &= \{r_i(0), r_i(\infty)\} + \{r_i\sigma(0), r_i\sigma(\infty)\} \\ &= \{r_i(0), r_i(\infty)\} + \{r_i(\infty), r_i(0)\} \\ &= 0.\end{aligned}$$

For the second relation we have

$$\begin{aligned}[r_i] + [r_i]\tau + [r_i]\tau^2 &= \{r_i(0), r_i(\infty)\} + \{r_i\tau(0), r_i\tau(\infty)\} + \{r_i\tau^2(0), r_i\tau^2(\infty)\} \\ &= \{r_i(0), r_i(\infty)\} + \{r_i(\infty), r_i(1)\} + \{r_i(1), r_i(0)\} \\ &= 0\end{aligned}$$

The proof that Ψ is injective requires more work; see [Man72, §1.7]. $\square$

Example 3.2.8. By default SAGE computes modular symbols spaces over $\mathbb{Q}$, i.e., $\mathbb{M}_2(\Gamma_0(N); \mathbb{Q}) \cong \mathbb{M}_2(\Gamma_0(N)) \otimes \mathbb{Q}$. SAGE represents (weight 2) Manin symbols as pairs (c, d). Here c, d are integers that satisfy $0 \le c, d < N$; they define a point $(c : d) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, hence a right coset of $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$ (see Proposition 3.2.4).

Create $\mathbb{M}_2(\Gamma_0(N); \mathbb{Q})$ in SAGE by typing `ModularSymbols(N, 2)`. We then use the SAGE command `manin_generators` to enumerate a list of generators $[r_0], \ldots, [r_n]$ as in Theorem 3.2.7 for several spaces of modular symbols.

```
sage: M = ModularSymbols(2,2)
sage: M
Full Modular Symbols space for Gamma_0(2) of weight 2 with
sign 0 and dimension 1 over Rational Field
sage: M.manin_generators()
[(0,1), (1,0), (1,1)]

sage: M = ModularSymbols(3,2)
sage: M.manin_generators()
[(0,1), (1,0), (1,1), (1,2)]

sage: M = ModularSymbols(6,2)
sage: M.manin_generators()
[(0,1), (1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (2,1),
 (2,3), (2,5), (3,1), (3,2)]
```

Given `x=(c,d)`, the command `x.lift_to_sl2z(N)` finds an element `[a,b,c',d']` of $\mathrm{SL}_2(\mathbb{Z})$ whose lower two entries are congruent to (c, d) modulo N.

```
sage: M = ModularSymbols(2,2)
sage: [x.lift_to_sl2z(2) for x in M.manin_generators()]
[[1, 0, 0, 1], [0, -1, 1, 0], [0, -1, 1, 1]]
sage: M = ModularSymbols(6,2)
sage: x = M.manin_generators()[9]
sage: x
(2,5)
sage: x.lift_to_sl2z(6)
[1, 2, 2, 5]
```

The `manin_basis` command returns a list of indices into the Manin generator list such that the corresponding symbols form a basis for the quotient of the $\mathbb{Q}$-vector space spanned by Manin symbols modulo the 2 and 3-term relations of Theorem 3.2.7.

```
sage: M = ModularSymbols(2,2)
sage: M.manin_basis()
[1]
sage: [M.manin_generators()[i] for i in M.manin_basis()]
[(1,0)]
sage: M = ModularSymbols(6,2)
sage: M.manin_basis()
[1, 10, 11]
sage: [M.manin_generators()[i] for i in M.manin_basis()]
[(1,0), (3,1), (3,2)]
```

Thus, e.g., every element of $\mathbb{M}_2(\Gamma_0(6))$ is a $\mathbb{Q}$-linear combination of the symbols $[(1,0)]$, $[(3,1)]$, and $[(3,2)]$. We can write each of these as a modular symbol using the `modular_symbol_rep` function.

```
sage: M.basis()
((1,0), (3,1), (3,2))
sage: [x.modular_symbol_rep() for x in M.basis()]
[{Infinity,0}, {0,1/3}, {-1/2,-1/3}]
```

The `manin_gens_to_basis` function returns a matrix whose rows express each Manin symbol generator in terms of the subset of Manin symbols that forms a basis (as returned by `manin_basis`.

```
sage: M = ModularSymbols(2,2)
sage: M.manin_gens_to_basis()
[-1]
[ 1]
[ 0]
```

Since the basis is $(1,0)$ this means that in $\mathbb{M}_2(\Gamma_0(2);\mathbb{Q})$, we have $[(0,1)] = -[(1,0)]$ and $[(1,1)] = 0$. (Since no denominators are involved, we have in fact computed a presentation of $\mathbb{M}_2(\Gamma_0(2);\mathbb{Z})$.)

Convert a Manin symbol $x = (c,d)$ to an element of a modular symbols space M, use `M(xx)`:

```
sage: M = ModularSymbols(2,2)
sage: x = (1,0); M(x)
(1,0)
sage: M( (3,1) )    # entries are reduced modulo $2$ first
0
sage: M( (10,19) )
-(1,0)
```

Next consider $\mathbb{M}_2(\Gamma_0(6);\mathbb{Q})$:

```
sage: M = ModularSymbols(6,2)
sage: M.manin_gens_to_basis()
[-1  0  0]
[ 1  0  0]
[ 0  0  0]
[ 0 -1  1]
[ 0 -1  0]
[ 0 -1  1]
[ 0  0  0]
[ 0  1 -1]
[ 0  0 -1]
[ 0  1 -1]
[ 0  1  0]
[ 0  0  1]
```

Recalling that our choice of basis for $\mathbb{M}_2(\Gamma_0(6);\mathbb{Q})$ is $[(1,0)], [(3,1)], [(3,2)]$. Thus, e.g., first row of this matrix says that $[(0,1)] = -[(1,0)]$, and the fourth row asserts that $[(1,2)] = -[(3,1)] + [(3,2)]$.

```
sage: M = ModularSymbols(6,2)
sage: M((0,1))
-(1,0)
sage: M((1,2))
-(3,1) + (3,2)
```

3.2.3 Hecke operators on modular symbols

When p is a prime not dividing N, define

$$T_p\{\alpha,\beta\} = \begin{pmatrix} p & 0 \\ 0 & 1 \end{pmatrix}\{\alpha,\beta\} + \sum_{r \bmod p} \begin{pmatrix} 1 & r \\ 0 & p \end{pmatrix}\{\alpha,\beta\}.$$

As mentioned before, this definition is compatible with the integration pairing $\langle\, ,\, \rangle$ of Section 3.1, in the sense that $\langle fT_p, x\rangle = \langle f, T_p x\rangle$. When $p \mid N$, the definition is the same, except that the matrix $\left(\begin{smallmatrix} p & 0 \\ 0 & 1 \end{smallmatrix}\right)$ is not included in the sum. (There is a similar definition of T_n for n composite; see Section 8.3.1 for the general definition.)

Example 3.2.9. For example, when $N = 11$ we have

$$\begin{aligned} T_2\{0,1/5\} &= \{0,2/5\} + \{0,1/10\} + \{1/2,3/5\} \\ &= -2\{0,1/5\}. \end{aligned}$$

In [Mer94], L. Merel gives a description of the action of T_p directly on Manin symbols $[r_i]$ (see Section 8.3.2 for details). For example, when $p = 2$ and N is odd, we have

$$T_2([r_i]) = [r_i]\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} + [r_i]\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} + [r_i]\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} + [r_i]\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}. \qquad (3.2.2)$$

The SAGE command `HeilbronnMerelList(n)` gives a list of matrices $[a,b,c,d]$ that compute T_n on Manin symbols. The command `HeilbronnCremonaList(p)`, for p prime, gives a list of matrices that computes T_p on Manin symbols for $p \nmid N$.

```
sage: HeilbronnMerelList(2)
[[1, 0, 0, 2], [1, 0, 1, 2], [2, 0, 0, 1], [2, 1, 0, 1]]
sage: HeilbronnMerelList(4)
[[1, 0, 0, 4], [1, 0, 1, 4], [1, 0, 2, 4], [1, 0, 3, 4],
 [2, 0, 0, 2], [2, 1, 0, 2], [2, 0, 1, 2], [2, 1, 2, 3],
 [3, 2, 1, 2], [4, 0, 0, 1], [4, 1, 0, 1], [4, 2, 0, 1],
 [4, 3, 0, 1]]
sage: HeilbronnMerelList(5)
[[1, 0, 0, 5], [1, 0, 1, 5], [1, 0, 2, 5], [1, 0, 3, 5],
 [1, 0, 4, 5], [2, 1, 1, 3], [2, 1, 3, 4], [3, 1, 1, 2],
 [3, 2, 2, 3], [4, 3, 1, 2], [5, 0, 0, 1], [5, 1, 0, 1],
 [5, 2, 0, 1], [5, 3, 0, 1], [5, 4, 0, 1]]
```

Notice that Cremona's list is shorter than Merel's (see [Cre97a] for a derivation that Cremona's list can be used to compute Hecke operators).

```
sage: HeilbronnCremonaList(5)
[[1, 0, 0, 5], [5, 2, 0, 1], [2, 1, 1, 3], [1, 0, 3, 5],
 [5, 1, 0, 1], [1, 0, 1, 5], [5, 0, 0, 1], [5, -1, 0, 1],
 [-1, 0, 1, -5], [5, -2, 0, 1], [-2, 1, 1, -3], [1, 0, -3, 5]]
sage: len(HeilbronnCremonaList(97))
392
sage: len(HeilbronnMerelList(97))
1039
```

Example 3.2.10. Using SAGE we compute the matrix of T_2 on $\mathbb{M}_2(\Gamma_0(2))$:

```
sage: M = ModularSymbols(2,2)
sage: M.T(2).matrix()
[1]
```

We can do this more explicitly as follows, recalling that $(1, 0)$ is a basis for $\mathbb{M}_2(\Gamma_0(2))$ from Example 3.2.8 and using (3.2.2):

```
sage: M = ModularSymbols(2,2)
sage: M.basis()
((1,0),)
sage: M((1,0)) + M((2,1)) + M((1,0))
(1,0)
```

Note that we do not include (2,0) since 2 divides the level.

Example 3.2.11. We use SAGE to compute Hecke operators on $\mathbb{M}_2(\Gamma_0(6))$:

```
sage: M = ModularSymbols(6, 2)
sage: M.T(2).matrix()
[ 2  1 -1]
[-1  0  1]
[-1 -1  2]
sage: M.T(3).matrix()
[3 2 0]
[0 1 0]
[2 2 1]
sage: M.T(5).matrix()
[6 0 0]
[0 6 0]
[0 0 6]
sage: M.T(97).matrix()
[98  0  0]
[ 0 98  0]
[ 0  0 98]
```

In fact for $p \geq 5$ we have $T_p = p + 1$, since $M_2(\Gamma_0(6))$ is spanned by generalized

Eisenstein series (see Chapter 5).

Example 3.2.12. We use SAGE to compute Hecke operators on $\mathbb{M}_2(\Gamma_0(39))$:

```
sage: M = ModularSymbols(39, 2)
sage: T2 = M.T(2)
sage: T2.matrix()
[ 3  0 -1  0  0  1  1 -1  0]
[ 0  0  2  0 -1  1  0  1 -1]
[ 0  1  0 -1  1  1  0  1 -1]
[ 0  0  1  0  0  1  0  1 -1]
[ 0 -1  2  0  0  1  0  1 -1]
[ 0  0  1  1  0  1  1 -1  0]
[ 0  0  0 -1  0  1  1  2  0]
[ 0  0  0  1  0  0  2  0  1]
[ 0  0 -1  0  0  0  1  0  2]
sage: T2.charpoly()
x^9 - 7*x^8 + 4*x^7 + 68*x^6 - 142*x^5 - 78*x^4
                      + 460*x^3 - 468*x^2 + 189*x - 27
sage: factor(T2.charpoly())
(x - 3)^3 * (x - 1)^2 * (x^2 + 2*x - 1)^2
```

Notice that the Hecke operators commute, so their eigenspace structure is similar.

```
sage: T2 = M.T(2).matrix()
sage: T5 = M.T(5).matrix()
sage: T2*T5 - T5*T2 == 0
True
sage: T5.charpoly().factor()
(x - 6)^3 * (x - 2)^2 * (x^2 - 8)^2
```

The rational decomposition of T_2 is a list of the kernels of $(f^e)(T_2)$, where f runs through the irreducible factors of the characteristic polynomial of T_2 and f^e exactly divides this characteristic polynomial. Using SAGE we find them:

```
sage: T2.decomposition()
[(Vector space of degree 9 and dimension 3 over Rational Field
Basis matrix:
[ 1  0  0  0  0  0 -1 -1 -1]
[ 0  1  4  0  1  0 -6 -2 -8]
[ 0  0  0  0  0  1  2  1  1],
  False),
 (Vector space of degree 9 and dimension 2 over Rational Field
Basis matrix:
[ 0  1  0  0 -1  0  0  0  0]
[ 0  0  0  0  0  1  0 -1  1],
  False),
 (Vector space of degree 9 and dimension 4 over Rational Field
Basis matrix:
[   0    1    0    0    1   -1    0   -1    1]
[   0    0    1    0    0 -1/2    0 -1/2  1/2]
[   0    0    0    1    0 -1/2    0 -1/2  1/2]
[   0    0    0    0    0    0    1   -1    0],
  False)]
```

The space of modular symbols also decomposes under all Hecke operators (of index coprime to 39) as follows:

```
sage: M.decomposition()
[Dimension 3 subspace of a modular symbols space of level 39,
 Dimension 2 subspace of a modular symbols space of level 39,
 Dimension 4 subspace of a modular symbols space of level 39]
```

3.3 Computing the boundary map

In Section 3.2 we defined a map $\mathbb{M}_2(\Gamma_0(N)) \to \mathbb{B}_2(\Gamma_0(N))$ whose kernel $\mathbb{S}_2(\Gamma_0(N))$ is called the space of cuspidal modular symbols. This kernel will be important in computing cuspforms in Section 3.5 below.

To compute the boundary map on Manin symbols, note that $[\gamma] = \{\gamma(0), \gamma(\infty)\}$, so if $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, then

$$\delta([\gamma]) = \{\gamma(\infty)\} - \{\gamma(0)\} = \{a/c\} - \{b/d\}.$$

Computing this boundary map would appear to first require an algorithm to compute the set $C(\Gamma_0(N)) = \Gamma_0(N)\backslash\mathbb{P}^1(\mathbb{Q})$ of cusps for $\Gamma_0(N)$. In fact, there is a trick to compute the set of cusps in the course of running the algorithm. First, give an algorithm for deciding whether or not two elements of $\mathbb{P}^1(\mathbb{Q})$ are equivalent modulo the action of $\Gamma_0(N)$. Then simply construct $C(\Gamma_0(N))$ in the course of computing the boundary map, i.e., keep a list of cusps found so far, and whenever a new cusp class is discovered add it to the list. The following proposition, which is proved in [Cre97a, Prop. 2.2.3], explains how to determine whether two cusps are equivalent.

Proposition 3.3.1 (Cremona). *Let (c_i, d_i), $i = 1, 2$ be pairs of integers with $\gcd(c_i, d_i) = 1$, and possibly $d_i = 0$. There exists $g \in \Gamma_0(N)$ such that $g(c_1/d_1) = c_2/d_2$ in $\mathbb{P}^1(\mathbb{Q})$ if and only if*

$$s_1 d_2 \equiv s_2 d_1 \pmod{\gcd(d_1 d_2, N)}$$

where s_j satisfies $c_j s_j \equiv 1 \pmod{d_j}$.

In SAGE the command `boundary_map()` computes the boundary map from $\mathbb{M}_2(\Gamma_0(N))$ to $\mathbb{B}_2(\Gamma_0(N))$, and the `cuspidal_submodule()` command computes its kernel. For example, for level 2 the boundary map is given by the matrix $[1 \;\; -1]$, and its kernel is the 0 space.

```
sage: M = ModularSymbols(2, 2)
sage: M.boundary_map()
Hecke module morphism boundary map defined by the matrix
[ 1 -1]
Domain: Full Modular Symbols space for Gamma_0(2) of weight 2 with sign ...
Codomain: Space of Boundary Modular Symbols for Gamma0(2) of weight 2 and ...
sage: M.cuspidal_submodule()
Dimension 0 subspace of a modular symbols space of level 2
```

The smallest level for which the boundary map has nontrivial kernel, i.e., for which $\mathbb{S}_2(\Gamma_0(N)) \neq 0$ is $N = 11$.

```
sage: M = ModularSymbols(11, 2)
sage: M.boundary_map().matrix()
[ 1 -1]
[ 0  0]
[ 0  0]
sage: M.cuspidal_submodule()
Dimension 2 subspace of a modular symbols space of level 11
sage: S = M.cuspidal_submodule(); S
Dimension 2 subspace of a modular symbols space of level 11
sage: S.basis()
((1,8), (1,9))
```

The following illustrates that the Hecke operators preserve $\mathbb{S}_2(\Gamma_0(N))$:

```
sage: S.T(2).matrix()
[-2  0]
[ 0 -2]
sage: S.T(3).matrix()
[-1  0]
[ 0 -1]
sage: S.T(5).matrix()
[1 0]
[0 1]
```

A nontrivial fact (the Eichler-Shimura relation, etc.) is that for p prime the eigenvalue of each of these matrices is the same as $p + 1 - \#E(\mathbb{F}_p)$, where E is the elliptic curve $X_0(11)$ given by the equation

$$y^2 + y = x^3 - x^2 - 10x - 20.$$

```
sage: E = EllipticCurve([0,-1,1,-10,-20])
sage: 2 + 1 - E.Np(2)
-2
sage: 3 + 1 - E.Np(3)
-1
sage: 5 + 1 - E.Np(5)
1
sage: print [S.T(p).matrix()[0,0] - (p+1-E.Np(p)) for p in primes(100)]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

3.4 Computing a basis for $S_2(\Gamma_0(N))$

In this section we explain a method for using what we know how to compute using modular symbols to compute a basis for $S_2(\Gamma_0(N))$.

Let $\mathbb{M}_2(\Gamma_0(N); \mathbb{Q})$ and $\mathbb{S}_2(\Gamma_0(N); \mathbb{Q})$ denote modular symbols and cuspidal modular symbols over $\mathbb{Q}$. Before we begin, we describe a simple but crucial fact about the relation between cusp forms and the Hecke algebra.

If $f = \sum b_n q^n \in \mathbb{C}[[q]]$ is a power series, let $a_n(f) = b_n$ be the n coefficient of f. Notice that a_n is a linear map from $\mathbb{C}[[q]]$ to itself.

As explained in [Lan95, §VII.3], the Hecke operators T_n acts on elements of $M_2(\Gamma_0(N))$ as follows:

$$T_n\left(\sum_{m=0}^{\infty} a_m q^m\right) = \left(\sum_{1 \leq d \,|\, \gcd(n,m)} \varepsilon(d) \cdot d \cdot a_{mn/d^2}\right) q^m, \qquad (3.4.1)$$

where $\varepsilon(d) = 1$ if $\gcd(d, N) = 1$ and $\varepsilon(d) = 0$ if $\gcd(d, N) \neq 1$.

Lemma 3.4.1. *Suppose f is a modular form and n is a positive integer. Then*

$$a_1(T_n(f)) = a_n(f).$$

Proof. The coefficient of q in (3.4.1) is $\varepsilon(1) \cdot 1 \cdot a_{1 \cdot n/1^2} = a_n$. □

Let $\mathbb{T}'$ denote the image of the Hecke algebra in $\mathrm{End}(S_2(\Gamma_0(N)))$, and let $\mathbb{T}'_{\mathbb{C}} = \mathbb{T}' \otimes \mathbb{C}$ be the $\mathbb{C}$-span of the Hecke operators.

Proposition 3.4.2. *There is a perfect bilinear pairing of complex vector spaces*

$$S_2(\Gamma_0(N)) \times \mathbb{T}'_{\mathbb{C}} \to \mathbb{C}$$

given by

$$\langle f, t \rangle = a_1(t(f)).$$

Proof. The pairing is bilinear since both t and a_1 are linear. Suppose $f \in S_2(\Gamma_0(N))$ is such that $\langle f, t \rangle = 0$ for all $t \in \mathbb{T}'_{\mathbb{C}}$. Then in particular $\langle f, T_n \rangle = 0$ for each positive integer n. But by Lemma 3.4.1 we have

$$a_n(f) = a_1(T_n(f)) = 0$$

for all n; thus $f = 0$.

Next suppose that $t \in \mathbb{T}'_{\mathbb{C}}$ is such that $\langle f, t \rangle = 0$ for all $f \in S_2(\Gamma_0(N))$. Then $a_1(t(f)) = 0$ for all f. For any n, the image $T_n(f)$ is also a cuspform, so $a_1(t(T_n(f))) = 0$ for all n and f. Finally $\mathbb{T}'$ is commutative and Lemma 3.4.1 together imply that for all n and f,

$$0 = a_1(t(T_n(f))) = a_1(T_n(t(f))) = a_n(t(f)),$$

so $t(f) = 0$ for all f. Thus t is the 0 operator. □

By Proposition 3.4.2 there is an isomorphism of vector spaces

$$\Psi : S_2(\Gamma_0(N)) \xrightarrow{\cong} \mathrm{Hom}(\mathbb{T}', \mathbb{C})$$

that sends $f \in S_2(\Gamma_0(N))$ to the homomorphism

$$t \mapsto a_1(t(f)).$$

For any linear map $\varphi : \mathbb{T}'_{\mathbb{C}} \to \mathbb{C}$, let

$$f_\varphi = \sum_{n=1}^{\infty} \varphi(T_n) q^n \in \mathbb{C}[[q]].$$

By Lemma 3.4.1, we have

$$\langle f_\varphi, T_n \rangle = a_1(T_n(f_\varphi)) = a_n(f_\varphi) = \varphi(T_n).$$

Thus f_φ must be the q-expansion of the modular form that corresponds to φ under the isomorphism Ψ. In paritcular, $f_\varphi \in S_2(\Gamma_0(N))$, and the cuspforms f_φ, as φ runs through a basis, form a basis for $S_2(\Gamma_0(N))$.

> *We can compute $S_2(\Gamma_0(N))$ by computing* $\mathrm{Hom}(\mathbb{T}', \mathbb{C})$*, where we compute $\mathbb{T}'$ in any way we want, e.g., using a space that contains an isomorphic copy of $S_2(\Gamma_0(N))$.*

Algorithm 3.4.3 (Basis of Cuspforms). *Given a positive integers N and B, this algorithm computes a basis for $S_2(\Gamma_0(N))$ to precision $O(q^B)$.*

1. Compute the modular symbols space $\mathbb{M}_2(\Gamma_0(N); \mathbb{Q})$ via the presentation of Section 3.2.2.
2. Compute the subspace $\mathbb{S}_2(\Gamma_0(N); \mathbb{Q})$ of cuspidal modular symbols as in Section 3.3.
3. Let $d = \frac{1}{2} \cdot \dim \mathbb{S}_2(\Gamma_0(N); \mathbb{Q})$. This is the dimension of $S_2(\Gamma_0(N))$.
4. Use the Hecke operators T_2, T_3, etc., of Section 3.2.3 to find the unique subspace V of $\mathrm{Hom}(\mathbb{M}_2(\Gamma_0(N); \mathbb{Q}), \mathbb{Q})$ that is isomorphic to $\mathbb{S}_2(\Gamma_0(N); \mathbb{Q})$ as a $\mathbb{T}$-module. (The Hecke operators act via their transpose; find the subspace V of the dual with the same characteristic polynomials.)
5. Let $[T_n]$ denote the matrix of T_n acting on some fixed basis of V. For a matrix A, let $a_{ij}(A)$ denote the ij-th entry of A. For various integers i, j with $0 \leq i, j \leq d-1$, compute formal q-expansions

$$f_{ij}(q) = \sum_{n=1}^{B-1} a_{ij}([T_n])q^n + O(q^B) \in \mathbb{Q}[[q]]$$

until we find enough to span a space of dimension d (or exhaust all of them, in which case B is too small). These f_{ij} then form a basis for $S_2(\Gamma_0(N))$.

3.4.1 Examples

In this section we use SAGE to demonstrate Algorithm 3.4.3 for computing $S_2(\Gamma_0(N))$ for various N.

Example 3.4.4. The smallest N with $S_2(\Gamma_0(N)) \neq 0$ is $N = 11$.

```
sage: M = ModularSymbols(11)
sage: M.basis()
((1,0), (1,8), (1,9))
sage: S = M.cuspidal_subspace()
sage: S
Dimension 2 subspace of a modular symbols space of level 11
sage: S.basis()
((1,8), (1,9))
sage: d = S.dimension() // 2; d
1
```

The command `dual_free_module` computes the vector space V of Algorithm 3.4.3.

```
sage: S.dual_free_module()
Vector space of degree 3 and dimension 2 over Rational Field
Basis matrix:
[1 0 5]
[0 1 0]
```

View each of the basis vectors $(1, 0, 5)$ and $(0, 1, 0)$ as defining a linear map (via dot product) $\mathbb{S}_2(\Gamma_0(11)) \to \mathbb{Q}$, where we view elements of $\mathbb{S}_2(\Gamma_0(11))$ as linear combinations of our fixed basis $(1, 0)$, $(1, 8)$, $(1, 9)$ for $\mathbb{M}_2(\Gamma_0(11))$.

The command `dual_hecke_matrix` computes the matrix of T_n on the above basis for V.

```
sage: S.dual_hecke_matrix(1)
[1 0]
[0 1]
sage: S.dual_hecke_matrix(2)
[-2  0]
[ 0 -2]
sage: S.dual_hecke_matrix(3)
[-1  0]
[ 0 -1]
```

Thus

$$f_{0,0} = q - 2q^2 - q^3 + \cdots \in S_2(\Gamma_0(11)).$$

Since $\dim S_2(\Gamma_0(11)) = 1$, this form must be a basis.

Example 3.4.5. Next consider $N = 23$, where we have $d = \dim S_2(\Gamma_0(23)) = 2$. The command `q_expansion_cuspforms` computes V and the matrices $[T_n]|V$ and returns a function f such that $f(i, j)$ is the q-expansion of $f_{i,j}$ to some precision.

```
sage: M = ModularSymbols(23)
sage: S = M.cuspidal_subspace()
sage: S
Dimension 4 subspace of a modular symbols space of level 23
sage: f = S.q_expansion_cuspforms(6)
sage: f(0,0)
q - 2/3*q^2 + 1/3*q^3 - 1/3*q^4 - 4/3*q^5 + O(q^6)
sage: f(0,1)
O(q^6)
sage: f(1,0)
-1/3*q^2 + 2/3*q^3 + 1/3*q^4 - 2/3*q^5 + O(q^6)
```

Thus a basis for $S_2(\Gamma_0(23))$ is

$$f_{0,0} = q - \frac{2}{3}q^2 + \frac{1}{3}q^3 - \frac{1}{3}q^4 - \frac{4}{3}q^5 + \cdots$$
$$f_{1,0} = -\frac{1}{3}q^2 + \frac{2}{3}q^3 + \frac{1}{3}q^4 - \frac{2}{3}q^5 + \cdots$$

Or, in echelon form,

$$q - q^3 - q^4 + \cdots$$
$$q^2 - 2q^3 - q^4 + 2q^5 + \cdots$$

which we computed using

```
sage: S.q_expansion_basis(6)
 [q - q^3 - q^4 + O(q^6),
    q^2 - 2*q^3 - q^4 + 2*q^5 + O(q^6)]
```

3.5 Computing $S_2(\Gamma_0(N))$ using eigenvectors

In this section we describe how to use modular symbols to construct a basis of $S_2(\Gamma_0(N))$ consisting of modular forms that are eigenvectors for every element of the ring $\mathbb{T}'$ generated by the Hecke operator T_p, with $p \nmid N$. Such eigenvectors are called *eigenforms*.

Suppose M is a positive integer that divides N. As explained in [Lan95, VIII.1–2], for each divisor d of N/M there is a natural *degeneracy map* $\beta_{M,d} : S_2(M) \to S_2(\Gamma_0(N))$ given by $\beta_{M,d}(f(q)) = f(q^d)$. The *new subspace* of $S_2(\Gamma_0(N))$, denoted $S_2(\Gamma_0(N))^{\text{new}}$, is the complementary $\mathbb{T}$-submodule of the $\mathbb{T}$-module generated by the images of all maps $\beta_{M,d}$, with M and d as above. (It is a nontrivial fact that this complement is well defined; one possible proof uses the Petersson inner product.)

The theory of Atkin and Lehner [AL70] (see Section 6.1.1) asserts that, as a $\mathbb{T}'$-module, $S_2(\Gamma_0(N))$ decomposes as follows:

$$S_2(\Gamma_0(N)) \quad = \bigoplus_{M|N,\ d|N/M} \beta_{M,d}(S_2(M)^{\text{new}}).$$

To compute $S_2(\Gamma_0(N))$ it thus suffices to compute $S_2(M)^{\text{new}}$ for each positive divisor M of N.

We now turn to the problem of computing $S_2(\Gamma_0(N))^{\text{new}}$. Atkin and Lehner [AL70] also proved that $S_2(\Gamma_0(N))^{\text{new}}$ is spanned by eigenforms, each of which occurs with multiplicity one in $S_2(\Gamma_0(N))^{\text{new}}$. Moreover, if $f \in S_2(\Gamma_0(N))^{\text{new}}$ is an eigenform then the coefficient of q in the q-expansion of f is nonzero, so it is possible to normalize f so that coefficient of q is 1. With f so normalized, if $T_p(f) = a_p f$, then the pth Fourier coefficient of f is a_p. If $f = \sum_{n=1}^{\infty} a_n q^n$ is a normalized eigenvector for all T_p, then the a_n, with n composite, are determined by the a_p, with p prime, by the following formulas: $a_{nm} = a_n a_m$ when n and m are relatively prime, and $a_{p^r} = a_{p^{r-1}} a_p - p a_{p^{r-2}}$ for $p \nmid N$ prime. When $p \mid N$, $a_{p^r} = a_p^r$. We conclude that in order to compute $S_2(\Gamma_0(N))^{\text{new}}$, it suffices to compute all systems of eigenvalues $\{a_2, a_3, a_5, \ldots\}$ of the Hecke operators $T_2, T_3, T_5, \ldots$ acting on $S_2(\Gamma_0(N))^{\text{new}}$. Given a system of eigenvalues, the corresponding eigenform is $f = \sum_{n=1}^{\infty} a_n q^n$, where the a_n, for n composite, are determined by the recurrence given above.

In light of the pairing $\langle\, ,\, \rangle$ introduced in Section 3.1, computing the above systems of eigenvalues $\{a_2, a_3, a_5, \ldots\}$ amounts to computing the systems of eigenvalues of the Hecke operators T_p on the subspace V of $\mathbb{S}_2(\Gamma_0(N))$ that corresponds to the new subspace of $S_2(\Gamma_0(N))$. For each proper divisor M of N and each divisor d of N/M, let $\phi_{M,d} : \mathbb{S}_2(\Gamma_0(N)) \to \mathbb{S}_2(\Gamma_0(M))$ be the map sending x to $\left(\begin{smallmatrix} d & 0 \\ 0 & 1 \end{smallmatrix}\right) x$. Then V is the intersection of the kernels of all maps $\phi_{M,d}$.

The computation of the systems of eigenvalues of a collection of commuting diagonalizable endomorphisms involves standard linear algebra techniques, such as computation of characteristic polynomials and kernels of matrices. There are, however, several tricks that greatly speed up this process, some of which are described in Chapter 7.

Example 3.5.1. All forms in $S_2(\Gamma_0(39))$ are new. Up to Galois conjugacy, the eigenvalues of the Hecke operators T_2, T_3, T_5, and T_7 on $\mathbb{S}_2(\Gamma_0(39))$ are $\{1, -1, 2, -4\}$ and $\{a, 1, -2a-2, 2a+2\}$, where $a^2 + 2a - 1 = 0$. Each of these eigenvalues occur in $\mathbb{S}_2(\Gamma_0(39))$ with multiplicity two; for example, the characteristic polynomial of T_2 on $\mathbb{S}_2(\Gamma_0(39))$ is $(x-1)^2 \cdot (x^2 + 2x - 1)^2$. Thus $S_2(\Gamma_0(39))$ is spanned by

$$f_1 = q + q^2 - q^3 - q^4 + 2q^5 - q^6 - 4q^7 + \cdots,$$
$$f_2 = q + aq^2 + q^3 + (-2a-1)q^4 + (-2a-2)q^5 + aq^6 + (2a+2)q^7 + \cdots,$$

and the Galois conjugate of f_2.

3.5.1 Summary

To compute the q-expansion, to some precision, of each eigenforms in $S_2(\Gamma_0(N))$, we use the degeneracy maps so that we only have to solve the problem for $S_2(\Gamma_0(N))^{\text{new}}$. Here, using modular symbols, we compute all systems of eigenvalues $\{a_2, a_3, a_5, \ldots\}$, then write down each of the corresponding eigenforms $f = q + a_2 q^2 + a_3 q^3 + \cdots$.

3.6 Exercises

3.1 Let p be a prime.

   (a) List representative elements of $\mathbb{P}^1(\mathbb{Z}/3\mathbb{Z})$.

   (b) What is the cardinality of $\mathbb{P}^1(\mathbb{Z}/p\mathbb{Z})$ as a function of p?

(c) Prove that there is a bijection between the right cosets of $\Gamma_0(p)$ in $\mathrm{SL}_2(\mathbb{Z})$ and the elements of $\mathbb{P}^1(\mathbb{Z}/p\mathbb{Z})$. (As mentioned in this chapter this is also true for composite level; see [Cre97a, §2.2] for complete details.)

3.2 Use the inductive proof of Proposition 3.2.5 to write $\{0, 4/7\}$ in terms of Manin symbols for $\Gamma_0(7)$.

3.3 Show that the Hecke operator T_2 acts as multiplication by 3 on the space $\mathbb{M}_2(\Gamma_0(3))$ as follows:

(a) Write down right coset representatives for $\Gamma_0(3)$ in $\mathrm{SL}_2(\mathbb{Z})$.

(b) List all 8 relations coming from 3.2.7.

(c) Find a single Manin symbols $[r_i]$ so that the three other Manin symbols are a nonzero multiple of $[r_i]$ modulo the relations found in the previous step.

(d) Use formula (3.2.2) to compute the image of your symbol $[r_i]$ under T_2. You will obtain a sum of four symbols. Using the relations above, write this sum as a multiple of $[r_i]$. (The multiple must be 3 or you made a mistake.)

Chapter 4

Dirichlet Characters

In this chapter we develop a systematic theory for computing with Dirichlet characters, which are extremely important to computations with modular forms for (at least) two reasons:

- To compute the Eisenstein subspace $E_k(\Gamma_1(N))$ of $M_k(\Gamma_1(N))$ we explicitly write down Eisenstein series attached to pairs of Dirichlet characters (see Chapter 5).

- To compute $S_k(\Gamma_1(N))$, we instead compute a decomposition

$$M_k(\Gamma_1(N)) = \bigoplus M_k(\Gamma_1(N), \varepsilon)$$

  then compute each factor. Here the sum is over all Dirichlet characters ε modulo N.

Example 4.0.1. Expanding on the second point, the spaces $M_k(\Gamma_1(N), \varepsilon)$ are frequently much easier to compute with than the full $M_k(\Gamma_1(N))$. As we will see, if $\varepsilon = 1$ is the trivial character, then $M_k(\Gamma_1(N), 1) = M_k(\Gamma_0(N))$, which has much smaller dimension than $M_k(\Gamma_1(N))$. For example, $M_2(\Gamma_1(100))$ has dimension 370, whereas $M_2(\Gamma_1(100), 1)$ has dimension only 24, and $M_2(\Gamma_1(389))$ has dimension 6499, whereas $M_2(\Gamma_1(389), 1)$ has dimension only 33.

```
sage: dimension_modular_forms(Gamma1(100),2)
370
sage: dimension_modular_forms(Gamma0(100),2)
24
sage: dimension_modular_forms(Gamma1(389),2)
6499
sage: dimension_modular_forms(Gamma0(389),2)
33
```

4.1 The Definition

Fix an integral domain R and a root ζ of unity in R.

Definition 4.1.1 (Dirichlet Character). A *Dirichlet character* modulo N over R is a map $\varepsilon : \mathbb{Z} \to R$ such that there is a homomorphism $f : (\mathbb{Z}/N\mathbb{Z})^* \to \langle\zeta\rangle$ for which

$$\varepsilon(a) = \begin{cases} 0 & \text{if } (a, N) > 1, \\ f(a \bmod N) & \text{if } (a, N) = 1. \end{cases}$$

We denote the group of such Dirichlet characters by $D(N, R)$. Note that elements of $D(N, R)$ are in bijection with homomorphisms $(\mathbb{Z}/N\mathbb{Z})^* \to \langle\zeta\rangle$.

One familiar example of a Dirichlet characters is the Legendre symbol $\left(\frac{a}{p}\right)$ that appears in quadratic reciprocity theory. It is a Dirichlet character modulo p that takes the value 1 on integers that are congruent to a nonzero square modulo p, the value -1 on integers that are congruent to a nonzero non-square modulo p, and 0 on integers divisible by p.

4.2 Dirichlet Characters in SAGE

To create a Dirichlet character in SAGE you first create the group $D(N, R)$ of Dirichlet characters, then obtain elements of that group. First we make $D(11, \mathbb{Q})$:

```
sage: G = DirichletGroup(11, RationalField())
sage: G
Group of Dirichlet characters of modulus 11 over Rational Field
```

A Dirichlet character prints as a matrix that gives the values of the character on canonical generators of $(\mathbb{Z}/N\mathbb{Z})^*$ (as discussed below).

```
sage: list(G)
[[1], [-1]]
sage: eps = G.0      # 0th generator for Dirichlet group
sage: eps
[-1]
```

The character takes the value -1 on the unit generator.

```
sage: G.unit_gens()
[2]
sage: eps(2)
-1
sage: eps(3)
1
```

It is 0 on any integer not coprime to 11:

```
sage: eps(22)
0
```

We can also create groups of Dirichlet characters taking values in other rings or fields. For example, we create the cyclotomic field $\mathbb{Q}(\zeta_4)$.

```
sage: R = CyclotomicField(4)
sage: CyclotomicField(4)
Cyclotomic Field of order 4 and degree 2
```

Then we define $G = D(15, \mathbb{Q}(\zeta_4))$.

```
sage: G = DirichletGroup(15, R)
sage: G
Group of Dirichlet characters of modulus 15 over Cyclotomic Field
of order 4 and degree 2
```

And we list each of its elements.

```
sage: list(G)
[[1, 1], [-1, 1], [1, zeta_4], [-1, zeta_4], [1, -1], [-1, -1],
 [1, -zeta_4], [-1, -zeta_4]]
```

Now lets evaluate the second generator of G on various integers:

```
sage: e = G.1
sage: e(4)
-1
sage: e(-1)
-1
sage: e(5)
0
```

Finally we make a list of all the values of e.

```
sage: [e(n) for n in range(15)]
[0, 1, zeta_4, 0, -1, 0, 0, zeta_4, -zeta_4,
    0, 0, 1, 0, -zeta_4, -1]
```

We can also compute with groups of Dirichlet characters with values in a finite field.

```
sage: G = DirichletGroup(15, GF(5))
sage: G
Group of Dirichlet characters of modulus 15 over Finite field of size 5
```

We list all the elements of G, again represented by matrices that give the images of each unit generator, as an element of $\mathbb{F}_5$.

```
sage: list(G)
  [[1, 1], [4, 1], [1, 2], [4, 2], [1, 4], [4, 4], [1, 3], [4, 3]]
```

We evaluate the second generator of G on several integers.

```
sage: e = G.1
sage: e(-1)
4
sage: e(2)
2
sage: e(5)
0
sage: print [e(n) for n in range(15)]
[0, 1, 2, 0, 4, 0, 0, 2, 3, 0, 0, 1, 0, 3, 4]
```

4.3 Representing Dirichlet Characters

Lemma 4.3.1. *The groups $(\mathbb{Z}/N\mathbb{Z})^*$ and $D(N, \mathbb{C})$ are non-canonically isomorphic.*

Proof. This follows from the more general fact that for any finite abelian group G, we have that $G \approx \mathrm{Hom}(G, \mathbb{C}^*)$. To prove that this latter non-canonical isomorphism exists, first reduce to the case when G is cyclic of order n, in which case the statement follows because $\mathbb{C}^*$ contains the nth root of unity $e^{2\pi i/n}$, so $\mathrm{Hom}(G, \mathbb{C}^*)$ is also cyclic of order n. $\square$

Corollary 4.3.2. *We have $\#D(N, R) \mid \varphi(N)$, with equality if and only if the order of our choice of $\zeta \in R$ is a multiple of the exponent of the group $(\mathbb{Z}/N\mathbb{Z})^*$.*

Example 4.3.3. The group $D(5, \mathbb{C})$ has elements $\{[1], [i], [-1], [-i]\}$, so is cyclic of order $\varphi(5) = 4$. In contrast, the group $D(5, \mathbb{Q})$ has only the two elements $[1]$ and $[-1]$ and order 2. In SAGE the command `DirichletGroup(N)` with no second argument create the group of Dirichlet characters with values in the cyclotomic field $\mathbb{Q}(\zeta_n)$, where n is the exponent of the group $(\mathbb{Z}/N\mathbb{Z})^*$. Every element in $D(N, \mathbb{C})$ takes values in $\mathbb{Q}(\zeta_n)$, so $D(N, \mathbb{Q}(\zeta_n)) \cong D(N, \mathbb{C})$.

```
sage: list(DirichletGroup(5))
[[1], [zeta_4], [-1], [-zeta_4]]
sage: list(DirichletGroup(5, Q))
[[1], [-1]]
```

Fix a positive integer N, and write $N = \prod_{i=0}^{n} p_i^{e_i}$ where $p_0 < p_1 < \cdots < p_n$ are the prime divisors of N. By Exercise 4.1, each factor $(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^*$ is a cyclic group $C_i = \langle g_i \rangle$, except if $p_0 = 2$ and $e_0 \geq 3$, in which case $(\mathbb{Z}/p_0^{e_0}\mathbb{Z})^*$ is a product of the cyclic subgroup $C_0 = \langle -1 \rangle$ of order 2 with the cyclic subgroup $C_1 = \langle 5 \rangle$. In all cases we have

$$(\mathbb{Z}/N\mathbb{Z})^* \cong \prod_{0 \leq i \leq n} C_i = \prod_{0 \leq i \leq n} \langle g_i \rangle.$$

For i such that $p_i > 2$, choose the generator g_i of C_i to be the element of $\{2, 3, \ldots, p_i^{e_i} - 1\}$ that is smallest and generates. Finally, use the Chinese Remainder Theorem (see [Coh93, §1.3.3])) to lift each g_i to an element in $(\mathbb{Z}/N\mathbb{Z})^*$, also denoted g_i, that is 1 modulo each $p_j^{e_j}$ for $j \neq i$.

Algorithm 4.3.4 (Minimal generator for $(\mathbb{Z}/p^r\mathbb{Z})^*$). *Given an odd prime power p^r, this algorithm computes the minimal generator for $(\mathbb{Z}/p^r\mathbb{Z})^*$.*

1. [Factor Group Order] Factor $n = \phi(p^r) = p^{r-1} \cdot 2 \cdot ((p-1)/2)$ as a product $\prod p_i^{e_i}$ of primes. This is equivalent in difficulty to factoring $(p-1)/2$. (See, e.g., [Coh93, Ch.8, 10] for integer factorization algorithms.)
2. [Initialize] Set $g = 2$.
3. [Generator?] Using the binary powering algorithm (see [Coh93, §1.2]), compute $g^{n/p_i} \pmod{p^r}$, for each prime divisor p_i of n. If any of these powers are 1, set $g = g + 1$ and go to Step 2. If no powers are 1, output g and terminate.

For the proof, see Exercise 4.2.

Example 4.3.5. A minimal generator for $(\mathbb{Z}/49\mathbb{Z})^*$ is 3. We have $n = \varphi(49) = 42 = 2 \cdot 3 \cdot 7$, and

$$2^{n/2} \equiv 1, \qquad 2^{n/3} \equiv 18, \qquad 2^{n/7} \equiv 15 \pmod{49}.$$

so 2 is not a generator for $(\mathbb{Z}/49\mathbb{Z})^*$. (We see this just from $2^{n/2} \equiv 1 \pmod{49}$.) However 3 is since

$$3^{n/2} \equiv 48, \qquad 3^{n/3} \equiv 30, \qquad 3^{n/7} \equiv 43 \pmod{49}.$$

Example 4.3.6. In this example we compute minimal generators for $N = 25$, 100, and 200:

1. The minimal generator for $(\mathbb{Z}/25\mathbb{Z})^*$ is 2.
2. Minimal generators for $(\mathbb{Z}/100\mathbb{Z})^*$, lifted to numbers modulo 100, are $g_0 = 51$ and $g_1 = 77$. Notice that $g_0 \equiv -1 \pmod 4$ and $g_0 \equiv 1 \pmod{25}$, and $g_1 \equiv 2 \pmod{25}$ is the minimal generator modulo 25.
3. Minimal generators for $(\mathbb{Z}/200\mathbb{Z})^*$, lifted to numbers modulo 200, are $g_0 = 151$, $g_1 = 101$, and $g_2 = 177$. Note that $g_0 \equiv -1 \pmod 4$, that $g_1 \equiv 5 \pmod 8$, and $g_2 \equiv 2 \pmod{25}$.

The command `Integers(N)` creates $\mathbb{Z}/N\mathbb{Z}$.

```
sage: R = Integers(49)
sage: R
Ring of integers modulo 49
```

The `unit_gens()` command computes the unit generators as defined above.

```
sage: R.unit_gens()
[3]
sage: Integers(25).unit_gens()
[2]
sage: Integers(100).unit_gens()
[51, 77]
sage: Integers(200).unit_gens()
[151, 101, 177]
sage: Integers(2005).unit_gens()
[402, 1206]
sage: Integers(200000000).unit_gens()
[174218751, 51562501, 187109377]
```

Fix an element ζ of finite multiplicative order in a ring R, and let $D(N, R)$ denote the group of Dirichlet characters modulo N over R, with image in $\langle\zeta\rangle \cup \{0\}$. We specify an element $\varepsilon \in D(N, R)$ by giving the list

$$[\varepsilon(g_0), \varepsilon(g_1), \ldots, \varepsilon(g_n)] \tag{4.3.1}$$

of images of the generators of $(\mathbb{Z}/N\mathbb{Z})^*$. (Note if N is even, the number of elements of the list (4.3.1) does *not* depend on whether or not $8 \mid N$—there are always two factors corresponding to 2.) This representation completely determines ε and is convenient for arithmetic operations with Dirichlet characters. It is analogous to representing a linear transformation by a matrix. See Section 4.7 for a discussion of alternative ways to represent Dirichlet characters.

4.4 Evaluation of Dirichlet Characters

This section is about how to compute $\varepsilon(n)$, where ε is a Dirichlet character and n is an integer. We begin with an example.

Example 4.4.1. If $N = 200$, then $g_0 = 151$, $g_1 = 101$ and $g_2 = 177$, as we saw in Example 4.3.6. The exponent of $(\mathbb{Z}/200\mathbb{Z})^*$ is 20, since that is the least common multiple of the exponents of $4 = \#(\mathbb{Z}/8\mathbb{Z})^*$ and $20 = \#(\mathbb{Z}/25\mathbb{Z})^*$. The orders of g_0, g_1 and g_2 are 2, 2, and 20. Let $\zeta = \zeta_{20}$ be a primitive 20th root of unity in $\mathbb{C}$. Then the following are generators for $D(200, \mathbb{C})$:

$$\varepsilon_0 = [-1, 1, 1], \qquad \varepsilon_1 = [1, -1, 1], \qquad \varepsilon_2 = [1, 1, \zeta],$$

and $\varepsilon = [1, -1, \zeta^5]$ is an example element of order 4. To evaluate $\varepsilon(3)$, we write 3 in terms of g_0, g_1, and g_2. First, reducing 3 modulo 8, we see that $3 \equiv g_0 \cdot g_1 \pmod{8}$. Next reducing 3 modulo 25, and trying powers of $g_2 = 2$, we find that $e \equiv g_2^7 \pmod{25}$. Thus

$$\begin{aligned}\varepsilon(3) &= \varepsilon(g_0 \cdot g_1 \cdot g_2^7)\\ &= \varepsilon(g_0)\varepsilon(g_1)\varepsilon(g_2)^7\\ &= 1 \cdot (-1) \cdot (\zeta^5)^7\\ &= -\zeta^{35} = -\zeta^{15}.\end{aligned}$$

We next illustrate the above computation of $\varepsilon(3)$ in SAGE. First we make the group $D(200, \mathbb{Q}(\zeta_8))$, and list its generators.

```
sage: G = DirichletGroup(200)
sage: G
Group of Dirichlet characters of modulus 200 over Cyclotomic Field
      of order 20 and degree 8
sage: G.exponent()
20
sage: G.gens()
[[-1, 1, 1], [1, -1, 1], [1, 1, zeta_20]]
```

Next we construct ε.

```
sage: K = G.base_ring()
sage: zeta = K.gen()
sage: eps = G([1,-1,zeta^5])
sage: eps
[1, -1, zeta_20^5]
```

Finally, we evaluate ε at 3.

```
sage: eps(3)
zeta_20^5
sage: -zeta^15
zeta_20^5
```

Example 4.4.1 illustrates that if ε is represented using a list as described above, evaluation of ε on an arbitrary integer is inefficient without extra information; it requires solving the discrete log problem in $(\mathbb{Z}/N\mathbb{Z})^*$. In fact, for a general character ε calculation of ε will probably be at least as hard as finding discrete logarithms no matter what representation we use (quadratic characters are easier—see Algorithm 4.4.5).

Algorithm 4.4.2 (Evaluate ε). *Given a Dirichlet character ε modulo N, represented by a list $[\varepsilon(g_0), \varepsilon(g_1), \ldots, \varepsilon(g_n)]$, and an integer a, this algorithm computes $\varepsilon(a)$.*

1. [GCD] Compute $g = \gcd(a, N)$. If $g > 1$, output 0 and terminate.
2. [Discrete Log] For each i, write $a \pmod{p_i^{e_i}}$ as a power m_i of g_i using some algorithm for solving the discrete log problem (see below). (If $p_i = 2$, write $a \pmod{p_i^{e_i}}$ as $(-1)^{m_0} \cdot 5^{m_1}$.) This step is analogous to writing a vector in terms of a basis.
3. [Multiply] Compute and output $\prod \varepsilon(g_i)^{m_i}$ as an element of R, and terminate. This is analogous to multiplying a matrix times a vector.

By Exercise 4.3 we have an isomorphism of groups

$$(1 + p^{n-1}(\mathbb{Z}/p^n\mathbb{Z}), \times) \cong (\mathbb{Z}/p\mathbb{Z}, +),$$

so one sees by induction that Step 2 is "about as difficult" as finding a discrete log in $(\mathbb{Z}/p\mathbb{Z})^*$. There is an algorithm called "baby-step giant-step", which solves the discrete log problem in $(\mathbb{Z}/p\mathbb{Z})^*$ in time $O(\sqrt{\ell})$, where ℓ is the largest prime factor of $p-1 = \#(\mathbb{Z}/p\mathbb{Z})^*$ (note that the discrete log problem in $(\mathbb{Z}/p\mathbb{Z})^*$ reduces to a series of discrete log problems in each prime order cyclic factor). This is unfortunately still exponential in the number of digits of ℓ.

Algorithm 4.4.3 (Baby-Step Giant Step Discrete Log). *Given a prime p, a generator g of $(\mathbb{Z}/p\mathbb{Z})^*$, and an element $a \in (\mathbb{Z}/p\mathbb{Z})^*$, this algorithm finds an n such that $g^n = a$. (Note that this algorithm works in any cyclic group, not just $(\mathbb{Z}/p\mathbb{Z})^*$.)*

1. [Make Lists] Let $m = \lceil \sqrt{p} \rceil$ be the ceiling of $\sqrt{p}$, and construct two lists

   $$g,\, g^m,\, \ldots,\, g^{(m-1)m}, g^{m^2} \qquad \text{(giant steps)}$$

   and

   $$ag,\, ag^2,\, \ldots,\, ag^{m-1}, ag^m \qquad \text{(baby steps)}.$$

2. [Find Match] Sort the two lists and find a match $g^{im} = ag^j$. Then $a = g^{im-j}$.

Proof. We prove that there will always be a match. Since we know that $a = g^k$ for some k with $0 \le k \le p-1$ and any such k can be written in the form $im - j$ for $0 \le i, j \le m-1$, we will find such a match. □

Algorithm 4.4.3 uses nothing special about $(\mathbb{Z}/p\mathbb{Z})^*$, so it works in a generic group. It is a theorem that there is no faster algorithm to find discrete logs in a "generic group" (see [Sho97, Nec94]). Fortunately there are much better subexponential algorithms for solving the discrete log problem in $(\mathbb{Z}/p\mathbb{Z})^*$, which use the special structure of this group. They use the number field sieve (see e.g., [Gor93]), which is also the best known algorithm for factoring integers.

This class of algorithms has been very well studied by cryptographers; though sub-exponential, solving discrete log problems when p is large is still extremely difficult. For a more in-depth survey see [Gor04].

The specific applications of Dirichlet characters in this book involve computing modular forms, and for these applications N will be fairly small, e.g., $N < 10^6$. Also we will evaluate ε on a *huge* number of random elements, inside inner loops of algorithms. Thus for our purposes it will often be better to make a table of all values of ε, so that evaluation of ε is extremely fast. The following algorithm computes a table of all values of ε, and it does not require computing any discrete logs since we are computing *all* values.

Algorithm 4.4.4 (Values of ε). *Given a Dirichlet character ε represented by the list of values of ε on the minimal generators g_i of $(\mathbb{Z}/N\mathbb{Z})^*$, this algorithm creates a list of all the values of ε.*

1. [Initialize] For each minimal generator g_i, set $a_i = 0$. Let $n = \prod g_i^{a_i}$, and set $z = 1$. Create a list v of N values, all initially set equal to 0. When this algorithm terminates the list v will have the property that
$$v\,[x \pmod N] = \varepsilon(x).$$
Notice that we index v starting at 0.
2. [Add Value to Table] Set $v[n] = z$.
3. [Finished?] If each a_i is one less than the order of g_i, output v and terminate.
4. [Increment] Set $a_0 = a_0 + 1$, $n = n \cdot g_0 \pmod N$, and $z = z \cdot \varepsilon(g_0)$. If $a_0 \geq \mathrm{ord}(g_0)$, set $a_0 \to 0$, then set $a_1 = a_1 + 1$, $n = n \cdot g_1 \pmod N$, and $z = z \cdot \varepsilon(g_1)$. If $a_1 \geq \mathrm{ord}(g_1)$, do what you just did with a_0, but with all subscripts replaced by 1. Etc. (Imagine a car odometer.) Go to Step 2.

Frequently people describe quadratic characters in terms of the Kronecker symbol. The following algorithm gives a way to go between the two representations.

Algorithm 4.4.5 (Kronecker Symbol). *Given an integer N, this algorithm computes a representation of the Kronecker symbol $\left(\frac{a}{N}\right)$ as a Dirichlet character.*

1. Compute the minimal generators g_i of $(\mathbb{Z}/N\mathbb{Z})^*$ using Algorithm 4.3.4.
2. Compute $\left(\frac{g_i}{N}\right)$ for each g_i using one of the algorithms of [Coh93, §1.1.4].

Remark 4.4.6. The algorithms in [Coh93, §1.1.4] for computing the Kronecker symbol run in time quadratic in the number of digits of the input, so they do not require computing discrete logarithms. (They use, e.g., that $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p$, when p is an odd prime.) If N is very large and we are only interested in evaluating $\varepsilon(a) = \left(\frac{a}{N}\right)$ for a few a, then viewing ε as a Dirichlet character in the sense of this chapter leads to a less efficient way to compute with ε. The algorithmic discussion of characters in this chapter is most useful for working with the full group of characters, and characters that cannot be expressed in terms of Kronecker characters.

Example 4.4.7. We compute the Dirichlet character associated to the Kronecker symbol $\left(\frac{a}{200}\right)$. We find that $\left(\frac{g_i}{200}\right)$, for $i = 0, 1, 2$, where the g_i are as in Example 4.4.1:

```
sage: kronecker(151,200)
1
sage: kronecker(101,200)
-1
sage: kronecker(177,200)
1
```

Thus the corresponding character is defined by $[1, -1, 1]$.

Remark 4.4.8 (Elkies). Jacobi reciprocity must be used to efficiently compute the Jacobi symbol $\left(\frac{m}{n}\right)$. It's faster than computing $a^{(p-1)/2}$ when p is prime, but more significantly it makes it possible to compute Jacobi symbols $\left(\frac{m}{n}\right)$ for all m, n without knowing the factorization of n—which of course would be a computation much longer than quadratic.

Example 4.4.9. We compute the character associated to $\left(\frac{a}{420}\right)$. We have $420 = 4 \cdot 3 \cdot 5 \cdot 7$, and minimal generators are

$$g_0 = 211, \quad g_1 = 1, \quad g_2 = 281, \quad g_3 = 337, \quad g_4 = 241.$$

We have $g_0 \equiv -1 \pmod 4$, $g_2 \equiv 2 \pmod 3$, $g_3 \equiv 2 \pmod 5$ and $g_4 \equiv 3 \pmod 7$. Using PARI again we find $\left(\frac{g_0}{420}\right) = \left(\frac{g_1}{420}\right) = 1$ and $\left(\frac{g_2}{420}\right) = \left(\frac{g_3}{420}\right) = \left(\frac{g_4}{420}\right) = -1$, so the corresponding character is $[1, 1, -1, -1, -1]$.

4.5 Conductors of Dirichlet Characters

The following algorithm for computing the order of ε reduces the problem to computing the orders of powers of ζ in R.

Algorithm 4.5.1 (Order of Character). *This algorithm computes the order of a Dirichlet character $\varepsilon \in D(N, R)$.*

1. Compute the order r_i of each $\varepsilon(g_i)$, for each minimal generator g_i of $(\mathbb{Z}/N\mathbb{Z})^*$. Since the order of $\varepsilon(g_i)$ is divisor of $n = \#(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^*$, we can compute its order by factoring n and considering the divisors of n.
2. Compute and output the least commmon multiple of the integers r_i.

Remark 4.5.2. Computing the order of $\varepsilon(g_i) \in R$ is potentially difficult and tedious. Using a different (simultaneous) representation of Dirichlet characters avoids having to compute the order of elements of R. See Section 4.7.

The next algorithm factors ε as a product of "local" characters, one for each prime divisor of N. It is useful for other algorithms, and also for explicit computations with the Hijikita trace formula (see [Hij74]). This factorization is easy to compute because of how we represent ε.

Algorithm 4.5.3 (Factorization of Character). *Given a Dirichlet character $\varepsilon \in D(N,R)$, with $N = \prod p_i^{e_i}$, this algorithm finds Dirichlet characters ε_i modulo $p_i^{e_i}$, such that for all $a \in (\mathbb{Z}/N\mathbb{Z})^*$, we have $\varepsilon(a) = \prod \varepsilon_i(a \pmod{p_i^{e_i}})$. If $2 \mid N$, the steps are as follows:*

1. Let g_i be the minimal generators of $(\mathbb{Z}/N\mathbb{Z})^*$, so ε is given by a list
$$[\varepsilon(g_0), \ldots, \varepsilon(g_n)].$$
2. For $i = 2, \ldots, n$, let ε_i be the element of $D(p_i^{e_i}, R)$ defined by the singleton list $[\varepsilon(g_i)]$.
3. Let ε_1 be the element of $D(2^{e_1}, R)$ defined by the list $[\varepsilon(g_0), \varepsilon(g_1)]$ of length 2. Output the ε_i and terminate.

If $2 \nmid N$, then omit Step 3, and include all i in Step 2.

The factorization of Algorithm 4.5.3 is unique since each ε_i is determined by the image of the canonical map $(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^*$ in $(\mathbb{Z}/N\mathbb{Z})^*$, which sends $a \pmod{p_i^{e_i}}$ to the element of $(\mathbb{Z}/N\mathbb{Z})^*$ that is $a \pmod{p_i^{e_i}}$ and $1 \pmod{p_j^{e_j}}$ for $j \neq i$.

Example 4.5.4. If $\varepsilon = [1, -1, \zeta^5] \in D(200, \mathbb{C})$, then $\varepsilon_1 = [1, -1] \in D(8, \mathbb{C})$ and $\varepsilon_2 = [\zeta^5] \in D(25, \mathbb{C})$.

Definition 4.5.5 (Conductor). The *conductor* of a Dirichlet character $\varepsilon \in D(N,R)$ is the smallest positive divisor $c \mid N$ such that there is a character $\varepsilon' \in D(c,R)$ for which $\varepsilon(a) = \varepsilon'(a)$ for all $a \in \mathbb{Z}$ with $(a,N) = 1$. A Dirichlet character is *primitive* if its modulus equals its conductor. The character ε' associated to ε with modulus equal to the conductor of ε is called the *primitive character associated to ε*.

We will be interested in conductors later, when computing new subspaces of spaces of modular forms with character. Also certain formulas for special values of L functions are only valid for primitive characters.

Algorithm 4.5.6 (Conductor). *This algorithm computes the conductor of a Dirichlet character $\varepsilon \in D(N,R)$.*

1. [Factor Character] Using Algorithm 4.5.3, find characters ε_i whose product is ε.
2. [Compute Orders] Using Algorithm 4.5.1, compute the orders r_i of each ε_i.
3. [Conductors of Factors] For each i, either set $c_i \to 1$ if ε_i is the trivial character (i.e., of order 1), or set $c_i = p_i^{\mathrm{ord}_{p_i}(r_i)+1}$, where $\mathrm{ord}_p(n)$ is the largest power of p that divides n.

4. [Adjust at 2?] If $p_1 = 2$ and $\varepsilon_1(5) \neq 1$, set $c_1 = 2c_1$.
5. [Finished] Output $c = \prod c_i$ and terminate.

Proof. Let ε_i be the local factors of ε, as in Step 1. We first show that the product of the conductors f_i of the ε_i is the conductor f of ε. Since ε_i factors through $(\mathbb{Z}/f_i\mathbb{Z})^*$, the product ε of the ε_i factors through $(\mathbb{Z}/\prod f_i\mathbb{Z})^*$, so the conductor of ε divides $\prod f_i$. Conversely, if $\mathrm{ord}_{p_i}(f) < \mathrm{ord}_{p_i}(f_i)$ for some i, then we could factor ε as a product of local (prime power) characters differently, which contradicts that this factorization is unique.

It remains to prove that if ε is a nontrivial character modulo p^n, where p is a prime, and r is the order of ε, then the conductor of ε is $p^{\mathrm{ord}_p(r)+1}$, except possibly if $8 \mid p^n$. Since the order and conductor of ε and of the associated primitive character ε' are the same, we may assume ε is primitive, i.e., that p^n is the conductor of ε; note that that $n > 0$, since ε is nontrivial.

First suppose p is odd. Then the abelian group $D(p^n, R)$ splits as a direct sum $D(p, R) \oplus D(p^n, R)'$, where $D(p^n, R)'$ is the p-power torsion subgroup of $D(p^n, R)$. Also ε has order $u \cdot p^m$, where u, which is coprime to p, is the order of the image of ε in $D(p, R)$ and p^m is the order of the image in $D(p^n, R)'$. If $m = 0$, then the order of ε is coprime to p, so ε is in $D(p, R)$, which means that $n = 1$, so $n = m + 1$, as required. If $m > 0$, then $\zeta \in R$ must have order divisible by p, so R has characteristic not equal to p. The conductor of ε does not change if we adjoin roots of unity to R, so in light of Lemma 4.3.1 we may assume that $D(N, R) \approx (\mathbb{Z}/N\mathbb{Z})^*$. It follows that for each $n' \leq n$, the p-power subgroup $D(p^{n'}, R)'$ of $D(p^{n'}, R)$ is the $p^{n'-1}$-torsion subgroup of $D(p^n, R)'$. Thus $m = n - 1$, since $D(p^n, R)'$ is by assumption the smallest such group that contains the projection of ε. This proves the formula of Step 3. We leave the argument when $p = 2$ as an exercise (see Exercise 4.4). □

Example 4.5.7. If $\varepsilon = [1, -1, \zeta^5] \in D(200, \mathbb{C})$, then as we saw in Example 4.5.4, ε is the product of $\varepsilon_1 = [1, -1]$ and $\varepsilon_2 = [\zeta^5]$. Because $\varepsilon_1(5) = -1$, the conductor of ε_1 is 8. The order of ε_2 is 4 (since ζ is a 20th root of unity), so the conductor of ε_2 is 5. Thus the conductor of ε is $40 = 8 \cdot 5$.

4.6 Restriction, Extension, and Galois Orbits

The following two algorithms restrict and extend characters to a compatible modulus. Using them it is easy to define multiplication of two characters $\varepsilon \in D(N, R)$ and $\varepsilon' \in D(N', R')$, as long as R and R' are subrings of a common ring. To carry out the multiplication, just extend bother characters to characters modulo $\mathrm{lcm}(N, N')$, then multiply.

Algorithm 4.6.1 (Restriction of Character). *Given a Dirichlet character $\varepsilon \in D(N, R)$ and a divisor N' of N that is a multiple of the conductor of ε, this algorithm finds a characters $\varepsilon' \in D(N', R)$, such that $\varepsilon'(a) = \varepsilon(a)$, for all $a \in \mathbb{Z}$ with $(a, N) = 1$.*

1. [Conductor] Compute the conductor of ε using Algorithm 4.5.6, and verify that indeed N' is divisible by the conductor and divides N.
2. [Minimal Generators] Compute the minimal generators g_i for $(\mathbb{Z}/N'\mathbb{Z})^*$.
3. [Values of Restriction] For each i, compute $\varepsilon'(g_i)$ as follows. Find a multiple aN' of N' such that $(g_i + aN', N) = 1$; then $\varepsilon'(g_i) = \varepsilon(g_i + aN')$.
4. [Output Character] Output the Dirichlet character modulo N' defined by $[\varepsilon'(g_0), \ldots, \varepsilon'(g_n)]$.

Proof. The only part that is not clear is that in Step 3 there is an a such that $(g_i + aN', N) = 1$. If we write $N = N_1 \cdot N_2$, with $(N_1, N_2) = 1$, and N_1 divisible by all primes that divide N', then $(g_i, N_1) = 1$ since $(g_i, N') = 1$. By the Chinese Remainder Theorem, there is an $x \in \mathbb{Z}$ such that $x \equiv g_i \pmod{N_1}$ and $x \equiv 1 \pmod{N_2}$. Then $x = g_i + bN_1 = g_i + (bN_1/N') \cdot N'$ and $(x, N) = 1$, which completes the proof. $\square$

Algorithm 4.6.2 (Extension of Character). *Given a Dirichlet character $\varepsilon \in D(N, R)$ and a multiple N' of N, this algorithm finds a characters $\varepsilon' \in D(N', R)$, such that $\varepsilon'(a) = \varepsilon(a)$, for all $a \in \mathbb{Z}$ with $(a, N') = 1$.*

1. [Minimal Generators] Compute the minimal generators g_i for $(\mathbb{Z}/N'\mathbb{Z})^*$.
2. [Evaluate] Compute $\varepsilon(g_i)$ for each i. Since $(g_i, N') = 1$, we also have $(g_i, N) = 1$.
3. [Output Character] Output the character defined by $[\varepsilon(g_0), \ldots, \varepsilon(g_n)]$.

We finish with an algorithm that computes the Galois orbit of an element in $D(N, R)$. This can be used to divide $D(N, R)$ up into Galois orbits, which is useful for modular forms computations, because, e.g., the spaces $M_k(\Gamma_1(N))(\varepsilon)$ and $M_k(\Gamma_1(N))(\varepsilon')$ are canonically isomorphic if ε and ε' are conjugate.

Algorithm 4.6.3 (Galois Orbit). *Given a Dirichlet character $\varepsilon \in D(N, R)$, this algorithm computes the orbit of ε under the action of $G = \mathrm{Gal}(\overline{F}/F)$, where F is the prime subfield of $\mathrm{Frac}(R)$, so $F = \mathbb{F}_p$ or $\mathbb{Q}$.*

1. [Order of ζ] Let n be the order of the chosen root $\zeta \in R$.
2. [Nontrivial Automorphisms] If $\mathrm{char}(R) = 0$, let
$$A = \{a : 2 \le a < n \text{ and } (a, n) = 1\}.$$
If $\mathrm{char}(R) = p > 0$, compute the multiplicative order r of p modulo n, and let
$$A = \{p^m : 1 \le m < r\}.$$
3. [Compute Orbit] Compute and output the *set* of unique elements ε^a for each $a \in A$ (there could be repeats, so we output unique elements only).

Proof. We prove that the nontrivial automorphisms of $\langle\zeta\rangle$ in characteristic p are as in Step 2. It is well-known that every automorphism in characteristic p on $\zeta \in \overline{\mathbb{F}}_p$ is of the form $x \mapsto x^{p^s}$, for some s. The images of ζ under such automorphisms are

$$\zeta, \zeta^p, \zeta^{p^2}, \ldots.$$

Suppose $r > 0$ is minimal such that $\zeta = \zeta^{p^r}$. Then the orbit of ζ is $\zeta, \ldots, \zeta^{p^{r-1}}$. Also $p^r \equiv 1 \pmod{n}$, where n is the multiplicative order of ζ, so r is the multiplicative order of p modulo n, which completes the proof. □

Example 4.6.4. The Galois orbits of characters in $D(20, \mathbb{C}^*)$ are as follows:

$$\begin{aligned}
G_0 &= \{[1,1,1]\},\\
G_1 &= \{[-1,1,1]\},\\
G_2 &= \{[1,1,\zeta_4],\ [1,1,-\zeta_4]\}\\
G_3 &= \{[-1,1,\zeta_4],\ [-1,1,-\zeta_4]\}\\
G_4 &= \{[1,1,-1]\},\\
G_5 &= \{[-1,1,-1]\}
\end{aligned}$$

The conductors of the characters in orbit G_0 are 1, in order G_1 are 4, in orbit G_2 they are 5, in G_3 they are 20, in G_4 the conductor is 5, and in G_5 the conductor is 20. (You should verify this.)

4.7 Alternative Representations of Characters

Let N be a positive integer and R an integral domain, with fixed root of unity ζ order n, and let $D(N,R) = D(N,R,\zeta)$. As in the rest of this chapter, write $N = \prod p_i^{e_i}$, and let $C_i = \langle g_i \rangle$ be the corresponding cyclic factors of $(\mathbb{Z}/N\mathbb{Z})^*$. In this section we discuss other ways to represent elements $\varepsilon \in D(N,R)$. Each representation has advantages and disadvantages, and no single representation is best. It emerged while writing this chapter that simultaneously using more than one representation of elements of $D(N,R)$ would be best. It is easy to convert between them, and some algorithms are much easier using one representation, than when using another. In this section we present two other representations, each which has advantages and disadvantages. But, we emphasize that there is frequently no reason to restrict to only one representation!

We could represent ε by giving a list $[b_0, \ldots, b_n]$, where each $b_i \in \mathbb{Z}/n\mathbb{Z}$ and $\varepsilon(g_i) = \zeta^{b_i}$. Then arithmetic in $D(N,R)$ is arithmetic in $(\mathbb{Z}/n\mathbb{Z})^{n+1}$, which is very efficient. A drawback to this approach is that it is easy to accidently consider sequences that do not actually correspond to elements of $D(N,R)$, though it is not really any easier to do this than with the representation we use elsewhere in this chapter. Also the choice of ζ is less clear, which can cause confusion. Finally, the orders of the local factors is more opaque, e.g., compare $[-1, \zeta_{40}]$ with $[20, 1]$. Overall this representation is not too bad, and is more like representing a linear transformation by a matrix. It has the **advantage** over the representation discussed earlier in this chapter that arithmetic in $D(N,R)$ is very efficient, and doesn't require any operations in the ring R; such operations could be quite slow, e.g., if R were a large cyclotomic field.

Another way to represent ε would be to give a list $[b_0,\ldots,b_n]$ of integers, but this time with $b_i \in \mathbb{Z}/\gcd(s_i,n)\mathbb{Z}$, where s_i is the order of g_i. Then

$$\varepsilon(g_i) = \zeta^{b_i \cdot n/(\gcd(s_i,n))},$$

which is already complicated enough to ring warning bells. With this representation we set up an identification

$$D(N,R) \cong \bigoplus_i \mathbb{Z}/\gcd(s_i,n)\mathbb{Z},$$

and arithmetic is efficient. This approach is seductive because every sequence of integers determines a character, and the sizes of the integers in the sequence nicely indicate the local orders of the character. However, giving analogues of many of the algorithms discussed in this chapter that operate on characters represented this way is tricky. For example, the representation depends very much on the order of ζ, so it is difficult to correctly compute natural maps $D(N,R) \to D(N,S)$, for $R \subset S$ rings, whereas for the representation elsewhere in this chapter such maps are trivial to compute. This was the representation the author (Stein) implemented in MAGMA.

The PARI documentation says the following (where we have preserved the incorrect typesetting):

> "A *character* on the Abelian group $\oplus(\mathbb{Z}/N_i\mathbb{Z})g_i$ is given by a row vector $\chi = [a_1,\ldots,a_n]$ such that $\chi(\prod g_i^{n_i}) = exp(2i\pi \sum a_i n_i/N_i)$."

This means that the abelian group has independent generators g_i of order N_i. This definition says that, e.g., the value of the character on g_1 is

$$\chi(g_1) = (e^{2\pi i/N_1})^{a_1}.$$

Thus the integers a_i are integers modulo N_i, and this representation is basically the same as the one we described in the previous paragraph (and which the author does not like).

4.8 Exercises

4.1 This exercise is about the structure of the units of $\mathbb{Z}/p^n\mathbb{Z}$.

(a) If p is odd and n is a positive integer, prove that $(\mathbb{Z}/p^n\mathbb{Z})^*$ is cyclic.

(b) If $n \geq 3$ prove that $(\mathbb{Z}/2^n\mathbb{Z})^*$ is a direct sum of the cylclic subgroups $\langle -1 \rangle$ and $\langle 5 \rangle$, of orders 2 and 2^{n-2}, respectively.

4.2 Prove that Algorithm 4.3.4 works, i.e., that if $g \in (\mathbb{Z}/p^r\mathbb{Z})^*$ and $g^{n/p_i} \neq 1$ for all $p_i \mid n = \varphi(n)$, then g is a generator of $(\mathbb{Z}/p^r\mathbb{Z})^*$.

4.3 Let p be an odd prime and $n \geq 2$ an integer, and prove that

$$(1 + p^{n-1}(\mathbb{Z}/p^n\mathbb{Z}),\, \times) \cong (\mathbb{Z}/p\mathbb{Z},\, +).$$

Use this to show that solving the discrete log problem in $(\mathbb{Z}/p^n\mathbb{Z})^*$ is "not much harder" than solving the discrete log problem in $(\mathbb{Z}/p\mathbb{Z})^*$.

4.4 Suppose ε is a nontrivial Dirichlet character modulo 2^n of order r over the complex numbers $\mathbb{C}$. Prove that the conductor of ε is

$$c = \begin{cases} 2^{\mathrm{ord}_2(r)+1} & \text{if } \varepsilon(5) = 1 \\ 2^{\mathrm{ord}_2(r)+2} & \text{if } \varepsilon(5) \neq 1. \end{cases}$$

4.5 (a) Find an irreducible quadratic polynomial f over $\mathbb{F}_5$.

(b) Then $\mathbb{F}_{25} = \mathbb{F}_5[x]/(f)$. Find an element with multiplicative order 5 in $\mathbb{F}_{25}$.

(c) Make a list of all Dirichlet characters in $D(25, \mathbb{F}_{25}, \zeta)$.

(d) Divide these characters up into orbits for the action of $\mathrm{Gal}(\overline{\mathbb{F}}_5/\mathbb{F}_5)$.

Chapter 5

Eisenstein Series

We introduce generalized Bernoulli numbers attached to Dirichlet characters, and give an algorithm to enumerate the Eisenstein series in $M_k(N, \varepsilon)$. We will wait until Chapter 8 for an algorithm to compute all cusp forms in $M_k(N, \varepsilon)$.

5.1 Generalized Bernoulli Numbers

Suppose ε is a Dirichlet character modulo N over $\mathbb{C}$.

Definition 5.1.1 (Generalized Bernoulli Number). Define the *generalized Bernoulli numbers $B_{k,\varepsilon}$* attached to ε by the following identity of infinite series:

$$\sum_{a=1}^{N-1} \frac{\varepsilon(a) \cdot x \cdot e^{ax}}{e^{Nx} - 1} = \sum_{k=0}^{\infty} B_{k,\varepsilon} \cdot \frac{x^k}{k!}.$$

If ε is the trivial character of modulus 1 and B_k are as in Section 2.1, then $B_{k,\varepsilon} = B_k$, except when $k = 1$, in which case $B_{1,\varepsilon} = -B_1 = 1/2$ (see Exercise 5.5).

Let $\mathbb{Q}(\varepsilon)$ denote the field generated by the values of the character ε, so $\mathbb{Q}(\varepsilon)$ is the cyclotomic extension $\mathbb{Q}(\zeta_n)$, where n is the order of ε.

Algorithm 5.1.2 (Bernoulli Numbers). *Given an integer $k \geq 0$ and any Dirichlet character ε with modulus N, this algorithm computes the generalized Bernoulli numbers $B_{j,\varepsilon}$, for $j \leq k$.*

1. Compute $g = x/(e^{Nx} - 1) \in \mathbb{Q}[[x]]$ to precision $O(x^{k+1})$ by computing $e^{Nx} - 1 = \sum_{n \geq 1} N^n x^n / n!$ to precision $O(x^{k+2})$, and computing the inverse $x/(e^{Nx} - 1)$. For completeness, note that if $f = a_0 + a_1 x + a_2 x^2 + \cdots$, then we have the following recursive formula for the coefficients b_n of the expansion of $1/f$:

$$b_n = -\frac{b_0}{a_0} \cdot (b_{n-1} a_1 + b_{n-2} a_2 + \cdots + b_0 a_n).$$

2. For each $a = 1, \ldots, N$, compute $f_a = g \cdot e^{ax} \in \mathbb{Q}[[x]]$, to precision $O(x^{k+1})$. This requires computing $e^{ax} = \sum_{n \geq 0} a^n x^n / n!$ to precision $O(x^{k+1})$. (One can omit computation of e^{Nx} if $N > 1$.)

3. Then for $j \leq k$, we have

$$B_{j,\varepsilon} = j! \cdot \sum_{a=1}^{N} \varepsilon(a) \cdot c_j(f_a),$$

where $c_j(f_a)$ is the coefficient of x^j in f_a.

Note that in Steps 1 and 2 we compute the power series doing arithmetic only in $\mathbb{Q}[[x]]$, not in $\mathbb{Q}(\varepsilon)[[x]]$, which could be much less efficient if ε has large order. One could also write down a recurrence formula for $B_{j,\varepsilon}$, but this would simply encode arithmetic in power series rings and the definitions in a formula.

Example 5.1.3. Let ε be the nontrivial character with modulus 4. Thus ε has order 2 and takes values in $\mathbb{Q}$. Then the Bernoulli numbers $B_{k,\varepsilon}$ for k even are all 0 and for k odd they are

$$\begin{aligned}
B_{1,\varepsilon} &= -1/2\\
B_{3,\varepsilon} &= 3/2\\
B_{5,\varepsilon} &= -25/2\\
B_{7,\varepsilon} &= 427/2\\
B_{9,\varepsilon} &= -12465/2\\
B_{11,\varepsilon} &= 555731/2\\
B_{13,\varepsilon} &= -35135945/2\\
B_{15,\varepsilon} &= 2990414715/2\\
B_{17,\varepsilon} &= -329655706465/2\\
B_{19,\varepsilon} &= 45692713833379/2.
\end{aligned}$$

These Bernoulli numbers can be divisible by large primes. For example, $B_{17,\varepsilon} = 5 \cdot 17^2 \cdot 228135437/2$.

Example 5.1.4. This examples illustrates that the generalized Bernoulli numbers need not be rational numbers. Suppose ε is the mod 5 character such that

$\varepsilon(2) = i = \sqrt{-1}$. Then $B_{k,\varepsilon} = 0$ for k even and

$$\begin{aligned}
B_{1,\varepsilon} &= \frac{-i-3}{5} \\
B_{3,\varepsilon} &= \frac{6i+12}{5} \\
B_{5,\varepsilon} &= \frac{-86i-148}{5} \\
B_{7,\varepsilon} &= \frac{2366i+3892}{5} \\
B_{9,\varepsilon} &= \frac{-108846i-176868}{5} \\
B_{11,\varepsilon} &= \frac{7599526i+12309572}{5} \\
B_{13,\varepsilon} &= \frac{-751182406i-1215768788}{5} \\
B_{15,\varepsilon} &= \frac{99909993486i+161668772052}{5} \\
B_{17,\varepsilon} &= \frac{-17209733596766i-27846408467908}{5}
\end{aligned}$$

Proposition 5.1.5. *If $\varepsilon(-1) \neq (-1)^k$, then $B_{k,\varepsilon} = 0$.*

5.2 Explicit Basis for the Eisenstein Subspace

Suppose χ and ψ are primitive Dirichlet characters with conductors L and M, respectively. Let

$$E_{k,\chi,\psi}(q) = c_0 + \sum_{m \geq 1} \left(\sum_{n \mid m} \psi(n) \cdot \chi(m/n) \cdot n^{k-1} \right) q^m \in \mathbb{Q}(\chi,\psi)[[q]], \quad (5.2.1)$$

where

$$c_0 = \begin{cases} 0 & \text{if } L > 1, \\ -\dfrac{B_{k,\psi}}{2k} & \text{if } L = 1. \end{cases}$$

Note that when $\chi = \psi = 1$ and $k \geq 4$, then $E_{k,\chi,\psi} = E_k$, where E_k is from Chapter 1.

Miyake proves statements that imply the following theorems in [Miy89, Ch. 7]. We will not prove them in this book since developing the theory needed to prove them would take us far afield from our goal, which is to compute $M_k(N, \varepsilon)$.

Theorem 5.2.1. *Suppose t is a positive integer and χ, ψ are as above, and that k is a positive integer such that $\chi(-1)\psi(-1) = (-1)^k$. Except when*

$k = 2$ and $\chi = \psi = 1$, the power series $E_{k,\chi,\psi}(q^t)$ defines an element of $M_k(MLt, \chi/\psi)$. If $\chi = \psi = 1$, $k = 2$, $t > 1$, and $E_2 = E_{k,\chi,\psi}$, then $E_2(q) - tE_2(q^t)$ is a modular form in $M_2(\Gamma_0(t))$.

Theorem 5.2.2. *The Eisenstein series in $M_k(N, \varepsilon)$ coming from Theorem 5.2.1 form a basis for the Eisenstein subspace $E_k(N, \varepsilon)$.*

Theorem 5.2.3. *The Eisenstein series $E_{k,\chi,\psi}(q) \in M_k(ML)$ defined above is an eigenvector for all Hecke operators T_n. Also $E_2(q) - tE_2(q^t)$, for $t > 1$, is an eigenform.*

Since $E_{k,\chi,\psi}(q)$ is normalizes so the coefficient of q is 1, the eigenvalue of T_m is

$$\sum_{n|m} \psi(n) \cdot \chi(m/n) \cdot n^{k-1}.$$

Also for $f = E_2(q) - tE_2(q^t)$ with $t > 1$ prime, the coefficient of q is 1, and $T_m(f) = \sigma_1(m) \cdot f$ for $(m, t) = 1$, and $T_t(f) = ((t + 1) - t)f = f$.

Algorithm 5.2.4 (Enumerating Eisenstein Series). *Given a weight k and a Dirichlet character ε of modulus N, this algorithm computes a basis for the Eisenstein subspace $E_k(N, \varepsilon)$ of $M_k(N, \varepsilon)$ to precision $O(q^r)$.*

1. [Weight 2 Trivial Character?] If $k = 2$ and $\varepsilon = 1$, output the Eisenstein series $E_2(q) - tE_2(q^t)$, for each divisor $t \mid N$ with $t \neq 1$, then terminate.
2. [Compute Dirichlet Group] Let $G = D(N, \mathbb{Q}(\zeta_n))$ be the group of Dirichlet characters with values in $\mathbb{Q}(\zeta_n)$, where n is the exponent fo $(\mathbb{Z}/N\mathbb{Z})^*$.
3. [Compute Conductors] Compute the conductor of every element of G (which just involves computing the orders of the local components of each character).
4. [List Characters χ] Form a list V all Dirichlet characters $\chi \in G$ such that $\mathrm{cond}(\chi) \cdot \mathrm{cond}(\chi/\varepsilon)$ divides N.
5. [Compute Eisenstein Series] For each character χ in V, let $\psi = \chi/\varepsilon$, and compute $E_{k,\chi,\psi}(q^t) \pmod{q^r}$ for each divisor t of $N/(\mathrm{cond}(\chi) \cdot \mathrm{cond}(\psi))$. We compute $E_{k,\chi,\psi}(q^t) \pmod{q^r}$ using (5.2.1) and Algorithm 5.1.2.

Remark 5.2.5. Algorithm 5.2.4 is what I currently use in my programs. It might be better to first reduce to the prime power case by writing all characters as product of local characters and combine Steps 3 and 4 into a single step that involves orders. However, this might make things more complicated and obscure.

Example 5.2.6. The following is a basis of Eisenstein series $E_{2,\chi,\psi}$ for $E_2(\Gamma_1(13))$.

```
f1 = 1/2 + q + 3*q^2 + 4*q^3 + O(q^4)

f2 = (-7/13*zeta_12^2 - 11/13) + q + (2*zeta_12^2 + 1)*q^2
                               + (-3*zeta_12^2 + 1)*q^3 + O(q^4)

f3 = q + (zeta_12^2 + 2)*q^2 + (-1*zeta_12^2 + 3)*q^3 + O(q^4)

f4 = (-1*zeta_12^2) + q + (2*zeta_12^2 - 1)*q^2
                        + (3*zeta_12^2 - 2)*q^3 + O(q^4)

f5 = q + (zeta_12^2 + 1)*q^2 + (zeta_12^2 + 2)*q^3 + O(q^4)

f6 = (-1) + q + (-1)*q^2 + 4*q^3 + O(q^4)

f7 = q + q^2 + 4*q^3 + O(q^4)

f8 = (zeta_12^2 - 1) + q + (-2*zeta_12^2 + 1)*q^2
                         + (-3*zeta_12^2 + 1)*q^3 + O(q^4)

f9 = q + (-1*zeta_12^2 + 2)*q^2 + (-1*zeta_12^2 + 3)*q^3 + O(q^4)

f10 = (7/13*zeta_12^2 - 18/13) + q + (-2*zeta_12^2 + 3)*q^2
                                   + (3*zeta_12^2 - 2)*q^3 + O(q^4)

f11 = q + (-1*zeta_12^2 + 3)*q^2 + (zeta_12^2 + 2)*q^3 + O(q^4)
```

5.3 Exercises

5.1 Suppose $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ and N is a positive integer. Prove that there is a positive integer h such that $\left(\begin{smallmatrix} 1 & h \\ 0 & 1 \end{smallmatrix}\right) \in \gamma^{-1}\Gamma_1(N)\gamma$.

5.2 Prove that the map $\mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ is surjective. (Hint: There is a proof of a more general result near the beginning of Shimura's book [Shi94].)

5.3 Prove that $M_k(N, 1) = M_k(\Gamma_0(N))$.

5.4 Suppose A and B are diagonalizable linear transformations of a finite-dimensional vector space V and that both A and B are diagonalizable. Prove there is a basis for V so that the matrices of A and B with respect to that both are simultaneously diagonal.

5.5 If ε is the trivial character of modulus 1 and B_k are as in Section 2.1, then $B_{k,\varepsilon} = B_k$, except when $k = 1$, in which case $B_{1,\varepsilon} = -B_1 = 1/2$.

5.6 Prove that if $n > 1$ is odd, then the Bernoulli number B_n is 0.

Chapter 6

Dimensions Formulas

When computing with spaces of modular forms, it is helpful to have easy-to-compute formulas for dimensions of these spaces, and certain of their subspaces. For example, they provide a double-check on the output of the algorithms from Chapter 8 that compute explicit bases for spaces of modular forms. Alternatively, dimension formulas can be used to improve the efficiency of some of the algorithms in Chapter 8, since we can use them to determine the ranks of certain matrices without having to explicitly compute them. If we know the dimension of $M_k(N,\varepsilon)$, and we have a process for computing q-expansions of elements of $M_k(N,\varepsilon)$, e.g., multiplying together q-expansions of certain forms of smaller weight or searching for θ-series attached to quadratic forms, then we can tell when we are done generating $M_k(N,\varepsilon)$.

This chapter contains formulas the author knows for computing dimensions of spaces of modular forms, along with some hints about how to compute them, when this isn't obvious. In several cases we give dimension formulas for spaces that haven't yet been defined in this book, so we define them in this chapter (e.g., we will discuss newforms and oldforms further). We also give many examples, which were computed using the modular symbols algorithms from Chapter 8.

Many of the dimension formulas and algorithms we give below grew out of a program that Bruce Caskel wrote (around 1996) in PARI, which Kevin Buzzard extended. Their program codified dimension formulas that Buzzard and Caskel found or extracted from the literature (mainly [Shi94, §2.6]). The algorithm for dimensions of spaces with nontrivial character are from [CO77], with some slight refinements from Kevin Buzzard.

For the rest of this chapter, N denotes a positive integer and $k \geq 2$ is an integer. We give **no formulas** for dimensions of spaces of weight 1 modular forms, because it is an *open problem* to give such formulas; the geometric methods used to derive the formulas below do not apply in the case $k = 1$. If $k = 0$, the only modular forms are the constants, and for $k < 0$ the dimension of $M_k(N,\varepsilon)$ is 0.

For a nonzero integer N and a prime p, let $v_p(N)$ be the largest e such that $p^e \mid N$. In the formulas below, p always denotes a prime number. Let $M_k(N,\varepsilon)$ be the space of modular forms of level N weight k and character ε, and $S_k(N,\varepsilon)$

and $E_k(N,\varepsilon)$ the cuspidal and Eisenstein subspaces.

The dimension formulas below for $S_k(\Gamma_0(N))$, $S_k(\Gamma_1(N))$, $E_k(\Gamma_0(N))$ and $E_k(\Gamma_1(N))$ below are almost straight from [Shi94, §2.6] (see also [Miy89, §2.5]), and they are derived using the Riemann-Roch Theorem applied to the covering $X_0(N) \to X_0(1)$ or $X_1(N) \to X_1(1)$ and appropriately chosen divisors. It would be natural to give a sample argument along these lines at this point, but I will not since it easy to find such arguments in other books and survey papers (see, e.g., [DI95]). So you will not learn much about how to derive dimension formulas from this chapter. What you will learn is what is known about dimension formulas and what some of the obscure references are.

6.1 Modular Forms for $\Gamma_0(N)$

Define functions of a positive integer N by the following formulas:

$$\mu_0(N) = \prod_{p\mid N}\left(p^{v_p(N)} + p^{v_p(N)-1}\right)$$

$$\mu_{0,2}(N) = \begin{cases} 0 & \text{if } 4 \mid N, \\ \prod_{p\mid N}\left(1+\left(\frac{-4}{p}\right)\right) & \text{otherwise.}\end{cases}$$

$$\mu_{0,3}(N) = \begin{cases} 0 & \text{if } 2 \mid N \text{ or } 9 \mid N, \\ \prod_{p\mid N}\left(1+\left(\frac{-3}{p}\right)\right) & \text{otherwise.}\end{cases}$$

$$c_0(N) = \sum_{d\mid N}\varphi(\gcd(d, N/d))$$

$$g_0(N) = 1 + \frac{\mu_0(N)}{12} - \frac{\mu_{0,2}(N)}{4} - \frac{\mu_{0,3}(N)}{3} - \frac{c_0(N)}{2}$$

Note that $\mu_0(N)$ is the index of $\Gamma_0(N)$ in $\mathrm{SL}_2(\mathbb{Z})$. Also $g_0(N)$ is the genus of the modular curve $X_0(N)$, and $c_0(N)$ is the number of cusps of $X_0(N)$.

Proposition 6.1.1. *We have* $\dim S_2(\Gamma_0(N)) = g_0(N)$, *and for* $k \geq 4$ *even,*

$$\dim S_k(\Gamma_0(N)) = (k-1)\cdot(g_0(N)-1) \;+\; \left(\frac{k}{2}-1\right)\cdot c_0(N) \;+\; \mu_{0,2}(N)\cdot\left\lfloor\frac{k}{4}\right\rfloor \;+\; \mu_{0,3}(N)\cdot\left\lfloor\frac{k}{3}\right\rfloor.$$

The dimension of the Eisenstein subspace is as follows:

$$\dim E_k(\Gamma_0(N)) = \begin{cases} c_0(N) & \text{if } k \neq 2, \\ c_0(N)-1 & \text{if } k = 2.\end{cases}$$

The following table contains the dimension of $S_k(\Gamma_0(N))$ for some sample values of N and k:

N	$\dim S_2(\Gamma_0(N))$	$\dim S_4(\Gamma_0(N))$	$\dim S_6(\Gamma_0(N))$	$\dim S_{24}(\Gamma_0(N))$
1	0	0	0	2
10	0	3	5	33
11	1	2	4	22
100	7	36	66	336
389	32	97	161	747
1000	131	430	730	3430
2004	331	1002	1674	7722
100000	14801	44800	74800	344800

6.1.1 New and Old Subspaces

For each divisor N' of N, there are natural maps

$$\alpha_d : M_k(\Gamma_0(N')) \to M_k(\Gamma_0(N)),$$

corresponding to the divisors d of N/N', and maps

$$\beta_d : M_k(\Gamma_0(N)) \to M_k(\Gamma_0(N')).$$

such that $\beta_d \circ \alpha_d$ is multiplication by a nonzero scalar. On q-expansions, $\alpha_d(f(q)) = f(q^d)$, and the definition of β_d is a more complicated "trace map" (see, e.g., [Lan95]).

The space $M_k(\Gamma_0(N))$ decomposes as a direct sum

$$M_k(\Gamma_0(N)) = M_k(\Gamma_0(N))^{\mathrm{old}} \oplus M_k(\Gamma_0(N))^{\mathrm{new}},$$

where $M_k(\Gamma_0(N))^{\mathrm{old}}$ is the subspace generated by all images $\alpha_d(M_k(\Gamma_0(N'))$ where N' runs through proper divisors of N and d runs through all divisors of N/N'. The new subspace $M_k(\Gamma_0(N))^{\mathrm{new}}$ can be defined as either the intersection of the kernels of all maps β_d to lower level, or the largest Hecke-stable complement of $M_k(\Gamma_0(N))^{\mathrm{old}}$.

Atkin and Lehner [AL70] proved that the space $S_k(\Gamma_0(N))$ is built out of new subspaces, in the following sense.

Theorem 6.1.2 (Atkin-Lehner). *We have an isomorphism*

$$S_k(\Gamma_0(N)) = \sum_{M|N} \sum_{d|N/M} \alpha_d(S_k(\Gamma_0(M))^{\mathrm{new}}).$$

This is an isomorphism of $\mathbb{T}'$ modules, where $\mathbb{T}'$ is the anemic Hecke algebra, i.e., the subring generated by Hecke operators T_n with $\gcd(n, N) = 1$.

This theorem reduces the problem of computing $S_k(\Gamma_0(N))$ to that of computing $S_k(\Gamma_0(M))^{\mathrm{new}}$ for divisors M of N, a fact that will be central later in this book. Atkin and Lehner also prove that one can completely determine $S_k(\Gamma_0(M))^{\mathrm{new}}$ just from the information of how the Hecke operators act on it (their "multiplicity one" theory). Atkin and Lehner's work was generalized to fairly arbitrary congruence subgroups of $\mathrm{SL}_2(\mathbb{Z})$ by Winnie Li in her Berkeley Ph.D. thesis under A. Ogg (see [Li75]).

If $N'' \mid N' \mid N$, then the maps α_d from $M_k(\Gamma_0(N''))$ to $M_k(\Gamma_0(N))$ factor through $M_k(\Gamma_0(N'))$. Thus in the definition of $M_k(\Gamma_0(N))^{\mathrm{old}}$ and $M_k(\Gamma_0(N))^{\mathrm{new}}$, it would suffice to consider only proper divisors N' of N such that N/N' is prime.

Warning: For a fixed $N' = N/p$, the images of α_1 and α_p need *not* always be linearly independent (see Example 6.1.4 below). However, the images of the new subspace $S_k(\Gamma_0(N'))^{\mathrm{new}}$ are linearly independent, as asserted by Theorem 6.1.2.

Proposition 6.1.3. *The dimension of the new subspace is*

$$\dim S_k(\Gamma_0(N))^{\mathrm{new}} = \sum_{M|N} \overline{\mu}(N/M) \cdot \dim S_k(\Gamma_0(M)),$$

where the sum is over the positive divisors of N, and for an integer R,

$$\overline{\mu}(R) = \begin{cases} 0 & \text{if } p^3 \mid R \text{ for some } p \\ \prod_{p||R} -2 & \text{otherwise,} \end{cases}$$

where the product is over primes that exactly divide n. (Note that $\overline{\mu}$ is not *the Moebius function, but is similar to it.)*

Let $f(n) = \dim S_k(\Gamma_0(n))$ and $g(n) = \dim S_k(\Gamma_0(n))^{\mathrm{new}}$. Theorem 6.1.2 implies that

$$f(N) = \sum_{M|N} \sigma_0(N/M) g(M), \tag{6.1.1}$$

where $\sigma_0(N/M)$ is the number of divisors of N/M. Presumably there is an analogue of Moebius inversion, but for functions with the property in (6.1.1), which involves the function $\overline{\mu}$.

Example 6.1.4. The space $M_2(\Gamma_0(45))$ has dimension 10 and basis

```
1 + 12*q^15 + O(q^20),
q + q^7 + 3*q^16 + 6*q^19 + O(q^20),
q^2 + 4*q^11 + 3*q^14 + q^17 + O(q^20),
q^3 + q^12 + q^15 + 3*q^18 + O(q^20),
q^4 + q^7 + 2*q^13 + 4*q^16 + 2*q^19 + O(q^20),
q^5 + O(q^20),
q^6 + 2*q^12 + 2*q^15 - q^18 + O(q^20),
q^8 + q^14 + q^17 + O(q^20),
q^9 - 2*q^15 + 3*q^18 + O(q^20),
q^10 + O(q^20)
```

The new subspace is spanned by the single cusp form

```
q + q^2 - q^4 - q^5 - 3*q^8 - q^10 + 4*q^11 - 2*q^13 + O(q^14)
```

First consider $N' = 45/3 = 15$. The space $M_2(\Gamma_0(15))$ has basis

```
1 + 12*q^5 + O(q^8),
q + q^4 + q^5 + 3*q^6 + 2*q^7 + O(q^8),
q^2 + 2*q^4 + 2*q^5 - q^6 + 2*q^7 + O(q^8),
q^3 - 2*q^5 + 3*q^6 + O(q^8)
```

There are two maps α_1 and α_3 from $M_2(\Gamma_0(15))$ to $M_2(\Gamma_0(45))$. The one dimension space $M_2(\Gamma_0(5))$ embeds in $M_2(\Gamma_0(15))$ via $f(q) \mapsto f(q)$ and $f(q) \mapsto f(q^3)$. We have a commutative diagram

$$\begin{tikzcd}
 & M_2(\Gamma_0(15)) \arrow[dr, "\alpha_3"] & \\
M_2(\Gamma_0(5)) \arrow[ur, "\alpha_1"] \arrow[dr, "\alpha_3"'] & & M_2(\Gamma_0(45)). \\
 & M_2(\Gamma_0(15)) \arrow[ur, "\alpha_1"'] &
\end{tikzcd}$$

This diagram illustrates that the intersection of the two images of $M_2(\Gamma_0(15))$ has dimension at least 1. In fact, the sum of the images of the two maps from $M_2(\Gamma_0(15))$ is a 7-dimensional subspace of $M_2(\Gamma_0(45))$.

Next consider $N' = 45/5 = 9$, where the space $M_2(\Gamma_0(9)) = E_2(\Gamma_0(9))$ has as basis the three forms

```
1 + 12*q^3 + 36*q^6 + O(q^8),
q + 7*q^4 + 8*q^7 + O(q^8),
q^2 + 2*q^5 + O(q^8)
```

There are two maps α_1 and α_5 from $M_2(\Gamma_0(9))$ to $M_2(\Gamma_0(45))$. The images of these two maps span a space of dimension 6, and this space intersects the span of the images of $M_2(\Gamma_0(15))$ in a space of dimension 4. Thus the old subspace $M_2(\Gamma_0(45))^{\text{old}}$ has dimension 9, and the new subspace has dimension 1. The new subspace is spanned by the single cusp form

```
q + q^2 - q^4 - q^5 - 3*q^8 - q^10 + 4*q^11 + O(q^12)
```

Remark 6.1.5. Csirik, Wetherell, and Zieve prove in [CWZ01] that a random positive integer has probability 0 of being a value of $g_0(N) = \dim S_2(\Gamma_0(N))$, and give bounds on the size of the set of values of $g_0(N)$ below some given x. For example, they show that $150, 180, 210, 286, 304, 312, \ldots$ are the first few integers that are not of the form $g_0(N)$ for some N.

6.2 Modular Forms for $\Gamma_1(N)$

This section follows Section 6.1 closely, but with suitable modifications with $\Gamma_0(N)$ replaced by $\Gamma_1(N)$. The notion of new and old subspaces for $\Gamma_1(N)$ is exactly the same as for $\Gamma_0(N)$; simply replace $\Gamma_0(N)$ by $\Gamma_1(N)$ in the discussion of new and old forms in Section 6.1.

Define functions of a positive integer N by the following formulas:

$$\mu_1(N) = \begin{cases} \mu_0(N) & \text{if } N = 1, 2, \\ \dfrac{\phi(N)\cdot\mu_0(N)}{2} & \text{otherwise.} \end{cases}$$

$$\mu_{1,2}(N) = \begin{cases} 0 & \text{if } N \geq 4, \\ \mu_{0,2}(N) & \text{otherwise.} \end{cases}$$

$$\mu_{1,3}(N) = \begin{cases} 0 & \text{if } N \geq 4, \\ \mu_{0,3}(N) & \text{otherwise.} \end{cases}$$

$$c_1(N) = \begin{cases} c_0(N) & \text{if } N = 1, 2, \\ 3 & \text{if } N = 4, \\ \displaystyle\sum_{d\mid N} \frac{\phi(d)\phi(N/d)}{2} & \text{otherwise.} \end{cases}$$

$$g_1(N) = 1 + \frac{\mu_1(N)}{12} - \frac{\mu_{1,2}(N)}{4} - \frac{\mu_{1,3}(N)}{3} - \frac{c_1(N)}{2}$$

Note that $g_1(N)$ is the genus of the modular curve $X_1(N)$, and $c_1(N)$ is the number of cusps of $X_1(N)$. [[**TODO: Make sure this is right for $N \leq 5$.**]]

Proposition 6.2.1. *We have* $\dim S_2(\Gamma_1(N)) = g_1(N)$. *If* $N \leq 2$, *then*

$$\dim S_k(\Gamma_1(N)) = \dim S_k(\Gamma_0(N)),$$

where $\dim S_k(\Gamma_0(N))$ *is given by the formula of Proposition 6.1.1. If* $k \geq 3$, *let*

$$a = (k-1)(g_1(N)-1) + \left(\frac{k}{2} - 1\right)\cdot c_1(N).$$

Then for $N \geq 3$,

$$\dim S_k(\Gamma_1(N)) = \begin{cases} a + 1/2 & \textit{if } N = 4 \textit{ and } 2 \nmid k, \\ a + \lfloor k/3 \rfloor & \textit{if } N = 3, \\ a & \textit{otherwise.} \end{cases}$$

The dimension of the Eisenstein subspace is as follows:

$$\dim E_k(\Gamma_1(N)) = \begin{cases} c_1(N) & \textit{if } k \neq 2, \\ c_1(N) - 1 & \textit{if } k = 2. \end{cases}$$

The dimension of the new subspace of $M_k(\Gamma_1(N))$ is

$$\dim S_k(\Gamma_1(N))^{\text{new}} = \sum_{M\mid N} \overline{\mu}(N/M)\cdot \dim S_k(\Gamma_1(M)),$$

where $\overline{\mu}$ is as in the statement of Proposition 6.1.3.

Remark 6.2.2. Since $M_k = S_k \oplus E_k$, the formulas above also give a formula for the dimension of M_k.

The following table contains the dimension of $S_k(\Gamma_1(N))$ for some sample values of N and k:

N	$\dim S_2(\Gamma_1(N))$	$\dim S_3(\Gamma_1(N))$	$\dim S_4(\Gamma_1(N))$	$\dim S_{24}(\Gamma_1(N))$
1	0	0	0	2
10	0	2	5	65
11	1	5	10	110
100	231	530	830	6830
389	6112	12416	18721	144821
1000	28921	58920	88920	688920
2004	109893	221444	332996	2564036
100000	299792001	599792000	899792000	6899792000

6.3 Modular Forms with Character

Fix a Dirichlet character ε modulo N, and let c be the conductor of ε (we do *not* assume that ε is primitive). Assume that $\varepsilon \neq 1$, since otherwise $M_k(N,\varepsilon) = M_k(\Gamma_0(N))$ and the formulas of Section 6.1 apply. Also, assume that $\varepsilon(-1) = (-1)^k$, since otherwise $\dim M_k(\Gamma_0(N)) = 0$. In this section we discuss formulas for certain subspaces of $M_k(N,\varepsilon)$.

In [CO77], Cohen and Oesterle assert (without proof, see Remark 6.3.2 below) that for any $k \in \mathbb{Z}$ and N, ε as above, that

$$\begin{aligned}\dim S_k(N,\varepsilon) &- \dim M_{2-k}(N,\varepsilon)\\ &= \frac{k-1}{12}\cdot \mu_0(N) \;-\; \frac{1}{2}\cdot \prod_{p\mid N} \lambda(p,N,v_p(c))\\ &\quad + \gamma_4(k)\cdot \sum_{x\in A_4(N)} \varepsilon(x) \;+\; \gamma_3(k)\cdot \sum_{x\in A_3(N)} \varepsilon(x)\end{aligned}$$

where $\mu_0(N)$ is as in Section 6.1, $A_4(N) = \{x \in \mathbb{Z}/N\mathbb{Z} : x^2+1 = 0\}$ and

$A_3(N) = \{x \in \mathbb{Z}/N\mathbb{Z} : x^2 + x + 1 = 0\}$, and γ_3, γ_4 are:

$$\gamma_4(k) = \begin{cases} -1/4 & \text{if } k \equiv 2 \pmod 4 \\ 1/4 & \text{if } k \equiv 0 \pmod 4 \\ 0 & \text{if } k \text{ is odd} \end{cases}$$

$$\gamma_3(k) = \begin{cases} -1/3 & \text{if } k \equiv 2 \pmod 3 \\ 1/3 & \text{if } k \equiv 0 \pmod 3 \\ 0 & \text{if } k \equiv 1 \pmod 3 \end{cases}$$

It remains to define λ. Fix a prime divisor $p \mid N$ and let $r = v_p(N)$. Then

$$\lambda(p, N, v_p(c)) = \begin{cases} p^{\frac{r}{2}} + p^{\frac{r}{2}-1} & \text{if } 2 \cdot v_p(c) \leq r \text{ and } 2 \mid r, \\ 2 \cdot p^{\frac{r-1}{2}} & \text{if } 2 \cdot v_p(c) \leq r \text{ and } 2 \nmid r, \\ 2 \cdot p^{r - v_p(c)} & \text{if } 2 \cdot v_p(c) > r \end{cases}$$

The formula can be used to compute $\dim M_k(N, \varepsilon)$, $\dim S_k(N, \varepsilon)$, and $\dim E_k(N, \varepsilon)$ for any N, ε, $k \neq 1$, by using that

$$\begin{aligned} \dim S_k(N, \varepsilon) &= 0 && \text{if } k \leq 0 \\ \dim M_k(N, \varepsilon) &= 0 && \text{if } k < 0 \\ \dim M_0(N, \varepsilon) &= 1 && \text{if } k = 0 \end{aligned}$$

One thing that is not straightforward when implementing an algorithm to compute the above dimension formulas is how to efficiently compute the sets $A_4(N)$ and $A_6(N)$. Kevin Buzzard suggested the following two algorithms to the author. Note that if k is odd, then $\gamma_4(k) = 0$, so the sum over $A_4(N)$ is only needed when k is even.

Algorithm 6.3.1 (Compute Sum over $A_4(N)$). *INPUT: A positive integer N and an even Dirichlet character ε modulo N.*
OUTPUT: The sum $\sum_{x \in A_4(N)} \varepsilon(x)$.

1. [Factor N] Compute the prime factorization $p_1^{e_1} \cdots p_n^{e_n}$ of N.
2. [Initialize] Set $t = 1$ and $i = 0$.
3. [Loop over prime divisors] Set $i = i + 1$. If $i > n$, return t. Otherwise set $p = p_i$ and $e = e_i$.
   (a) If $p \equiv 3 \pmod 4$, return 0.
   (b) If $p = 2$ and $e > 1$, return 0.
   (c) If $p = 2$ and $e = 1$, go to Step 3.
   (d) Compute a generator $a \in (\mathbb{Z}/p\mathbb{Z})^*$ using Algorithm 4.3.4.
   (e) Compute $\omega = a^{(p-1)/4}$.
   (f) Using the Chinese Remainder Theorem to find $x \in \mathbb{Z}/N\mathbb{Z}$ such that $x \equiv a \pmod p$ and $x \equiv 1 \pmod{N/p^e}$.

(g) Set $x = x^{p^{r-1}}$.
(h) Set $s = \varepsilon(x)$.
(i) If $s = 1$, set $t = 2t$ and go to Step 3.
(j) If $s = -1$, set $t = -2t$ and go to Step 3.

Proof. Note that $\varepsilon(-x) = \varepsilon(x)$, since ε is even. By the chinese remainder theorem, the set $A_4(N)$ is empty if and only if there is no square root of -1 modulo some prime power divisor of p. If $A_4(N)$ is empty, the algorithm correctly detects this fact in steps 3a–3b. Thus assume $A_4(N)$ is non-empty. For each prime power $p_i^{e_i}$ that exactly divides N, let $x_i \in Z/N\mathbb{Z}$ be such that $x_i^2 = -1$ and $x_i \equiv 1 \pmod{p_j^{e_j}}$ for $i \neq j$. This is the value of x computed in steps 3d–3g (as one can see using elementary number theory).

The next key observation is that

$$\prod_i (\varepsilon(x_i) + \varepsilon(-x_i)) = \sum_{x \in A_4(N)} \varepsilon(x), \tag{6.3.1}$$

since by the chinese remainder theorem the elements of $A_4(N)$ are in bijection with the choices for a square root of -1 modulo each prime power divisors of N. The observation (6.3.1) is a huge gain from an efficiency point of view—if N had r prime factors, then $A_4(N)$ would have size 2^r, which could be prohibitive, where the product involves only r factors. To finish the proof, just note that Steps 3h–3j compute the local factors $\varepsilon(x_i) + \varepsilon(-x_i) = 2\varepsilon(x_i)$, where again we use that ε is even. (Note, e.g., that a solution of $x^2 + 1 \equiv 0 \pmod{p}$ lifts uniquely to a solution mod p^n for any n, because the kernel of the natural homomorphism $(\mathbb{Z}/p^n\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^*$ is a group of p-power order. $\square$

The algorithm for computing the sum over $A_3(N)$ is similar, but we omit it.

The following table contains the dimension of $S_k(N, \varepsilon)$ for some sample values of N and k. In each case, ε is the product of characters ε_p of maximal order corresponding to the prime power factors of N (i.e., the product of the generators of $D(N, \mathbb{C}^*)$).

N	$\dim S_2(N,\varepsilon)$	$\dim S_3(N,\varepsilon)$	$\dim S_4(N,\varepsilon)$	$\dim S_{24}(N,\varepsilon)$
1	0	0	0	2
10	0	1	0	0
11	0	1	0	0
100	13	0	43	343
389	0	64	0	0
1000	148	0	448	3448
2004	0	668	0	0

Remark 6.3.2. Cohen and Oesterle also give dimension formulas for spaces of half-integral weight modular forms, which we do not give in this chapter. Also [CO77] does not contain any *proofs* that their claimed formulas are correct, but instead say only that "Les formules qui les donnent sont connues de beaucoup de gens et il existe plusieurs méthodes permettant de les obtenir (théorème

de Riemann-Roch, application des formules de trace données par Shimura)." (The formulas that we give here are well known and there exist many methods to prove them, e.g., the Riemann-Roch theorem and applications of the trace formula of Shimura.)

6.4 Exercises

6.1 Fill in the elementary number theory details of the proof of Algorithm 6.3.1.

6.2 Track this down the analogue of Moebius inversion for $\overline{\mu}$ and give a quick presentation on it.

6.3 Implement in your favorite computer language an algorithm to compute $\dim S_k(\Gamma_0(N))$.

Chapter 7

Linear Algebra

This chapter is about exact matrix algebra with over the rational numbers and cyclotomic fields. Algorithms for linear algebra over exact fields are necessary in order to implement the modular symbols algorithms that we will describe in Chapter 7.

This chapter partly overlaps with [Coh93, §2.1–2.4].

7.1 Echelon Forms of Matrices

Definition 7.1.1 (Reduced Row Echelon Form). A matrix is in *row echelon form* if each row in the matrix starts with more zeros than the row above it. A matrix is in *reduced row echelon form* if it is in row echelon form, the first nonzero entry of any row is 1, and the first nonzero entry of any row is the only nonzero value in its column.

Given a matrix A, there is another matrix B such that B is obtained from A by left multiplication by an invertible matrix and B is in reduced row echelon form. This matrix B is called the reduced row echelon form of A. It is unique.

A *pivot column* of A is one such that the reduced row echelon form of A contains a leading 1.

Example 7.1.2. The following matrix is in row echelon form, but not reduced row echelon form:

```
[ 14,  2,  7,  228, -224;
   0,  0,  3,   78,  -70;
   0,  0,  0, -405,  381]
```

The reduced row echelon form of the above matrix is

```
[1, 1/7, 0, 0, -1174/945;
 0,   0, 1, 0,   152/135;
 0,   0, 0, 1,  -127/135]
```

Notice that the entries of the reduced row echelon form can easily be messy. Another example is the simple looking matrix

```
[ -9,  6,  7, 3, 1, 0, 0, 0;
 -10,  3,  8, 2, 0, 1, 0, 0;
   3, -6,  2, 8, 0, 0, 1, 0;
  -8, -6, -8, 6, 0, 0, 0, 1]
```

whose echelon form is

```
[1, 0, 0, 0,  42/1025,  -92/1025,  1/25,  -9/205;
 0, 1, 0, 0, 716/3075, -641/3075, -2/75,  -7/615;
 0, 0, 1, 0, -83/1025,  133/1025,  1/25, -23/410;
 0, 0, 0, 1, 184/1025, -159/1025,  2/25,   9/410]
```

One learns in a basic linear algebra course that two matrices A and B have the same reduced row echelon form if and only if there is an invertible matrix E such that $EA = B$. Also, many standard operations in linear algebra, e.g., computation of the kernel of a linear map, intersection of subspaces, membership checking, etc., can be encoded as a question about computing the echelon form of a matrix.

The following is a naive algorithm for computing the echelon form of a matrix.

Algorithm 7.1.3 (Gauss Elimination). *INPUT: An $m \times n$ matrix A over a field.*
OUTPUT: The reduced row echelon form of A.
We write `A[i,j]` *for the i, j entry of A, where $0 \le i \le m-1$ and $0 \le j \le n-1$.*

```
def echelon(A):
    start_row = 0
    nr = A.nrows        # The number of rows of A
    nc = A.ncols        # The number of columns of A
    for c in range(nc): # for c = 0, 1, 2, ..., nc-1
        for r in range(nr):
            a = A[r,c]
            # if a is nonzero
            if a != 0:
                # Rescale row r of A by 1/a.
                A.scale_row(r, 1/a)
                # Swap row r with the start_row row.
                A.swap_rows(r, start_row)
                # Clear the c-th column
                for i in range(nr):
                    if i != start_row:
                        if A[i,c] != 0:
                            # Add -A[i,c] times start_row to the i-th row
                            # in order to clear the leading entry of
                            # the i-th row.
                            A.add_multiple_of_row(start_row, -A[i,c], i)
                # Increment the start_row
                start_row = start_row + 1
                # The following break means that we skip the rest
                # of the for loop over r in range(nr), and
                # increase c and start a new for loop over r.
                break
```

This algorithm takes $O(mn^2)$ arithmetic operation in the base field, where A is an $m \times n$ matrix. If the base field is $\mathbb{Q}$, the entries can become huge and arithmetic operations can be increasingly expensive. See Section 7.2 for ways to mitigate this problem.

To conclude this section we mention how to convert a few standard problems into questions about reduced row echelon forms of matrices. Note that one can also phrase some of these answers in terms of the echelon form, which might be easier to compute, or an LUP decomposition (lower triangular times upper triangular times permutation matrix), which the numerical analysts use.

1. **Kernel of A:** Since passing to the reduced row echelon form of A is the same as multiplying on the left by an invertible matrix, the kernel of the reduce row echelon form is the same as the kernel of A. Thus we may assume A is in reduced row echelon form. There is a basis vector of $\ker(A)$ that corresponds to each non-pivot column of A. That vector has a 1 at the non-pivot column, 0's at all other non-pivot columns, and for each pivot column, the negative of the entry of A at the non-pivot column in the row with that pivot element.

2. **Intersection of Subspaces:** Suppose W_1 and W_2 are subspace of a finite-dimensional vector space V. Let A_1 and A_2 be matrices whose columns form a basis for W_1 and W_2, respectively. Let $A = [A_1|A_2]$ be the augmented matrix formed from A_1 and A_2. Let K be the kernel of the linear transformation defined by A. Then K is isomorphic to the desired intersection. To write down the intersection explicitly, suppose that $\dim(V) \leq \dim(W)$ and do the following: For each b in a basis for K, write down the linear combination of a basis for V got by taking the first $\dim(V)$ entries of the vector b. The fact that b is in $\mathrm{Ker}(A)$ implies that the vector we just wrote down is also in W. We took V to have smaller dimension just so that the linear combinations in the intersection could be written down slightly more quickly.

7.2 Echelon Forms over $\mathbb{Q}$

A major difficulty with computation of the echelon form of a dense matrix over the rational numbers is that arithmetic with large rational numbers is very time consuming, since each addition potentially requires a gcd and numerous additions and multiplications of integers. Moreover, the entries of A during intermediate steps of Algorithm 7.1.3 can be huge even though the entries of A and the answer are small. For example, suppose A is an invertible square matrix. Then the echelon form of A is the identity matrix, but during intermediate steps the entries of A could be quite large. One technique for mitigating this problem is to compute the echelon form using a multi-modular method. The following is a sketch of such a multi-modular method (we will give a more precise version; see Algorithm 7.2.3):

1. By clearing denominators, we may assume that the entries of A are integers.

2. Compute the echelon forms B_p of the reduction $A \pmod{p}$ of A modulo several primes $P = \{p, \ldots\}$, using some variant of Algorithm 7.1.3. (Note that arithmetic modulo p for a "machine size" prime p is *very* fast.)

3. Use the Chinese Remainder Theorem to find a matrix B with integer entries such that $B \equiv B_p \pmod{p}$ for all $p \in P$.

4. Use rational reconstruction (see below) to find a matrix C whose coefficients are rational numbers n/r such that $|n|, r \leq \sqrt{m/2}$, where m is the product of the primes in P, and $C \equiv B_p \pmod{p}$ for each prime p.

5. Use height bounds to verify that C is the reduced row echelon form of A.

Rational reconstruction is a process that allows one to sometimes lift an integer modulo m uniquely to a bounded rational number.

Algorithm 7.2.1 (Rational Reconstruction). *INPUT: An integer $a \geq 0$ and an integer $m \geq 1$.*

OUTPUT: The numerator and denominator n, d of the unique rational number n/d, if it exists, with

$$|n|, d \leq \sqrt{\frac{m}{2}} \qquad \textit{and} \qquad n \equiv ad \pmod{m},$$

or returns $n = d = 0$, if no such rational number exists.

```
def rational_reconstruction(a, m):
    # Reduce a modulo m
    a = a % m
    # Trivial special cases
    if a == 0: return (0,1)
    if a == 1: return (1,1)
    # Let bnd be the integer part of the square root of m/2.
    bnd = sqrt(m/2.0)
    # Initialize Euclidean algorithm.
    u = m
    v = a
    # Perform the extended Euclidean algorithm, but terminate
    # when V[2] is <= bnd.
    U = (1,0,u)
    V = (0,1,v)
    while abs(V[2]) > bnd:
        q = U[2]//V[2]      # // means divide and take the integer part
        tmp = (U[0]-q*V[0], U[1]-q*V[1], U[2]-q*V[2])
        U = V
        V = tmp
    d = abs(V[1])
    n = V[2]
    if V[1] < 0:  n = n * (-1)
    if d <= bnd and gcd(n,d) == 1:
        return (n,d)
    return (0,0)
```

Remark 7.2.2 (Technical Python Remarks). In Python, use the `sqrt` function from the `gmpy` GMP library, not the one from `math`. With Python integers, `a/b` also means divide and take the floor, i.e., what we denote by `a//b` above. Finally, `gcd` is not included with Python. Use, e.g., the `gmpy.gcd` function.

Algorithm 7.2.1 for rational reconstruction is described (with a complete nontrivial proof) in [Knu, pg.656–657] as the solution to exercise 51 on page 379. See in particular the paragraph right in the middle of page 657, which describes the algorithm. Knuth says this rational reconstruction algorithm is due to Wang, Kornerup, and Gregory from around 1983.

We now give an indication of why Algorithm 7.2.1 computes the rational reconstruction of $a \pmod{m}$, leaving the precise details and uniqueness to [Knu, pg.656–657]. At each step in Algorithm 7.2.1, the 3-tuple $V = (v_0, v_1, v_2)$ satisfies

$$m \cdot v_0 + a \cdot v_1 = v_2, \tag{7.2.1}$$

and similarly for U. When computing the usual extended gcd, at the end $v_2 = \gcd(a, m)$ and v_0, v_1 give a representation of the v_2 as a $\mathbb{Z}$-linear combination of m and a. In Algorithm 7.2.1, we are instead interested in finding a rational number n/d such that $n \equiv a \cdot d \pmod{m}$. If we set $n = v_2$ and $d = v_1$ in (7.2.1) and rearrange, we obtain

$$n = a \cdot d + m \cdot v_0.$$

Thus at *every* step of the algorithm we find a rational number n/d such that $n \equiv ad \pmod{m}$. The problem at intermediate steps is that, e.g., v_0 could be 0, or n or d could be too large.

If A is a matrix with rational entries, let $H(A)$ be the *height* of A, which is the maximum of the absolute values of the numerators and denominators of all entries of A.

Algorithm 7.2.3 (Modular Algorithm for Computing Echelon Form).
INPUT: An $m \times n$ matrix A with entries in $\mathbb{Q}$.
OUTPUT: The reduced row echelon form of A.

1. Rescale the input matrix A to have integer entries. This does not change the echelon form and makes reduction modulo many primes easier. Henceforth we assume A has integer entries.
2. Let c be a guess for the height of the echelon form.
3. List successive primes $p_1, p_2, \ldots$ such that the product of the p_i is bigger than $n \cdot c \cdot H(A) + 1$, where n is the number of columns of A.
4. Compute the echelon forms B_i of the reduction $A \pmod{p_i}$ using, e.g., Algorithm 7.1.3 or something similar.
5. Discard any B_i whose pivot column list is not maximal among pivot lists of all B_j found so far. (The pivot list associated to B_i is the ordered list of integers k such that the kth column of B_j is a pivot column. We mean maximal with respect to the following ordering on integer sequences: shorter integer sequences are smaller, and if two sequences have the same length, then order in reverse lexicographic order. Thus $[1, 2]$ is smaller than $[1, 2, 3]$, and $[1, 2, 7]$ is smaller than $[1, 2, 5]$. Think of maximal as "optimal", i.e., best possible pivot columns.)
6. Use the Chinese Remainder Theorem to find a matrix B with integer entries such that $B \equiv B_i \pmod{p_i}$ for all p_i.
7. Use rational reconstruction (Algorithm 7.2.1) to try to find a matrix C whose coefficients are rational numbers n/r such that $|n|, r \leq \sqrt{M/2}$, where $M = \prod p_i$, and $C \equiv B_i \pmod{p_i}$ for each prime p. If rational reconstruction fails, compute a few more echelon forms mod the next few primes (using the above steps), and attempt rational reconstruction again. Let E be the matrix over $\mathbb{Q}$ so obtained.

8. Compute the denominator d of E, i.e., the smallest positive integer such that dE has integer entries. If

$$H(dE) \cdot H(A) \cdot n \leq \prod p_i, \tag{7.2.2}$$

then E is the reduced row echelon form of A. If not, repeat the above steps with a few more primes.

Proof. We prove that if the bound (7.2.2) is satisfied, then the matrix E computed by the algorithm really is the reduced row echelon form R of A. The set of pivot columns of all matrices B_i used to construct E are the same, so the pivot columns of E are the same as those of any B_i. Thus E is in reduced row echelon form.

Recall from the end of Section 7.1 that a matrix whose columns are a basis for the kernel of A can be obtained from the reduced row echelon form of R. Let K be the matrix whose columns are the vectors in the kernel algorithm applied to E, so $EK = 0$. Since the reduced row echelon form is got by left multiplying by an invertible matrix, for each i, there is an invertible matrices C_i mod p_i such that $A = C_i B_i$ so

$$A \cdot dK \equiv dC_i B_i K \equiv C_i \cdot dE \cdot K \equiv 0 \pmod{p_i}.$$

Since dK and A are integer matrices,

$$A \cdot dK \equiv 0 \pmod{\prod p_i}.$$

The integer entries of $A \cdot dK$ are all at most $H(A) \cdot H(dK) \cdot n$, where n is the number of columns of A. Since $H(K) \leq H(E)$, the bound (7.2.2) implies that $A \cdot dK = 0$. Thus $AK = 0$, so $\mathrm{Ker}(E) \subset \mathrm{Ker}(A)$. On the other hand, the rank of E equals the rank of each B_i (since the pivot columns are the same), so

$$\mathrm{rank}(E) = \mathrm{rank}(B_i) = \mathrm{rank}(A \pmod{p_i}) \leq \mathrm{rank}(A).$$

Thus $\dim(\mathrm{Ker}(A)) \leq \dim(\mathrm{Ker}(E))$, and combining this with the bound obtained above we see that $\mathrm{Ker}(E) = \mathrm{Ker}(A)$. This implies that E is the reduced row echelon form of A, since two matrices have the same kernel if and only if they have the same reduced row echelon form (the echelon form is an invariant of the row space, and the kernel is the orthogonal complement of the row space).

The reason for Step 5 is that the matrices B_i need *not* be the reduction of R modulo p_i, and indeed this reduction might not even be defined, e.g., if p_i divides the denominator of some element of R, then this reduction makes no sense. For example, set $p = p_i$ and suppose $A = \left(\begin{smallmatrix} p & 1 \\ 0 & 0 \end{smallmatrix}\right)$. Then $R = \left(\begin{smallmatrix} 1 & 1/p \\ 0 & 0 \end{smallmatrix}\right)$, which has no reduction modulo p; also, the reduction of A modulo B_i is $B_i = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right) \pmod{p}$, which is already in reduced row echelon form. However if we were to combine B_i with the echelon form of A modulo another prime, the result could never be lifted using rational reconstruction. Thus the reason we exclude all B_i with non-maximal pivot column sequence is so that a rational reconstruction will exist. There are only finitely many primes that divide denominators of entries of R, so eventually all B_i will have maximal pivot column sequences, i.e., are the reduction of the true reduced row echelon form R, so the algorithm terminates. □

Remark 7.2.4.

1. I learned about rational reconstruction in the context of computing echelon forms from Allan Steel, who is one of the developers of Magma. I learned from Allan that Magma does not use the above algorithm; instead it uses a Strassen "divide and conquer" echelon procedure that involves random permuting of rows, etc., and takes advantage of asymptotically fast matrix multiplication algorithms. The matrix multiplies are done using a modular CRT technique. This is probably better in many cases, especially for dense matrices.

2. I have tested an implementation of Algorithm 7.2.3 against MAGMA V2.11-8. For large square matrices over $\mathbb{Q}$, e.g., over a hundred rows, (a case of importance when cutting out eigenspaces for Hecke operators), Algorithm 7.2.3 is much more efficient (both in time and memory usage) than MAGMA. In contrast, for matrices with more columns than rows (an important case, e.g., when intersecting subspaces), MAGMA is often an order of magnitude faster. Thus an optimal package should probably implement both Algorithm 7.2.3 for square matrices and a divide and conquer echelon strategy for non-square matrices.

3. I have never seen Algorithm 7.2.3 anywhere else, and found the details and proof myself. I have seen the idea of using a multi-modular method for linear algebra problems hinted out or explicitly suggested *many times*; I've just never seen a discussion of computing reduced row echelon forms this way.

4. There is also an iterative p-adic method for lifting solutions modulo p to an equation $Ax = v$ to characteristic 0. This is supposed to be faster for a single solution, but slower for lifting many solutions. See `http://magma.maths.usyd.edu.au/users/allan/gb/faugere_f4.ps.gz` for a discussion.

5. Algorithm 7.2.3, with all matrices **sparse**, seems to work very well in practice. A simple but helpful modification to Algorithm 7.1.3 in the sparse case is to clear each column using a row with a minimal number of nonzero entries, so as to reduce the amount of "fill in" (denseness) of the matrix. There are more sophisticated methods along these lines called "intelligent Gauss elimination". (Cryptographers are interested in linear algebra with huge sparse linear, since they come up in factor basis attacks on the discrete log problem or integer factorization.)

One can likely adapt Algorithm 7.2.3 to computation of reduced row echelon forms of matrices A over cyclotomic fields $\mathbb{Q}(\zeta_n)$. Assume A has denominator 1. Let p be a prime that splits completely in $\mathbb{Q}(\zeta_n)$. Compute the homomorphisms $f_i : \mathbb{Z}_p[\zeta_n] \to \mathbb{F}_p$ by finding the elements of order n in $\mathbb{F}_p^*$. Then compute the mod p matrix $f_i(A)$ for each i, and find its reduced row echelon form. Taken together, the maps f_i together induce an isomorphism $\Psi : \mathbb{F}_p[X]/\Phi_n(X) \cong \mathbb{F}_p^d$, where $\Phi_n(X)$ is the nth cyclotomic polynomial and d is its degree. It's easy to compute $\Psi(f(x))$ by evaluating $f(x)$ at each element of order n in $\mathbb{F}_p$. To compute Ψ^{-1} simply use linear algebra over $\mathbb{F}_p$ to invert a matrix that represents Ψ. Use Ψ^{-1} to compute the the reduced row echelon form of $A \pmod{p}$, where (p) is the non-prime ideal in $\mathbb{Z}[\zeta_n]$ generated by p. Do this for several primes p, and use rational reconstruction on each coefficient of each power of ζ_n, to recover the echelon form of A. Problems: What is the analogue of (7.2.2)?

7.3 Polynomials

There are several linear algebra algorithms that involve polynomials and are important to modular forms algorithms.

Computation of characteristic polynomials of matrices is crucial to modular forms computations. There are many approaches to this problems: compute $\det(xI - A)$ symbolically (bad), compute the traces of the powers of A (bad), or compute the Hessenberg form modulo many primes and use CRT (not so bad, see [Coh93, §2.2.4]). Another more sophisticated method is to compute the rational canonical form of A using Giesbrecht's algorithms, which involve computing Krylov subspaces (i.e., cyclic spaces spanned by a single vector), and building up the whole space on which A acts. This latter method may be viewed as a generalization of Wiedemann's algorithm for computing minimal polynomials (see Section 7.4.1), but with more structure. The algorithm used in MAGMA is similar to Giesbrecht's (probably independently discovered). PARI uses only Lagrange interpolation (?) and Hessenberg form.

Factorization of polynomials in $\mathbb{Z}[X]$ is an important step in computing an explicit basis of newforms for a space of modular forms. The best algorithm is the van Hoeij method, which uses LLL in a novel way to solve the sort of optimization problems that come up in trying to lift factorizations mod p to $\mathbb{Z}$. It has aparently been generalized to number fields and is included in new versions of PARI, MAGMA, and NTL. For more details, see van Hoeij's web page: `http://www.math.fsu.edu/~hoeij/papers.html`.

7.4 Decomposing Spaces

Fix a weight k, integer N, and Dirichlet character ε modulo N. Let

$$V = \mathbb{S}_k(\Gamma_1(N), \varepsilon)^{+\,\mathrm{new}}$$

be the new subspace of the $+1$ quotient of cuspidal modular symbols, viewed as a $K = \mathbb{Q}(\varepsilon)$ vector space. In this section we will describe an algorithm to write V as a direct sum of simple $\mathbb{T}$-submodules. It is a consequence of Atkin-Lehner-Li theory and the isomorphism between cusp forms and certain modular symbols that V is a direct sum of distinct simple modules, and that the Hecke operators T_n all act diagonalizably on V.

Let R denote the image of $\mathbb{T} \otimes K$ in $\mathrm{End}(V)$, and let $n = \dim(V)$. Since R is semisimple and finite dimensional over a field, R is a product $\prod K_i$ of number fields, so a random Hecke operator T will, with high probability, generate R as a K-algebra. (The elements that don't generate lie in proper K-subalgebras of R, and those subalgebras are direct sums of subsets of the K_i.) If T generates R as an algebra, then the minimal polynomial f of T has degree n, so it equals the characteristic polynomial of T. Also since T is diagonalizable, the minimal polynomial of T is square free. Thus we are led to the following problem:

Problem 7.4.1. Suppose T is an $n \times n$ matrix with entries in K and that the minimal polynomial of T is square free and has degree n. View T as acting on $V = K^n$. Find the (unique up to order) simple module decomposition $W_0 \oplus \cdots \oplus W_m$ of V as a direct sum of simple $K[T]$-modules. Equivalently, find an invertible matrix A such that $A^{-1}TA$ is a block direct sum of matrices $T_0, \ldots, T_m$ such that the minimal polynomial of each T_i is irreducible.

Remark 7.4.2. A natural generalization of Problem 7.4.1 to arbitrary matrices is to find the *rational Jordan form* of T. This form is like the usual Jordan form, but the summands corresponding to eigenvalues are replaced by certain matrices with minimal polynomials the minimal polynomials of the eigenvalues. The rational Jordan form was extensively studied by Geisbrecht in his Ph.D. thesis and many successive papers, where he carefully analyzes the complexity of his algorithms in terms of bit operations, and observes that the limiting step is factoring polynomials over K. The reason is that given a polynomial $f \in K[x]$, one can easily write down a matrix T such that one can can read off the factorization of f from the rational Jordan form of T. See also Allan Steel's related paper (*A New Algorithm for the Computation of Canonical Forms of Matrices over Fields*, J. Symbolic Computation (1997) **24**, 409–432). The author would also like to thank Allan Steel for discussions related to this chapter.

7.4.1 Wiedemann's Minimal Polynomial Algorithm

In this section we describe an algorithm due to Wiedemann for computing the minimal polynomial of an $n \times n$ matrix A over a field.

Choose a random vector v and compute the iterates

$$v_0 = v, \quad v_1 = A(v), \quad v_2 = A^2(v), \quad \ldots, \quad v_{2n-1} = A^{2n-1}(v).$$

If $f = x^m + c_{m-1}x^{m-1} + \cdots + c_1 x + c_0$ is the minimal polynomial of A, then

$$A^m + c_{m-1}A^{m-1} + \cdots + c_0 I_n = 0,$$

where I_n is the $n \times n$ identity matrix. For any $k \geq 0$, by multiplying both sides on the right by $A^k v$, we see that

$$A^{m+k}v + c_{m-1}A^{m-1+k}v + \cdots + c_0 A^k v = 0,$$

hence

$$v_{m+k} + c_{m-1}v_{m-1+k} + \cdots + c_0 v_k = 0, \qquad \text{all } k \geq 0.$$

Wiedemann's clever idea is to observe that any single component of the vectors $v_0, \ldots, v_{2n-1}$ satisfies a linear recurrence with coefficients $1, c_{m-1}, \ldots, c_0$. There is an algorithm (see Algorithm 7.4.4 below) called the Berlekamp-Massey algorithm (which was introduced in the 1960s in the context of coding theory) that finds the minimal polynomial of a linear recurrence sequence $\{a_r\}$. The minimal polynomial of this linear recurrence is by definition the unique monic polynomial g, such that if $\{a_r\}$ satisfies a linear recurrence $a_{j+k} + b_{j-1}a_{j-1+k} + \cdots + b_0 a_k = 0$ (for all $k \geq 0$), then g divides the polynomial $x^j + \sum_{i=0}^{j-1} b_i x^i$. In particular, if we apply Berlekamp-Massey to the top coordinates of the v_i, we obtain a polynomial g_0, which divides f. We then apply it to the second to the top coordinates and find a polynomial g_1 that divides f, etc., Taking the least common multiple of the first few g_i, we find a divisor of the minimal polynomial of f. One can show that with "high probability" one quickly finds f, instead of just a proper divisor of f.

Remark 7.4.3. In the literature, techniques that involve iterating a vector are often called Krylov methods. The subspace generated by the iterates of a vector under a matrix is called a Krylov subspace.

In the context of decomposing spaces of modular forms, we will start with a matrix for which it is likely that the degree of the minimal polynomial f equals the number of rows of A.

Here's the Berlekamp-Massey algorithm.

Algorithm 7.4.4 (Berlekamp-Massey). *INPUT: The first $2n$ terms $a_0, \ldots, a_{2n-1}$ of a linear sequence that satisfies a linear recurrence of degree at most n.*
OUTPUT: The minimal polynomial f of the sequence.

1. *Let $R_0 = x^{2n}$, $R_1 = \sum_{i=0}^{2n-1} a_i x^i$, $V_0 = 0$, $V_1 = 1$.*
2. *While $\deg(R_1) \geq n$ do the following:*
   (a) *Compute Q and R such that $R_0 = QR_1 + R$.*
   (b) *Let $(V_0, V_1, R_0, R_1) = (V_1, V_0 - QV_1, R_1, R)$.*
3. *Let $d = \max(\deg(V_1), 1 + \deg(R_1))$ and set $P = x^d V_1(1/x)$.*
4. *Let c be the leading coefficient of P and output $f = P/c$.*

For a fresh viewpoint on Berlekamp-Massey and some ideas for improvement, see *The Berlekamp-Massey Algorithm revisited* by Atti, Diaz-Toca, and Lombardi (see `http://hlombardi.free.fr/publis/ABMAvar.html`) (Note: I essentially copied the above description of the Berlekamp-Massey algorithm from loc. cit.; my point is only to illustrate that the Berlekamp-Massey is basically just the Euclidean algorithm, i.e., it's not something really complicated.)

Now suppose T is an $n \times n$ matrix as in Problem 7.4.1. We find the minimal polynomial of T by computing the minimal polynomial of $T \pmod{\wp}$, using Wiedemann's algorithm, for many primes $\wp$ and using the Chinese remainder theorem. (One has to bound the number of primes that must be considered; see, e.g., [Coh93].)

One can also compute the characteristic polynomial of T directly from the Hessenberg form of T, which can be computed in $O(n^4)$ field operations, as described in [Coh93]. This is simple, but slow. Also, the T we consider will often be sparse, and Wiedemann is particularly good when T is sparse.

Example 7.4.5. We compute the minimal polynomial of the Hecke operator $A = T_2$ on $\mathbb{M}_2(\Gamma_0(23))^+$ using Wiedemann's algorithm. We have

$$A = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 2 \\ -1 & 1/2 & -1 \end{pmatrix}$$

Let $v = (1, 0, 0)^t$. Then

$$\begin{aligned} v = (1,0,0)^t, \quad Av = (3,0,-1)^t, \quad A^2v = (9,-2,-2)^t, \\ A^3v = (27,-4,-8)^t, \quad A^4v = (81,-16,-21)^t, \quad A^5v = (243,-42,-68)^t. \end{aligned}$$

The linear recurrence sequence coming from the first entries is

$$1, 3, 9, 27, 81, 243.$$

This sequence satisfies the linear recurrence

$$a_{k+1} - 3a_k = 0, \qquad \text{all } k > 0$$

so its minimal polynomial is $x - 3$. This implies that $x - 3$ divides the minimal polynomial of the matrix A. Next we use the sequence of second coordinates of the iterates of v, which is

$$0, 0, -2, -4, -16, -42.$$

The recurrence that this sequence satisfies is slightly less obvious, so we apply the Berlekamp-Massey algorithm to find it, with $n = 3$.

1. We have $R_0 = x^6$, $R_1 = -42x^5 - 16x^4 - 4x^3 - 2x^2$, $V_0 = 0, V_1 = 1$.

2. (a) Dividing R_0 by R_1, we find

$$R_0 = R_1\left(-\frac{1}{42}x + \frac{4}{441}\right) + \left(\frac{22}{441}x^4 - \frac{5}{441}x^3 + \frac{8}{441}x^2\right)$$

(b) The new V_0, V_1, R_0, R_1 are

$$\begin{aligned}
V_0 &= 1\\
V_1 &= \frac{1}{42}x - \frac{4}{441}\\
R_0 &= -42x^5 - 16x^4 - 4x^3 - 2x^2\\
R_1 &= \frac{22}{441}x^4 - \frac{5}{441}x^3 + \frac{8}{441}x^2
\end{aligned}$$

Since $\deg(R_1) \geq n = 3$, we have to do the above three steps again.

3. We repeat the preceeding three steps.

(a) Dividing R_0 by R_1, we find

$$R_0 = R_1\left(-\frac{9261}{11}x - \frac{123921}{242}\right) + \left(\frac{1323}{242}x^3 + \frac{882}{121}x^2\right)$$

(b) The new V_0, V_1, R_0, R_1 are [I'm running out of `\frac` steam.]

$$\begin{aligned}
V_0 &= 1/42x - 4/441\\
V_1 &= 441/22x^2 + 2205/484x + 441/121\\
R_0 &= 22/441x^4 - 5/441x^3 + 8/441x^2\\
R_1 &= 1323/242x^3 + 882/121x^2
\end{aligned}$$

4. Unfortunately we have to repeat the steps yet again. We get

$$\begin{aligned}
V_0 &= 441/22x^2 + 2205/484x + 441/121\\
V_1 &= -242/1323x^3 + 968/3969x^2 + 484/3969x - 242/3969\\
R_0 &= 1323/242x^3 + 882/121x^2\\
R_1 &= 484/3969x^2
\end{aligned}$$

5. We have $d = 3$, so $P = -242/3969x^3 + 484/3969x^2 + 968/3969x - 242/1323$.

6. Multiply through by $-3969/242$ and output

$$x^3 - 2x^2 - 4x + 3 = (x - 3)(x^2 + x - 1).$$

The minimal polynomial of T_2 is $(x-3)(x^2+x-1)$, since the minimal polynomial has degree at most 3 and is divisible by $(x - 3)(x^2 + x - 1)$.

7.4.2 Polynomial Factorization

There is a new algorithm due to Hoeij, which has been refined by Belebas, Klüners, and Steel, for factoring polynomials over number fields (and more general global fields). It involves factoring modulo many primes, lifting p-adically, and cleverly using LLL to solve a certain "knapsack problem" that reduces the number of subsets of factors that need to be considered. We will say nothing more about it here, except that it is rumored to be "very fast", and it is the algorithm to know about. [After a quick reading of Belebas, Hoeij, Klüners, and Steel, the O complexity is unclear to me.]

7.4.3 Decomposition Using Kernels

We now know enough to give an algorithm to solve Problem 7.4.1.

Algorithm 7.4.6 (Decomposition Using Kernels). *INPUT: An $n \times n$ matrix T over a field K as in Problem 7.4.1.*
OUTPUT: Decomposition of V as a direct sum of simple $K[T]$ modules.

1. *[Minimal Polynomial] Compute the minimal polynomial f of T, e.g., using the multi-modular Wiedemann algorithm.*
2. *[Factorization] Factor f using the Belebas, Hoeij, Klüners, and Steel algorithm.*
3. *[Compute Kernels] For each irreducible factor g_i of f:*
   - *(a) Compute the matrix $A_i = g_i(T)$. (This is difficult, and A will have huge coefficients.)*
   - *(b) Compute $W_i = \ker(A_i)$ using, e.g., a multi-modular kernel algorithm.*
4. *[Output Answer] Then $V = \oplus W_i$.*

Remark 7.4.7. In the worst case, perhaps Step 2 is most difficult step. In practice Step 3 is very time consuming. As mentioned in Remark 7.4.2, if one can compute such decompositions $V = \oplus W_i$, then one can easily factor polynomials f, hence the difficulty of polynomial factorization is a lower bound on the complexity of writing V as a direct sum of simples.

7.4.4 Multi-Modular Decomposition Algorithm

The following algorithm is a modification of Algorithm 7.4.6, which improves upon the difficult Step 3.

Algorithm 7.4.8 (Decomposition Algorithm II). *INPUT: An $n \times n$ matrix T over a field K as in Problem 7.4.1.*
OUTPUT: Decomposition of V as a direct sum of simple $K[T]$ modules.

1. *[Minimal Polynomial] Compute the minimal polynomial f of T, e.g., using the multi-modular Wiedemann algorithm.*

2. *[Factorization] Factor* $f = \prod g_i$ *using the Belebas, Hoeij, Klüners, and Steel algorithm.*

3. *[Cofactors] For each* i*, let* $h_i = f/g_i$*.*

4. *[Find Kernels] For several primes* $\wp$ *(how many?), compute reduced row echelon forms for basis of all the kernels* $\overline{W}_i = \ker(g_i(\overline{T}))$ *as follows:*

   (a) *Choose a random vector* $v \in \overline{V}$*.*

   (b) *Compute the iterates*

   $$v_0 = v, \quad v_1 = \overline{T}v, \quad \ldots, \quad v_{n-1} = \overline{T}^{n-1}v.$$

   (c) *For each* i *do the following:*

      i. *Compute* $w = h_i(\overline{T})v \in \ker(g_i(\overline{T}))$ *by taking the linear combination of the* v_i *given by the coefficients of* h_i*.*

      ii. *Generate a subspace of* $\ker(g_i(\overline{T}))$ *using* $w, \overline{T}w, \ldots, \overline{T}^i w$*, keeping the subspace basis in Echelon form at each step. If this subspace does not equal the full* $\ker(g_i(\overline{T}))$*, repeat the above steps with another* v*, and add the resulting iterates of the new* w *to this subspace. Repeat this process until we obtain a basis for* $\ker(g_i(\overline{T}))$*, in reduced row echelon form.*

5. *[Lift] Using the Chinese remainder theorem and rational reconstruction, lift the* $\overline{W}_i$ *to* K*-vector spaces* W_i *such that* $V = \oplus W_i$ *is the desired decomposition. (WARNING: It is probably necessary to throw away "bad" primes, just as we did in the multi-modular echelon algorithm.)*

Chapter 8

Modular Symbols of any Weight and Level

In this chapter we explain how to generalize the notion of modular symbols from Chapter 3 to compute most classical modular forms.

Modular symbols are a formalism that make it fairly easy and elementary to compute with homology or cohomology related to certain Kuga-Sato varieties (these are $\mathcal{E} \times_X \cdots \times_X \mathcal{E}$, where X is a modular curve and $\mathcal{E}$ is the univeral elliptic curve over it). It is not necessary to know anything about these Kuga-Sato varieties in order to compute with modular symbols.

This chapter is about spaces of modular symbols and how to compute with them. It is by far the most important chapter in this book. The algorithms that build on the theory in this chapter are central to all the computations we will do later in the book. We will start with the basics, in that the intended reader of this chapter is not assumed to have ever seen a modular symbol before.

Much of this chapter follows Loic Merel's paper [Mer94] very closely. First we define modular symbols of weight $k \geq 2$. Then we define the corresponding Manin symbols, and state a theorem of Merel-Shokurov, which gives all relations between Manin symbols. (The proof of the Merel-Shokurov theorem is beyond the scope of this book.) Next we describe how the Hecke operators act on both modular and Manin symbols, and how to compute trace and inclusion maps between spaces of modular symbols of different levels. We close the chapter with a discussion of computations with modular symbols over finite fields.

In this book we will view modular symbols primarily as a formalism that generates algorithms for computing with modular forms. I.e., *we view modular symbols as modular forms for computers*. However, modular symbols have also been used to prove theoretical results about modular forms. For example, certain technical calculations with modular symbols are used in Loic Merel's proof of the uniform boundedness conjecture for torsion points on elliptic curves over number fields; modular symbols arise, e.g., in order to understand linear independence of Hecke operators. Another example is Grigor Grigorov's in-progress

Ph.D. thesis, which distills hypotheses about Kato's Euler system in K_2 of modular curves to a simple formula involving modular symbols (when the hypotheses are satisfied, one obtains a lower bound on the Shafarevich-Tate group of an elliptic curve).

8.1 Modular Symbols

We begin by defining a free abelian group $\mathbb{M}$ of modular symbols, which you should think of as the homology of the extended upper half plane $\mathfrak{h}^* = \mathfrak{h} \cup \mathbb{P}^1(\mathbb{Q})$ relative to the cusps. This is the free abelian group on symbols $\{\alpha, \beta\}$ with

$$\alpha, \beta \in \mathbb{P}^1(\mathbb{Q}) = \mathbb{Q} \cup \{\infty\}$$

subject to the relations

$$\{\alpha, \beta\} + \{\beta, \gamma\} + \{\gamma, \alpha\} = 0,$$

for all $\alpha, \beta, \gamma \in \mathbb{P}^1(\mathbb{Q})$. More precisely, $\mathbb{M} = (F/R)/(F/R)_{\mathrm{tor}}$, where F is the free abelian group on all pairs (α, β) and R is the subgroup generated by all elements of the form $(\alpha, \beta) + (\beta, \gamma) + (\gamma, \alpha)$. Note that $\mathbb{M}$ is a huge free abelian group of countable rank.

Remark 8.1.1 (Warning!). The $\{\alpha, \beta\}$ satisfy the relations $\{\alpha, \beta\} = -\{\beta, \alpha\}$, since $\{\alpha, \beta\} + \{\beta, \alpha\} + \{\alpha, \alpha\} = 0$. Thus the order matters. The notation $\{\alpha, \beta\}$ looks like the set containing two elements, which strongly (and incorrectly) suggests that the order does not matter. This is annoying, but it is the standard notation, and we will stick with it.

Now fix an integer $k \geq 2$. Let $\mathbb{Z}_{k-2}[X, Y]$ be the abelian group of homogeneous polynomials of degree $k-2$ in two variables X, Y (so $\mathbb{Z}_{k-2}[X, Y]$ is isomorphic to $\mathrm{Sym}^{k-2}(\mathbb{Z})$ as a group, but certain natural actions are different). Set

$$\mathbb{M}_k = \mathbb{Z}_{k-2}[X, Y] \otimes_{\mathbb{Z}} \mathbb{M},$$

which is a torsion-free abelian group whose elements are sums of expressions of the form $X^i Y^{k-2-i} \otimes \{\alpha, \beta\}$. For example,

$$X^3 \otimes \{0, 1/2\} - 17XY^2 \otimes \{\infty, 1/7\} \in \mathbb{M}_5.$$

Fix a finite index subgroup G of $\mathrm{SL}_2(\mathbb{Z})$. Define a *left action of* G on $\mathbb{Z}_{k-2}[X, Y]$ as follows. If $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in G$ and $P(X, Y) \in \mathbb{Z}_{k-2}[X, Y]$, let

$$(g.P)(X, Y) = P(dX - bY, -cX + aY).$$

Note that if we think of $z = (X, Y)$ as a column vector, then

$$(g.P)(z) = P(g^{-1}z),$$

since $g^{-1} = \left(\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right)$, since $\det(g) = 1$. The reason for the inverse is so that this is a left action instead of a right action, which is what function pre-composition always is. As further explanation, observe that if $g, h \in G$, then

$$((gh).P)(z) = P((gh)^{-1}z) = P(h^{-1}g^{-1}z) = (h.P)(g^{-1}z) = (g.(h.P))(z).$$

Let G act on the left on $\mathbb{M}$ by

$$g.\{\alpha, \beta\} = \{g(\alpha), g(\beta)\}.$$

Here G is acting via linear fractional transformations, so if $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, then

$$g(\alpha) = \frac{a\alpha + b}{c\alpha + d}.$$

For example, useful special cases to remember are that if $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ then

$$g(0) = \frac{b}{d} \qquad \text{and} \qquad g(\infty) = \frac{a}{c}.$$

We now combine these two actions to obtain a left action of G on $\mathbb{M}_{k-2}$, which is given by

$$g.(P \otimes \{\alpha, \beta\}) = (g.P) \otimes \{g(\alpha), g(\beta)\}.$$

For example,

$$\begin{aligned} \begin{pmatrix} 1 & 2 \\ -2 & -3 \end{pmatrix}.(X^3 \otimes \{0, 1/2\}) &= (-3X - 2Y)^3 \otimes \left\{-\frac{2}{3}, -\frac{5}{8}\right\} \\ &= (-27X^3 - 54X^2Y - 36XY^2 - 8Y^3) \otimes \left\{-\frac{2}{3}, -\frac{5}{8}\right\}. \end{aligned}$$

We will often write $P(X,Y)\{\alpha, \beta\}$ for $P(X,Y) \otimes \{\alpha, \beta\}$.

Definition 8.1.2 (Modular Symbols). Let $k \geq 2$ be an integer and let G be a finite index subgroup of $\mathrm{SL}_2(\mathbb{Z})$. The space $\mathbb{M}_k(G)$ of weight k modular symbols for G is the quotient of $\mathbb{M}_k$ by all relations $g.x - x$ for $x \in \mathbb{M}_k$ and by any torsion.

Note that $\mathbb{M}_k$ is a torsion free abelian group, and it is a nontrivial fact that $\mathbb{M}_k$ has finite rank. We denote modular symbols for G in exactly the same way we denote elements of $\mathbb{M}_k$, but with surrounding text that hopefully makes the group G clear. Thus $X^3\{0, 1/2\}$ is an example element of $\mathbb{M}_5(\Gamma_0(8))$, because I say so. In practice this does not cause confusion.

The space of *modular symbols over a ring R* is

$$\mathbb{M}_k(G, R) = \mathbb{M}_k(G) \otimes_{\mathbb{Z}} R.$$

In Section **??** we will discuss computing $\mathbb{M}_k(G, R)$ when R is a finite field.

8.2 Manin Symbols

At this point you are probably wondering how one could possibly ever program a computer to *compute* $\mathbb{M}_k(G)$ for any specific k and G. As defined above, $\mathbb{M}_k(G)$ is the quotient of one infinitely generated abelian group by another one. This section is about Manin symbols, which are simply a distinguished subset of the elements of $\mathbb{M}_k(G)$ that lead to a finite presentation for $\mathbb{M}_k(G)$. Also, it has emerged that formulas written in terms of Manin symbols are frequently much easier to compute using a computer than formulas in terms of modular symbols.

The *Manin symbol* associated to $g \in \mathrm{SL}_2(\mathbb{Z})$ and $P \in \mathbb{Z}_{k-2}[X,Y]$ is

$$[P, g] = g.(P\{0, \infty\}) \in \mathbb{M}_k(G).$$

Notice that if $Gg = Gh$, then $[P, g] = [P, h]$, since the symbol $g.(P\{0, \infty\})$ is invariant by the action of G on the left (by definition, since it is a modular symbols for G). Thus we can also write $[P, Gg]$, and since G has finite index in $\mathrm{SL}_2(\mathbb{Z})$, the abelian group generated by Manin symbols is of finite rank, generated by

$$\bigl\{[X^{k-2-i}Y^i,\, Gg_j] \,:\, i = 0, \dots, k-2, \quad \text{and} \quad j = 0, \dots, r\bigr\},$$

where $g_0, \dots, g_r$ run through representatives for the right cosets $G\backslash \mathrm{SL}_2(\mathbb{Z})$.

The great thing about Manin symbols is that every modular symbols can be written as a $\mathbb{Z}$-linear combination of them, so they generate all $\mathbb{M}_k(G)$. The proof of this fact is known as "Manin's trick".

Proposition 8.2.1. *The Manin symbols generate* $\mathbb{M}_k(G)$.

Proof. Suppose that we are given a modular symbol $P\{\alpha, \beta\}$ and wish to represent it as a sum of Manin symbols. Because

$$P\{a/b, c/d\} = P\{a/b, 0\} + P\{0, c/d\},$$

it suffices to write $P\{0, a/b\}$ in terms of Manin symbols. Let

$$0 = \frac{p_{-2}}{q_{-2}} = \frac{0}{1},\ \frac{p_{-1}}{q_{-1}} = \frac{1}{0},\ \frac{p_0}{1} = \frac{p_0}{q_0},\ \frac{p_1}{q_1},\ \frac{p_2}{q_2},\ \dots,\ \frac{p_r}{q_r} = \frac{a}{b}$$

denote the continued fraction convergents of the rational number a/b. Then

$$p_j q_{j-1} - p_{j-1} q_j = (-1)^{j-1} \qquad \text{for } -1 \le j \le r.$$

If we let $g_j = \begin{pmatrix} (-1)^{j-1}p_j & p_{j-1} \\ (-1)^{j-1}q_j & q_{j-1} \end{pmatrix}$, then $g_j \in \mathrm{SL}_2(\mathbb{Z})$ and

$$\begin{aligned} P\{0, a/b\} &= P \sum_{j=-1}^{r} \left\{ \frac{p_{j-1}}{q_{j-1}}, \frac{p_j}{q_j} \right\} \\ &= \sum_{j=-1}^{r} g_j((g_j^{-1}P)\{0, \infty\}) \\ &= \sum_{j=-1}^{r} [g_j^{-1}P,\ g_j]. \end{aligned}$$

Since $g_j \in \mathrm{SL}_2(\mathbb{Z})$ and P has integer coefficients, the polynomial $g_j^{-1}P$ also has integer coefficients, so we introduce no denominators. $\square$

As is well known, the continued fraction expansion $[c_1, c_2, \ldots, c_n]$ of the rational number a/b can be computed using the Euclidean algorithm. The first term c_1 is the "quotient": $a = bc_1 + r$, with $0 \leq r < b$. Let $a' = b$, $b' = r$ and compute c_2 as $a' = b'c_2 + r'$, etc., terminating when the remainder is 0. For example, the expansion of $5/13$ is $[0, 2, 1, 1, 2]$. The numbers

$$d_i = c_1 + \cfrac{1}{c_2 + \cfrac{1}{c_3 + \cdots}}$$

will then be the (finite) convergents. For example if $a/b = 5/13$, then the convergents are

$$0/1,\ 1/0,\ d_1 = 0,\ d_2 = \frac{1}{2},\ d_3 = \frac{1}{3},\ d_4 = \frac{2}{5},\ d_5 = \frac{5}{13}.$$

Remark 8.2.2. One can prove Proposition 8.2.1 inductively without introducing continued fractions, but that proof is essentially the same one used to prove the existence of continued fractions of integers. (I think I saw this in [MTT86], but I can't seem to find the exact location in that paper right now.)

Now that we know the Manin symbols generate $\mathbb{M}_k(G)$, the next question is what are the relations between Manin symbols. Fortunately the answer is fairly simple (though the proof is not). Let

$$\sigma = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad \tau = \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}, \qquad J = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Define a *right action* of $\mathrm{SL}_2(\mathbb{Z})$ on Manin symbols as follows. If $h \in \mathrm{SL}_2(\mathbb{Z})$, let

$$[P, g].h = [h^{-1}.P, gh].$$

This is a right action because $P.h = h^{-1}P$ is a right action, and right multiplication $g \mapsto gh$ is also a right action.

Theorem 8.2.3. *If x is a Manin symbol, then*

$$x + x.\sigma = 0 \tag{8.2.1}$$

$$x + x.\tau + x.\tau^2 = 0 \tag{8.2.2}$$

$$x - x.J = 0. \tag{8.2.3}$$

Moreover, these are all the relations between Manin symbols, in the sense that the space $\mathbb{M}_k(G)$ of modular symbols is isomorphic to the quotient of the free abelian group on the finitely many symbols $[X^iY^{k-2-i}, Gg]$ (for $i = 0, \ldots, k-2$, and $Gg \in G\backslash\mathrm{SL}_2(\mathbb{Z})$) by the above relations and any torsion.

Proof. We will only prove the easy "half" of the theorem here. The proof of the difficult half, i.e., that the above relations are all the relations is more complicated. Merel remarks in [Mer94, §1.3] that the quotient of Manin symbols by the above relations and torsion is isomorphic to a space of Šokurov symbols, which is in turn isomorphic to $\mathbb{M}_k(G)$. He cites [Šok80] for most of the proof. See also [Ste03] for an exposition of Manin's proof from [Man72] when $k = 2$, which involves triangulating the Riemann surface $G\backslash\mathfrak{h}$.

For the proof of the easy half, i.e., that the expressions above are in fact relations, we follow Merel's proof from [Mer94, §1.2]. Note that

$$\sigma(0) = \sigma^2(\infty) = \infty \qquad \text{and} \qquad \tau(1) = \tau^2(0) = \infty.$$

Write $x = [P, g]$, we have

$$\begin{aligned}
[P, g] + [P, g].\sigma &= [P, g] + [\sigma^{-1}.P, g\sigma] \\
&= g.(P\{0, \infty\}) + g\sigma.(\sigma^{-1}.P\{0, \infty\}) \\
&= (g.P)\{g(0), g(\infty)\} + (g\sigma).(\sigma^{-1}.P)\{g\sigma(0), g\sigma(\infty)\} \\
&= (g.P)\{g(0), g(\infty)\} + (g.P)\{g(\infty), g(0)\} \\
&= (g.P)(\{g(0), g(\infty)\} + \{g(\infty), g(0)\}) \\
&= 0.
\end{aligned}$$

Also,

$$\begin{aligned}
&[P, g] + [P, g].\tau + [P, g].\tau^2 = [P, g] + [\tau^{-1}.P, g\tau] + [\tau^{-2}.P, g\tau^2] \\
&\quad = g.(P\{0, \infty\}) + g\tau.(\tau^{-1}.P\{0, \infty\}) + g\tau^2.(\tau^{-2}.P\{0, \infty\}) \\
&\quad = (g.P)\{g(0), g(\infty)\} + (g.P)\{g\tau(0), g\tau(\infty)\}) + (g.P)\{g\tau^2(0), \tau^2(\infty)\}) \\
&\quad = (g.P)\{g(0), g(\infty)\} + (g.P)\{g(1), g(0)\}) + (g.P)\{g(\infty), g(1)\}) \\
&\quad = (g.P)(\{g(0), g(\infty)\} + \{g(\infty), g(1)\} + \{g(1), g(0)\}) \\
&\quad = 0
\end{aligned}$$

Finally,

$$\begin{aligned}
[P, g] + [P, g].J &= g.(P\{0, \infty\}) - gJ.(J^{-1}P\{gJ(0), gJ(\infty)\} \\
&= (g.P)\{g(0), g(\infty)\} - (g.P)\{g(0), g(\infty)\} \\
&= 0,
\end{aligned}$$

where we use that J acts trivially via linear fractional transformations. $\square$

If G is a finite-index subgroup and we have an algorithm to enumerate the right cosets $G\backslash \mathrm{SL}_2(\mathbb{Z})$, and to decide which coset an arbitrary element of $\mathrm{SL}_2(\mathbb{Z})$ belongs to, then Theorem 8.2.3 and the algorithms of Chapter 7 yield an algorithm to compute $\mathbb{M}_k(G,\mathbb{Q})$. We will defer further discussion about precise details of algorithms to compute modular symbols until Chapter **??**). Note that if $J \in G$, then the relation $x - x.J = 0$ is automatic. Also note the matrices σ and τ *do not commute*, so one can *not* first quotient out by the two-term σ relations, then quotient out only the remaining free generators by the τ relations, and get the right answer in general.

8.2.1 Coset Representatives and Manin Symbols

Proposition 8.2.4. *The right cosets $\Gamma_1(N)\backslash \mathrm{SL}_2(\mathbb{Z})$ are in bijection with pairs (c,d) where $c,d \in \mathbb{Z}/N\mathbb{Z}$ and $\gcd(c,d,N) = 1$. The coset containing a matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ corresponds (c,d).*

Proof. This proof is copied from [Cre92, pg. 203], except in that paper Cremona works with the analogue of $\Gamma_1(N)$ in $\mathrm{PSL}_2(\mathbb{Z})$, so his result is slightly different. Suppose $\gamma_i = \left(\begin{smallmatrix} a_i & b_i \\ c_i & d_i \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z})$, for $i = 1,2$. We have

$$\gamma_1\gamma_2^{-1} = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}\begin{pmatrix} d_2 & -b_2 \\ -c_2 & a_2 \end{pmatrix} = \begin{pmatrix} a_1d_2 - b_1c_2 & * \\ c_1d_2 - d_1c_2 & a_2d_1 - b_2c_1 \end{pmatrix},$$

which is in $\Gamma_1(N)$ if and only if

$$c_1d_2 - d_1c_2 \equiv 0 \pmod{N} \tag{8.2.4}$$

and

$$a_2d_1 - b_2c_1 \equiv a_1d_2 - b_1c_2 \equiv 1 \pmod{N}. \tag{8.2.5}$$

Since the γ_i have determinant 1, if $(c_1,d_1) = (c_2,d_2) \pmod{N}$, then the congruences (8.2.4–8.2.5) hold. Conversely, if (8.2.4–8.2.5) hold, then

$$\begin{aligned} c_2 &\equiv a_2d_1c_2 - b_2c_1c_2 \\ &\equiv a_2d_2c_1 - b_2c_2c_1 && \text{since } d_1c_2 \equiv d_2c_1 \pmod{N} \\ &\equiv c_1 && \text{since } a_2d_2 - b_2c_2 = 1, \end{aligned}$$

and likewise

$$d_2 \equiv a_2d_1d_2 - b_2c_1d_2 \equiv a_2d_1d_2 - b_2d_1c_2 \equiv d_1 \pmod{N}.$$

∎

Thus we may view weight k Manin symbols for $\Gamma_1(N)$ as triples of integers (i,c,d), where $0 \le i \le k-2$ and $c,d \in \mathbb{Z}/N\mathbb{Z}$ with $\gcd(c,d,N) = 1$. Here (i,c,d) corresponds to the Manin symbol $[X^iY^{k-2-i}, \left(\begin{smallmatrix} a & b \\ c' & d' \end{smallmatrix}\right)]$, where c' and d'

lift c, d. The relations of Theorem 8.2.3 become

$$
\begin{aligned}
(i,\, c, d) + (-1)^i (k-2-i,\, d, -c) &= 0,\\
(i,\, c, d) \quad + \quad (-1)^{k-2} \sum_{j=0}^{k-2-i} (-1)^j \binom{k-2-i}{j} (j,\, d, -c-d) &\\
+ \quad (-1)^{k-2-i} \sum_{j=0}^{i} (-1)^j \binom{i}{j} (k-2-i+j,\, -c-d, c) &= 0,\\
(i,\, c, d) - (-1)^{k-2}(i,\, -c, -d) &= 0.
\end{aligned}
$$

There is a similar description of cosets for $\Gamma_0(N)$:

Proposition 8.2.5. *The right cosets $\Gamma_0(N)\backslash \mathrm{SL}_2(\mathbb{Z})$ are in bijection with the elements of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. The coset containing a matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ corresponds to the point $(c : d) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.*

For a proof, see [Cre97a, §2.2].

8.2.2 Modular Symbols With Character

Suppose now that $G = \Gamma_1(N) \subset \mathrm{SL}_2(\mathbb{Z})$. Merel defines an action of diamond bracket operators $\langle d \rangle$, with $\gcd(d, N) = 1$, on modular and Manin symbols. On Manin symbols the action is given by

$$\langle n \rangle ([P, (c, d)]) = [P, (nc, nd)]\,.$$

Let

$$\varepsilon : (\mathbb{Z}/N\mathbb{Z})^* \to \mathbb{Q}(\zeta)^*$$

be a Dirichlet character, where ζ is an nth root of unity and n is the order of ε. Let $\mathbb{M}_k(\Gamma_1(N), \varepsilon)$ be the quotient of $\mathbb{M}_k(\Gamma_1(N), \mathbb{Z}[\zeta])$ by the relations (given in terms of Manin symbols)

$$\langle d \rangle x - \varepsilon(d) x = 0,$$

for all $x \in \mathbb{M}_k(\Gamma_1(N), \mathbb{Z}[\zeta])$, and by any torsion. Thus $\mathbb{M}_k(\Gamma_1(N), \varepsilon)$ is a torsion free $\mathbb{Z}[\varepsilon]$-module.

Remark 8.2.6. I do not know whether or not $\mathbb{M}_k(\Gamma_1(N), \varepsilon)$ is necessarily free as a $\mathbb{Z}[\varepsilon]$-module.

8.3 Hecke Operators

Just as for modular forms, there is a *Hecke algebra* $\mathbb{T} = \mathbb{Z}[T_1, T_2, \ldots]$ of Hecke operators that act on $\mathbb{M}_k(\Gamma_0(N))$. Let

$$R_p = \left\{ \begin{pmatrix} 1 & r \\ 0 & p \end{pmatrix} : r = 0, 1, \ldots, p-1 \right\} \cup \left\{ \begin{pmatrix} p & 0 \\ 0 & 1 \end{pmatrix} \right\},$$

where we omit $\left(\begin{smallmatrix} p & 0 \\ 0 & 1 \end{smallmatrix}\right)$ if $p \mid N$. Then the *Hecke operator* T_p on $\mathbb{M}_k(\Gamma_0(N))$ is given by

$$T_p(x) = \sum_{g \in R} g.x.$$

Notice when $p \nmid N$, that T_p is defined by summing over $p+1$ matrices that correspond to the $p+1$ sublattices of $\mathbb{Z} \times \mathbb{Z}$ if index p. This is exactly how we defined T_p on modular forms.

You might think at this point that we've just formally defined a computable abelian group, and defined operators formally on it that look something like the usual Hecke operators, but perhaps there's no real connection. As it turns out, the ring generated by all the Hecke operators on modular symbols is commutative, and $\mathbb{M}_k(\Gamma_1(N), \mathbb{R})$ is non-canonically isomorphic as a $\mathbb{T}$-module to $M_k(\Gamma_1(N))$. Note that $\mathbb{M}_k(\Gamma_1(N), \mathbb{R})$ is a real vector space and $M_k(\Gamma_1(N))$ is a complex vector space, so this should be viewed also as an isomorphism of $\mathbb{R}$-vector spaces. In fact there is an extra conjugation structure on $\mathbb{M}_k(\Gamma_1(N), \mathbb{R})$, which we will discuss later.

8.3.1 General Definition of Hecke Operators

Let Γ be a finite index subgroup of $\mathrm{SL}_2(\mathbb{Z})$ and suppose

$$\Delta \subset \mathrm{GL}_2(\mathbb{Q})$$

is a set such that $\Gamma\Delta = \Delta\Gamma = \Delta$ and $\Gamma\backslash\Delta$ is finite. For example, $\Delta = \Gamma$ trivially satisfies this condition. Also, if $\Gamma = \Gamma_1(N)$, then for any positive integer n, the set

$$\Delta_n = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbb{Z}) \; : \; ad - bc = n, \text{ and } \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \pmod{N} \right\}$$

also satisfies this condition, as we will now prove.

Lemma 8.3.1. *We have*

$$\Gamma_1(N) \cdot \Delta_n = \Delta_n \cdot \Gamma_1(N) = \Delta_n$$

and

$$\Delta_n = \bigcup_{a,b} \Gamma_1(N) \cdot \sigma_a \begin{pmatrix} a & b \\ 0 & n/a \end{pmatrix},$$

where $\sigma_a \equiv \left(\begin{smallmatrix} 1/a & 0 \\ 0 & a \end{smallmatrix}\right) \pmod{N}$*, the union is disjoint and* $1 \le a \le n$ *with* $a \mid n$*,* $\gcd(a, N) = 1$*, and* $0 \le b < n/a$*. In particular, the set of cosets* $\Gamma_1(N)\backslash\Delta_n$ *is finite.*

Proof. If $\gamma \in \Gamma_1(N)$ and $\delta \in \Delta_n$, then

$$\begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \cdot \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \pmod{N}.$$

Thus $\Gamma_1(N)\Delta_n \subset \Delta_n$, and since $\Gamma_1(N)$ is a group $\Gamma_1(N)\Delta_n = \Delta_n$; likewise $\Delta_n\Gamma_1(N) = \Delta_n$.

For the coset decomposition, we first prove the statement for $N = 1$, i.e., for $\Gamma_1(N) = \mathrm{SL}_2(\mathbf{Z})$. If A is an arbitrary element of $M_2(\mathbf{Z})$ with determinant n, then using row operators on the left with determinant 1, i.e., left multiplication by elements of $\mathrm{SL}_2(\mathbf{Z})$, we can transform A into the form $\left(\begin{smallmatrix} a & b \\ 0 & n/a \end{smallmatrix}\right)$, with $1 \leq a \leq n$ and $0 \leq b < n$. (Just imagine applying the Euclidean algorithm to the two entries in the first column of A. Then a is the gcd of the two entries in the first column, and the lower left entry is 0. Next subtract n/a from b until $0 \leq b < n/a$.)

Next suppose N is arbitrary. Let $g_1, \ldots, g_r$ be such that

$$g_1\Gamma_1(N) \cup \cdots \cup g_r\Gamma_1(N) = \mathrm{SL}_2(\mathbf{Z})$$

is a disjoint union. If $A \in \Delta_n$ is arbitrary, then as we showed above, there is some $\gamma \in \mathrm{SL}_2(\mathbf{Z})$, so that $\gamma \cdot A = \left(\begin{smallmatrix} a & b \\ 0 & n/a \end{smallmatrix}\right)$, with $1 \leq a \leq n$ and $0 \leq b < n/a$, and $a \mid n$. Write $\gamma = g_i \cdot \alpha$, with $\alpha \in \Gamma_1(N)$. Then

$$\alpha \cdot A = g_i^{-1} \cdot \begin{pmatrix} a & b \\ 0 & n/a \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \pmod{N}.$$

It follows that

$$g_i^{-1} \equiv \begin{pmatrix} 1 & * \\ 0 & n \end{pmatrix} \cdot \begin{pmatrix} a & b \\ 0 & n/a \end{pmatrix}^{-1} \equiv \begin{pmatrix} 1/a & * \\ 0 & a \end{pmatrix} \pmod{N}.$$

Since $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma_1(N)$ and $\gcd(a, N) = 1$, there is $\gamma' \in \Gamma_1(N)$ such that

$$\gamma' g_i^{-1} \equiv \begin{pmatrix} 1/a & 0 \\ 0 & a \end{pmatrix} \pmod{N}.$$

We may then choose $\sigma_a = \gamma' g_i^{-1}$. Thus every $A \in \Delta_n$ is of the form $\gamma\sigma_a \left(\begin{smallmatrix} a & b \\ 0 & n/a \end{smallmatrix}\right)$, with $\gamma \in \Gamma_1(N)$ and a, b suitably bounded. This proves the second claim. $\square$

Let any element $\delta = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbf{Q})$ act on the left on modular symbols $\mathcal{M}_k$ by

$$\delta(P\{\alpha, \beta\}) = P(dX - bY, -cX + aY)\{\delta(\alpha), \delta(\beta)\}.$$

(Until now we had only defined an action of $\mathrm{SL}_2(\mathbf{Z})$ on modular symbols.) For $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbf{Q})$, let

$$\tilde{g} = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \det(g) \cdot g^{-1}. \tag{8.3.1}$$

Note that $\tilde{\tilde{g}} = g$. Also, $\delta.P(X, Y) = (P \circ \tilde{g})(X, Y)$, where we set

$$\tilde{g}(X, Y) = (dX - bY, -cX + aY).$$

Suppose Γ and Δ are as above. Fix a finite set R of representatives for $\Gamma\backslash\Delta$. Let

$$T_\Delta : \mathcal{M}_k(\Gamma) \to \mathcal{M}_k(\Gamma)$$

be the linear map

$$T_\Delta(x) = \sum_{\delta \in R} \delta \cdot x,$$

This map is well defined because if $\gamma \in \Gamma$ and $x \in \mathcal{M}_k(\Gamma)$, then

$$\sum_{\delta\in R} \delta\gamma \cdot x = \sum_{\text{certain } \delta'} \gamma\delta' \cdot x = \sum_{\text{certain } \delta'} \delta' \cdot x = \sum_{\delta \in R} \delta \cdot x,$$

where we have used that $\Delta\Gamma = \Gamma\Delta$, and Γ acts trivially on $\mathcal{M}_k(\Gamma)$.

Let $\Gamma = \Gamma_1(N)$ and $\Delta = \Delta_n$. Then the nth Hecke operator T_n is T_{Δ_n}, and by Lemma 8.3.1,

$$T_n(x) = \sum_{a,b} \sigma_a \begin{pmatrix} a & b \\ 0 & n/a \end{pmatrix} \cdot x,$$

where a, b are as in Lemma 8.3.1.

Given this definition, we can compute the Hecke operators on $M_k(\Gamma_1(N))$ as follows. Write x as a modular symbol $P\{\alpha, \beta\}$, compute $T_n(x)$ as a modular symbol, then convert back to Manin symbols using (many!) continued fractions expansions. This is extremely inefficient, and fortunately Loïc Merel found a much better way, which we now describe (see also [Mer94] and also [Maz73]).

8.3.2 Hecke Operators on Manin Symbols

If S is a subset of $\mathrm{GL}_2(\mathbb{Q})$, let

$$\tilde{S} = \{\tilde{g} : g \in S\}.$$

Also, for any ring R and any subset $S \subset M_2(\mathbb{Z})$, let $R[S]$ denote the free R-module with basis the elements of S, so the elements of $R[S]$ are the finite R-linear combinations of the elements of S.

One of the main theorems of [Mer94] is that for any Γ, Δ as above, if one can find $\sum u_M M \in \mathbb{C}[M_2(\mathbb{Z})]$ and a map

$$\phi : \tilde{\Delta}\,\mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z})$$

that satisfies a complicated list of conditions, then for any Manin symbol $[P, g] \in \mathcal{M}_k(\Gamma)$, we have

$$T_\Delta([P,g]) = \sum_{gM \in \tilde{\Delta}\,\mathrm{SL}_2(\mathbb{Z}) \text{ with } M \in \mathrm{SL}_2(\mathbb{Z})} u_M [\tilde{M} \cdot P,\ \phi(gM)].$$

Merel devotes substantial work to giving examples of ϕ and $\sum u_M M \in \mathbb{C}[M_2(\mathbb{Z})]$ that satisfy all his conditions.

When $\Gamma = \Gamma_1(N)$, the complicated list of conditions becomes simpler. Let $M_2(\mathbb{Z})_n$ be the set of 2×2 matrices with determinant n. An element

$$h = \sum u_M [M] \in \mathbb{C}[M_2(\mathbb{Z})_n]$$

satisfies condition C_n if for every $K \in M_2(\mathbb{Z})_n / \mathrm{SL}_2(\mathbb{Z})$, we have that

$$\sum_{M \in K} u_M([M\infty] - [M0]) = [\infty] - [0] \in \mathbb{C}[P^1(\mathbb{Q})]. \tag{8.3.2}$$

If h satisfies condition C_n, then for any Manin symbol $[P, g] \in M_k(\Gamma_1(N))$, Merel proves that

$$T_n([P,(u,v)]) = \sum_M u_M[P(aX + bY,\, cX + dY), (u,v)M]. \tag{8.3.3}$$

Here $(u, v) \in (\mathbb{Z}/N\mathbb{Z})^2$ corresponds to a coset of $\Gamma_1(N)$ in $\mathrm{SL}_2(\mathbb{Z})$, as in Proposition 8.2.4, and if $(u', v') = (u, v)M \in (\mathbb{Z}/N\mathbb{Z})^2$, and $\gcd(u', v', N) \neq 1$, then we omit the corresponding summand.

For example, we will now check directly that the element

$$h_2 = \left[\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}\right] + \left[\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}\right] + \left[\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}\right] + \left[\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}\right]$$

satisfies condition C_2. We have, as in the proof of Lemma 8.3.1, but using elementary column operations, that

$$\begin{aligned} M_2(\mathbb{Z})_2 / \mathrm{SL}_2(\mathbb{Z}) &= \left\{ \begin{pmatrix} a & 0 \\ b & 2/a \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}) : a = 1, 2 \text{ and } 0 \leq b < 2/a \right\} \\ &= \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}),\ \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}),\ \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}) \right\}. \end{aligned}$$

To verify condition C_2, we consider each of the three elements of $M_2(\mathbb{Z})_2 / \mathrm{SL}_2(\mathbb{Z})$ and check that (8.3.2) holds. We have that

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \in \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}),$$

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \in \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}),$$

and

$$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \in \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \mathrm{SL}_2(\mathbb{Z}).$$

Thus if $K = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right) \mathrm{SL}_2(\mathbb{Z})$, the left sum of (8.3.2) is $[\left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)(\infty)] - [\left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)(0)] = [\infty] - [0]$, as required. If $K = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 2 \end{smallmatrix}\right) \mathrm{SL}_2(\mathbb{Z})$, then the left side of (8.3.2) is

$$[\left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right)(\infty)] - [\left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right)(0)] + [\left(\begin{smallmatrix} 1 & 0 \\ 1 & 2 \end{smallmatrix}\right)(\infty)] - [\left(\begin{smallmatrix} 1 & 0 \\ 1 & 2 \end{smallmatrix}\right)(0)] = [\infty] - [1] + [1] - [0] = [\infty] - [0].$$

Finally, for $K = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right) \mathrm{SL}_2(\mathbb{Z})$ we also have $[\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)(\infty)] - [\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)(0)] = [\infty] - [0]$, as required. Thus by (8.3.3) we can compute T_2 on *any* Manin symbol, by summing over the action of the four matrices $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 1 & 2 \end{smallmatrix}\right)$.

Proposition 8.3.2 (Merel). *The element*

$$\sum_{\substack{a>b\geq 0\\ d>c\geq 0\\ ad-bc=n}} \left[\begin{pmatrix} a & b\\ c & d\end{pmatrix}\right] \in \mathbb{Z}[M_2(\mathbb{Z})_n]$$

satisfies condition C_n.

Merel's proof isn't too difficult, but takes two pages.

Remark 8.3.3. In [Cre97a, §2.4], Cremona discusses the work of Merel and Mazur on Heilbronn matrices in the special cases $\Gamma = \Gamma_0(N)$ and weight 2. He gives a fairly simple proof that the action of T_p on Manin symbols can be computed by summing the action of some set R_p of matrices of determinant p. He then describes the set R_p, and gives an efficient continued fractions algorithm for computing it (but he does not seem to prove that his description of R_p is correct). (Note: My experience is that Cremona's set R_p is significantly smaller than the sets appearing in Merel's paper, but when I've tried to use R_p to do certain more general higher-weight computations that are correct using Merel's sets, they do not work.)

8.3.3 Remarks on Complexity

Merel also gives another family $\mathcal{S}_n$ of matrices that satisfy condition C_n, and he proves that as $n \to \infty$,

$$\#\mathcal{S}_n \sim \frac{12\log(2)}{\pi^2}\cdot \sigma_1(n)\log(n),$$

where $\sigma_1(n)$ is the sum of the divisors of n. Thus for a fixed space $M_k(\Gamma)$ of modular symbols, one can compute the Hecke operator T_n using $O(\sigma_1(n)\log(n))$ arithmetic operations in the base field. Note that we've fixed $M_k(\Gamma)$, so we ignore the linear algebra involved in computation of a presentation; also, adding elements takes a bounded number of field operations when the space is fixed. Thus using Manin symbols the complexity of computing T_p, for p prime, is $O((p+1)\log(p))$ field operations, which is *exponential* in the number of digits of p.

There is a trick of Basmaji (see [Bas96]) for computing a matrix of T_n on $\mathbb{M}_k(\Gamma)$, when n is very large, and it is more efficient than one might naively expect. Basmaji's trick doesn't improve the big-oh complexity for a fixed space, but does improve the complexity by a constant factor of the dimension of $\mathbb{M}_k(\Gamma, \mathbb{Q})$. Suppose we are interested in computing the matrix for T_n for some massive integer n, and that $\mathbb{M}_k(\Gamma, \mathbb{Q})$ as has fairly large dimension. The trick is as follows. Choose, a list

$$x_1 = [P_1, g_1], \ldots, x_r = [P_r, g_r] \in V = \mathbb{M}_k(\Gamma, \mathbb{Q})$$

of Manin symbols such that the map $\Psi : \mathbb{T} \to V^r$ given by

$$t \mapsto (tx_1, \ldots, tx_r)$$

is injective. In practice, it is often possible to do this with r “very small”. Also, we emphasize that V^r is a $\mathbb{Q}$-vector space of dimension $r \cdot \dim(V)$.

Next find Hecke operators T_i, with i small, whose images form a basis for the image of Ψ. Now with the above data precomputed, which only required working with Hecke operators T_i for small i, we are ready to compute T_n with n huge. Compute $y_i = T_n(x_i)$, for each $i = 1, \ldots, r$, which we can compute using Heilbronn matrices since each $x_i = [P_i, g_i]$ is a Manin symbol. We thus obtain $\Psi(T_n) \in V^r$. Since we have precomputed Hecke operators T_j such that $\Psi(T_j)$ generate V^r, we can find a_j such that $\sum a_j \Psi(T_j) = \Psi(T_n)$. Then since Ψ is injective, we have $T_n = \sum a_j T_j$, which gives the full matrix of T_n on $M_k(\Gamma, \mathbb{Q})$.

8.4 Cuspidal Modular Symbols

Let $\mathbb{B}$ be the free abelian group on symbols $\{\alpha\}$, for $\alpha \in \mathbb{P}^1(\mathbb{Q})$, and set

$$\mathbb{B}_k = \mathbb{Z}_{k-2}[X, Y] \otimes \mathbb{B}.$$

Define a left action of $\mathrm{SL}_2(\mathbb{Z})$ on $\mathbb{B}_k$ by

$$g.(P\{\alpha\}) = (g.P)\{g(\alpha)\},$$

for $g \in \mathrm{SL}_2(\mathbb{Z})$. For any finite index subgroup $\Gamma \subset \mathrm{SL}_2(\mathbb{Z})$, let $\mathbb{B}_k(\Gamma)$ be the quotient of $\mathbb{B}_k$ by the relations $x - g.x$ for all $g \in \Gamma$ and by any torsion. Thus $\mathbb{B}_k(\Gamma)$ is a torsion free abelian group.

The *boundary map* is the map

$$b : \mathbb{M}_k(\Gamma) \to \mathbb{B}_k(\Gamma)$$

given by extending the map

$$b(P\{\alpha, \beta\}) = P\{\beta\} - P\{\alpha\}$$

linearly. The space $\mathbb{S}_k(\Gamma)$ of *cuspidal modular symbols* is the kernel

$$\mathbb{S}_k(\Gamma) = \ker(\mathbb{M}_k(\Gamma) \to \mathbb{B}_k(\Gamma)),$$

so we have an exact sequence

$$0 \to \mathbb{S}_k(\Gamma) \to \mathbb{M}_k(\Gamma) \to \mathbb{B}_k(\Gamma).$$

One can prove that when $k > 2$ then this sequence is exact on the right. Also, there is a presentation of $\mathbb{B}_k(\Gamma)$ in terms of “boundary Manin symbols”. [[**TODO: Add this later to the book. It is crucial to add this, since this is something nontrivial that I have in my thesis, and it’s very important to know when implementing.**]]

8.5 The Pairing Between Modular Symbols and Modular Forms

In this section we define a pairing between modular symbols and modular forms, and prove that the Hecke operators respect this pairing. We also define an involution on modular symbols, and study its relationship with the pairing. This pairing is crucial in much that follows, because it gives rise to period maps from modular symbols to certain complex vector spaces.

Fix an integer weight $k \geq 2$ and a finite-index subgroup Γ of $\mathrm{SL}_2(\mathbb{Z})$. Let $M_k(\Gamma)$ denote the space of holomorphic modular forms of weight k for Γ, and $S_k(\Gamma)$ its cuspidal subspace. Following [Mer94, §1.5], let

$$\overline{S}_k(\Gamma) = \{\overline{f} : f \in S_k(\Gamma)\}$$

denote the space of *antiholomorphic* cuspforms. Here $\overline{f}$ is the function on $\mathfrak{h}^*$ given by $\overline{f}(z) = \overline{f(z)}$.

Define a pairing

$$(S_k(\Gamma) \oplus \overline{S}_k(\Gamma)) \times \mathbb{M}_k(\Gamma) \to \mathbb{C} \tag{8.5.1}$$

by

$$\langle (f_1, f_2), P\{\alpha, \beta\} \rangle = \int_\alpha^\beta f_1(z) P(z, 1) dz + \int_\alpha^\beta f_2(z) P(\overline{z}, 1) d\overline{z},$$

and extending linearly. Here the integral is a complex path integral along a great circle (or vertical line) from α to β (so, e.g., write $z(t) = x(t) + iy(t)$, where $(x(t), y(t))$ traces out the path, and consider two real integrals; see any introductory book on complex analysis for more details).

The integration pairing is well defined, which means that if we replace $P\{\alpha, \beta\}$ by an equivalent modular symbols (equivalent modulo the left action of Γ), then the integral is the same. This follows from the change of variables formulas for integration and the fact that $f_1 \in S_k(\Gamma)$ and $f_2 \in \overline{S}_k(\Gamma)$. For example, if $k = 2$, $g \in \Gamma$ and $f \in S_k(\Gamma)$, then

$$\begin{aligned}
\langle f, g\{\alpha, \beta\} \rangle &= \langle f, \{g(\alpha), g(\beta)\} \rangle \\
&= \int_{g(\alpha)}^{g(\beta)} f(z) dz \\
&= \int_\alpha^\beta f(g(z)) dg(z) \\
&= \int_\alpha^\beta f(z) dz = \langle f, \{\alpha, \beta\} \rangle,
\end{aligned}$$

where in the last step we use that f is a weight 2 modular form.

Remark 8.5.1. The integration pairing is related to special values of L-functions. The L-function attached to a cusp form $f = \sum a_n q^n \in S_k(\Gamma_1(N))$ is

$$L(f, s) = (2\pi)^s \Gamma(s)^{-1} \int_0^\infty f(it) t^s \frac{dt}{t} \tag{8.5.2}$$

Note that one can show that $L(f,s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$ by switching the order of summation and integration, which is justified using standard estimates on $|a_n|$ (see, e.g., [Kna92, §VIII.5]).

For each integer j with $1 \leq j \leq k-1$, we have setting $s = j$ and making the change of variables $t \mapsto -it$ in (8.5.2), that

$$L(f,j) = \frac{(-2\pi i)^j}{(j-1)!} \cdot \left\langle f,\ X^{j-1}Y^{k-2-(j-1)}\{0,\infty\}\right\rangle. \qquad (8.5.3)$$

The integers j as above are called *critical integers*, and when f is an eigenform, they have deep conjectural significance. We will discuss tricks to efficiently compute $L(f,j)$ later in this book.

Theorem 8.5.2 (Shokoruv). *The pairing $\langle\cdot\,,\,\cdot\rangle$ is nondegenerate when restricted to cuspidal modular symbols:*

$$\langle\cdot\,,\,\cdot\rangle : (S_k(\Gamma) \oplus \overline{S}_k(\Gamma)) \times \mathbb{S}_k(\Gamma) \to \mathbb{C}.$$

The pairing is also compatible with Hecke operators. Before proving this, we define an action of *Hecke operators* on $M_k(\Gamma_1(N))$ and on $\overline{S}_k(\Gamma_1(N))$. The definition is very similar to the one we gave in Section 2.4 for modular forms of level 1. For a positive integer n, let R_n be a set of coset representatives for $\Gamma_1(N)\backslash\Delta_n$ from Lemma 8.3.1. For any $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Q})$ and $f \in M_k(\Gamma_1(N))$ set

$$f|[\gamma]_k = \det(\gamma)^{k-1}(cz+d)^{-k}f(\gamma(z)).$$

Also, for $f \in \overline{S}_k(\Gamma_1(N))$, set

$$f|[\gamma]_k' = \det(\gamma)^{k-1}(c\overline{z}+d)^{-k}f(\gamma(z)).$$

Then for $f \in M_k(\Gamma_1(N))$,

$$T_n(f) = \sum_{\gamma \in R_n} f|[\gamma]_k$$

and for $f \in \overline{S}_k(\Gamma_1(N))$,

$$T_n(f) = \sum_{\gamma \in R_n} f|[\gamma]_k'.$$

This agrees with the definition from 2.4 when $N = 1$.

Remark 8.5.3. If Γ is an arbitrary finite index subgroup of $\mathrm{SL}_2(\mathbb{Z})$, then we can define operators T_Δ on $M_k(\Gamma)$ for any Δ with $\Delta\Gamma = \Gamma\Delta = \Delta$ and $\Gamma\backslash\Delta$ finite. For concreteness we do not do the general case here or in the theorem below, but the proof is exactly the same (see [Mer94, §1.5]).

Finally we prove the promised Hecke compatibility of the pairing. This proof should convince you that the definition of modular symbols is sensible, in that they are "natural" expressions to integrate against modular forms.

Theorem 8.5.4. *If $f = (f_1, f_2) \in S_k(\Gamma_1(N)) \oplus \overline{S}_k(\Gamma_1(N))$ and $x \in \mathbb{M}_k(\Gamma_1(N))$, then for any n,*

$$\langle T_n(f), x\rangle = \langle f, T_n(x)\rangle.$$

Proof. We exactly follow [Mer94, §2.1], and will only prove the theorem when $f = f_1 \in S_k(\Gamma_1(N))$, the proof in the general case being the same.

Let $\alpha, \beta \in \mathbb{P}^1(\mathbb{Q})$, $P \in \mathbb{Z}_{k-2}[X, Y]$, and for $g = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Q})$, set $j(g, z) = (cz+d)$. Let n be any positive integer, and let R_n be a set of coset representatives for $\Gamma_1(N)\backslash\Delta_n$ from Lemma 8.3.1.

We have

$$\begin{aligned}\langle T_n(f), P\{\alpha, \beta\}\rangle &= \int_\alpha^\beta T_n(f)P(z, 1)dz \\ &= \sum_{\delta\in R} \int_\alpha^\beta \det(\delta)^{k-1} f(\delta(z)) j(\delta, z)^{-k} P(z, 1)dz.\end{aligned}$$

Now for each summand corresponding to the $\delta \in R$, make the change of variables $u = \delta z$. Thus we make $\#R$ change of variables. Also, recall the notation from (8.3.1), which we will use below.

$$\begin{aligned}\langle T_n(f), P\{\alpha, \beta\}\rangle &= \sum_{\delta\in R} \int_{\delta(\alpha)}^{\delta(\beta)} \det(\delta)^{k-1} f(u) j(\delta, \delta^{-1}(u))^{-k} P(\delta^{-1}(u), 1) d(\delta^{-1}(u)) \\ &= \sum_{\delta\in R} \int_{\delta(\alpha)}^{\delta(\beta)} \det(\delta)^{k-1} f(u) j(\tilde{\delta}, u)^{k} \det(\delta)^{-k} P(\tilde{\delta}(u), 1) \frac{\det(\delta)du}{j(\tilde{\delta}, u)^2} \\ &= \sum_{\delta\in R} \int_{\delta(\alpha)}^{\delta(\beta)} f(u) j(\tilde{\delta}, u)^{k-2} P(\tilde{\delta}(u), 1)du \\ &= \sum_{\delta\in R} \int_{\delta(\alpha)}^{\delta(\beta)} f(u) \cdot ((\delta.P)(u, 1))du \\ &= \langle f, T_n(P\{\alpha, \beta\})\rangle.\end{aligned}$$

The second equality is the trickiest. First, note that $\delta^{-1}(u) = \tilde{\delta}(u)$, since a linear fractional transformation is unchanged by a nonzero rescaling of a matrix that induces it. Thus by the quotient rule, using that $\tilde{\delta}$ has determinant $\det(\delta)$, we see that

$$d(\delta^{-1}(u)) = \frac{\det(\delta)du}{j(\tilde{\delta}, u)^2}.$$

The other part of the second equality asserts that

$$j(\delta, \delta^{-1}(u))^{-k} P(\delta^{-1}(u), 1) = j(\tilde{\delta}, u)^{k} \det(\delta)^{-k} P(\tilde{\delta}(u), 1). \tag{8.5.4}$$

From the definitions, and again using that $\delta^{-1}(u) = \tilde{\delta}(u)$, we see that

$$j(\delta, \delta^{-1}(u)) = \frac{\det(\delta)}{j(\tilde{\delta}, u)},$$

which proves that (8.5.4) holds. In the third equality, we use that

$$(\delta.P)(u,1) = j(\tilde{\delta},u)^{k-2}P(\tilde{\delta}(u),1).$$

To see this, note that $P(X,Y) = P(X/Y,1)\cdot Y^{k-2}$. Using this we see that

$$\begin{aligned}(\delta.P)(X,Y) &= (P\circ\tilde{\delta})(X,Y)\\ &= P\left(\tilde{\delta}\left(\frac{X}{Y}\right),1\right)\cdot\left(-c\cdot\frac{X}{Y}+a\right)^{k-2}\cdot Y^{k-2}.\end{aligned}$$

Now substituting $(u,1)$ for $(X,1)$, we see that

$$(\delta.P)(u,1) = P(\tilde{\delta}(u),1)\cdot(-cu+a)^{k-2},$$

as required. □

Remark 8.5.5. The theorem is true more generally for any Γ and any operator T_Δ, via the same proof.

Suppose that Γ is finite index subgroup of $\mathrm{SL}_2(\mathbb{Z})$ such that if $\eta = \left(\begin{smallmatrix}-1&0\\0&1\end{smallmatrix}\right)$, then

$$\eta\Gamma\eta = \Gamma.$$

For example, $\Gamma = \Gamma_1(N)$ satisfies this condition. There is an involution ι^* on $\mathbb{M}_k(\Gamma)$ given by

$$\iota^*(P(X,Y)\{\alpha,\beta\}) = -P(X,-Y)\{-\alpha,-\beta\}, \tag{8.5.5}$$

which we call the *star involution*. On Manin symbols, ι^* it is

$$\iota^*[P,(u,v)] = -[P(-X,Y),(-u,v)].$$

Let $\mathbb{S}_k(\Gamma)^+$ be the $+1$ eigenspace for ι^* and $\mathbb{S}_k(\Gamma)^-$ the -1 eigenspace. There is also a map ι on modular forms, which is adjoint to ι^*.

Remark 8.5.6 (WARNING). Notice the $-$ sign in front of $-P(X,-Y)\{-\alpha,-\beta\}$ in (8.5.5). This sign is missing in [Cre97a], which confused me. Thus the $+1$ quotient in MAGMA is the quotient where η acts as -1. (This is a mistake.)

We now state the final result about the pairing, which explains how modular symbols and modular forms are related.

Theorem 8.5.7. *The pairing $\langle\cdot\,,\cdot\rangle$ restricts to give nondegenerate Hecke compatible bilinear pairings*

$$\mathbb{S}_k(\Gamma)^+\times S_k(\Gamma)\to\mathbb{C}\qquad\textit{and}\qquad\mathbb{S}_k(\Gamma)^-\times\overline{S}_k(\Gamma)\to\mathbb{C}.$$

In light of the Peterson inner product, the above theorem implies that there is a canonical isomorphism of $\mathbb{T}'$-modules

$$\mathbb{S}_k(\Gamma,\mathbb{C})^+\cong S_k(\Gamma),$$

where $\mathbb{T}'$ is the anemic Hecke algebra, i.e., the subring of $\mathbb{T}$ generated by Hecke operators T_n with $\gcd(n,N)=1$. In fact, one can prove, e.g., using Eichler-Shimura cohomology, that there is a non-canonical isomorphism over the full Hecke algebra

$$\mathbb{M}_k(\Gamma,\mathbb{C})\cong M_k(\Gamma)\oplus\overline{S}_k(\Gamma).$$

8.6 Explicitly Computing $\mathbb{M}_k(\Gamma_0(N))$

In this section we explicitly compute $\mathbb{M}_k(\Gamma_0(N))$ for various k and N. We represent Manin symbols for $\Gamma_0(N)$ as triples (i, u, v), where $(u, v) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, and (i, u, v) corresponds to $[X^i Y^{k-2-i}, (u, v)]$ in the usual notation. Also, recall that (u, v) corresponds to the right coset in $\Gamma_0(N)\backslash \mathrm{SL}_2(\mathbb{Z})$ that contains a matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $(u, v) \equiv (c, d)$ as elements of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, i.e., up to rescaling by an element of $(\mathbb{Z}/N\mathbb{Z})^*$.

8.6.1 Computing $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$

In this section we give an algorithm to compute a canonical representative for each element of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. This algorithm is extremely important because modular symbols implementations call a huge number of times. A more naive approach would be to store all pairs $(u, v) \in (\mathbb{Z}/N\mathbb{Z})^2$, and a fixed reduced representative, but this wastes a huge amount of memory. For example, if $N = 10^5$, we would have to store an array of

$$(10^5 \cdot 10^5)/10^6 = 10000 \text{ million integers,}$$

which is many terabytes.

Another approach to enumerating $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ is described at the end of [Cre97a, §2.2]. We use that it is easy to test whether two pairs $(u_0, v_0), (u_1, v_1)$ define the same element of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$; they do if and only if we have equality of cross terms $u_0 v_1 = v_0 u_1 \pmod{N}$ (see [Cre97a, Prop. 2.2.1]). So we list elements $(1, a)$ for $a = 0, 1, \ldots, N-1$, then elements (d, a) for $d \mid N$ and $a = 1, \ldots, N-1$, but checking each time we add a new element to our list whether we have already seen it. Unfortunately, given a random pair (u, v), which is something we encounter *very frequently* in practice, we have to compare (u, v) with each element of the list to find our chosen equivalent representative in $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. This is very expensive, since it requires a linear search through the list, hence takes time at least $O(n)$, where n is the number of elements of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. To get around this Cremona says he "used a simple 'hashing' system, so that given any particular symbole (c, d) we could quickly determine to which symbol in our standard list it is equivalent." (He doesn't say what hashing system he uses.)

Instead of either of the above methods, we use the following algorithm, which finds a canonical representative for each element of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. With this algorithm in hand, given an arbitrary (u, v), we first find the canonical equivalent elements (u', v'), then search a sorted lists of all canonical pairs, which takes time $O(\log(n))$, where $n = \#\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.

Algorithm 8.6.1 (Reduce). INPUT: Integers u and v, and a positive integer N. OUTPUT: If possible, this algorithm outputs a pair u_0, v_0 such that $(u, v) \equiv (u_0, v_0)$ as elements of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ and $s \in \mathbb{Z}$ such that $(u, v) = (su_0, sv_0) \pmod{\mathbb{Z}/n\mathbb{Z}}$. Moreover, the element (u_0, v_0) does not depend on the class of (u, v), i.e., for any s with $\gcd(N, s) = 1$ the input (su, sv) also outputs (u_0, v_0). If (u, v) is not in $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, this algorithm outputs $(0,0), 0$.

THE ALGORITHM: In the following algorithm, a%N denotes the residue of a modulo N that satisfies $0 \leq a < N$.

1. Reduce both u and v modulo N:

```
u = u % N;    v = v % N
```

2. Deal with the easy special case when $u = 0$, using that $(0, v) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ if and only if $\gcd(v, N) = 1$:

```
if u == 0:
    u0 = 0
    if gcd(v,N) == 1:
        v0 = 1
    else:
        v0 = 0
    s = v
    return (u0,v0), s
```

3. Compute $g = \gcd(u, N)$ and $s, t \in \mathbb{Z}$ such that $g = su + tN$:

```
g, s, t = XGCD(u, N)
s = s % N
```

4. We have $\gcd(u, v, N) = \gcd(g, v)$, so if $\gcd(g, v) > 1$, then $(u, v) \notin \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.

```
if gcd(g, v) != 1:
    return (0,0), 0
```

5. Now $g = su + tN$, so we may think of s as "pseudo-inverse" of $u \pmod N$, in the sense that su is as close as possible to being 1 modulo N. Note that since $g \mid u$, changing s modulo N/g does not change $su \pmod N$. We can adjust s modulo N/g so it is coprime to N. (This is because $1 = su/g + tN/g$, so s is a unit mod N/g, and the map $(\mathbb{Z}/N\mathbb{Z})^* \to (\mathbb{Z}/(N/g)\mathbb{Z})^*$ is surjective, e.g., as we saw in the proof of Algorithm 4.6.1.)

```
if g != 1:
    d = N/g
    while gcd(s,N) != 1:
        s = (s+d) % N
```

6. Multiply (u,v) by s, replacing (u,v) by the equivalent element (g,sv) of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.

```
u = g
v = (s*v) % N
```

7. Next we find the unique pair (g,v') equivalent to (g,v) that minimizes v. To do this, we note that if $1 \neq t \in (\mathbb{Z}/N\mathbb{Z})^*$ and $tg \equiv g \pmod{N}$, then $(t-1)g \equiv 0 \pmod{N}$, so $t-1 = kN/g$ for some k with $1 \leq k \leq g-1$. Then for $t = 1+kN/g$ coprime to N, we have $(gt, vt) = (g, v+kvN/g)$. The following part of the algorithm computes all $(g, v+kvN/g)$ pairs and picks out the one that minimizes the least nonnegative residue of vt modulo N:

```
min_v = v; min_t = 1
if g != 1:
    Ng = N/g
    vNg = (v*Ng) % N
    t = 1
    for k in xrange(1,g):      # for k satisfying 1<=k<g.
        v = (v + vNg) % N
        t = (t + Ng) % N
        if v < min_v and gcd(t,N) == 1:
            min_v = v; min_t = t
s = s * min_t
```

8. The s that we have computed in the above steps multiples the input (u,v) to give the output (u_0,v_0). Thus we have to invert it, since the output scalar is supposed to multiply (u_0,v_0) to give (u,v).

```
s = inverse_mod(s, N)
return (u,min_v), s
```

Remark 8.6.2. Allan Steel and the author jointly came up with Algorithm 8.6.1.

Remark 8.6.3. There might be an even better algorithm that uses that

$$\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z}) \cong \prod_{p \mid N} \mathbb{P}^1(\mathbb{Z}/p^{\nu_p}\mathbb{Z}).$$

This would also use that it is relatively easy to enumerate the elements of $\mathbb{P}^1(\mathbb{Z}/p^n\mathbb{Z})$ for a prime power p^n. I have not thought this through.

Algorithm 8.6.4 (List $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$). *This algorithm makes a sorted list of the distinct canonical representatives of $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, as output by Algorithm 8.6.1.*

INPUT: An integer $N > 1$.

OUTPUT: Sorted list of canonical representatives for $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$.

1. *First we make a list of the canonical representatives of enough pairs (c, d) to fill up $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. In the following code, we name Algorithm 8.6.1* `p1_normalize`.

```
lst = [(0,1), (1,0)]
for c in range(1,N):          # iterate c such that 1 <= c < N:
    lst.append((1,c))
    g = gcd(c,N)
    if g > 1:
        u, v, s = p1_normalize(c, 1,   N)
        lst.append((u,v))
        if g == c:              # so c is a divisor
            for d in xrange(2,N):              # 2 <= d < N
                if gcd(d,N) > 1 and gcd(d,c) == 1:
                    u,v,s = p1_normalize(c, d,   N)
                    lst.append((u,v))
```

2. *Next we sort the list of canonical pairs, then with one pass through the list delete any duplicates (or use the following Python code, which is slightly different).*

```
lst = list(set(lst))    # Python trick remove duplicates.
lst.sort()
```

8.6.2 Examples of Computation of $\mathbb{M}_k(\Gamma_0(N))$

In this section, we compute $\mathbb{M}_k(\Gamma_0(N))$ explicitly in a few cases.

Example 8.6.5. We compute $V = \mathbb{M}_4(\Gamma_0(1))$. Because $S_k(\Gamma_0(1)) = 0$, and $M_k(\Gamma_0(1)) = \mathbb{C}E_4$, we expect V to have dimension 1, and for the Hecke operator T_n to have eigenvalues the sum $\sigma_3(n)$ of the cubes of positive divisors of n.

The Manin symbols are

$$x_0 = (0, 0, 0), \quad x_1 = (1, 0, 0), \quad x_2 = (2, 0, 0).$$

The relation matrix is

$$\left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 2 & -2 & 2 \\ 1 & -1 & 1 \\ 2 & -2 & 2 \end{array}\right),$$

where the first 2 rows correspond to S relations and the second two to T relations. Note that we don't include all S relations, since it is obvious that some are redundant, e.g., $x + xS = 0$ and $(xS) + (xS)S = xS + x = 0$ are the same since S has order 2. (It's not clear to me what is going on with T relations when $k > 2$, though in this example two of the three T relations are redundant.)

The echelon form of the relation matrix is

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

where we've deleted the zero rows from the bottom. Thus we may replace the above complicated list of relations with the following simpler list of relations:

$$\begin{aligned} x_0 + x_2 &= 0 \\ x_1 &= 0 \end{aligned}$$

from which we immediately read off that the second generator x_1 is 0 and $x_0 = -x_2$. Thus $\mathbb{M}_4(\Gamma_0(1))$ has dimension 1, with basis the equivalence class of x_2 (or of x_0).

Next we compute the Hecke operator T_2 on $\mathbb{M}_4(\Gamma_0(1))$. The Heilbronn matrices of determinant 2 from Proposition 8.3.2 are

$$h_0 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad h_1 = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad h_3 = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix},$$

To compute T_2, we apply each of these matrices to x_0, then reduce modulo the relations. We have

$$\begin{aligned} x_2.\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} &= [X^2, (0,0)].\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} x_2 \\ x_2.\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} &= [X^2, (0,0)] = x_2 \\ x_2.\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} &= [(2X)^2, (0,0)] = 4x_2 \\ x_2.\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} &= [(2X+1)^2, (0,0)] = x_0 + 4x_1 + 4x_2 \sim 3x_2 \end{aligned}$$

Summing we see that $T_2(x_2) \sim 9x_2$ in $\mathbb{M}_4(\Gamma_0(1))$. Notice that

$$9 = 1^3 + 2^3 = \sigma_3(2).$$

The Merel Heilbronn matrices of determinant 3 from Proposition 8.3.2 are

$$\begin{aligned} &h_0 = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, \quad h_1 = \begin{pmatrix} 1 & 0 \\ 1 & 3 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 1 & 0 \\ 2 & 3 \end{pmatrix}, \quad h_3 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \\ &h_4 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad h_5 = \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix}, \quad h_6 = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

We have

$$\begin{aligned}
x_2.\begin{pmatrix}1 & 0\\ 0 & 3\end{pmatrix} &= [X^2,(0,0)].\begin{pmatrix}1 & 0\\ 0 & 3\end{pmatrix} = x_2\\
x_2.\begin{pmatrix}1 & 0\\ 1 & 3\end{pmatrix} &= [X^2,(0,0)] = x_2\\
x_2.\begin{pmatrix}1 & 0\\ 2 & 3\end{pmatrix} &= [X^2,(0,0)] = x_2\\
x_2.\begin{pmatrix}2 & 1\\ 2 & 2\end{pmatrix} &= [(2X+1)^2,(0,0)] = x_0+4x_1+4x_2 \sim 3x_2\\
x_2.\begin{pmatrix}3 & 0\\ 0 & 1\end{pmatrix} &= [(3X)^2,(0,0)] = 9x_2\\
x_2.\begin{pmatrix}3 & 1\\ 0 & 1\end{pmatrix} &= [(3X+1)^2,(0,0)] = x_0+6x_1+9x_2 \sim 8x_2\\
x_2.\begin{pmatrix}3 & 2\\ 0 & 1\end{pmatrix} &= [(3X+2)^2,(0,0)] = 4x_0+12x_1+9x_2 \sim 5x_2
\end{aligned}$$

Summing we see that

$$T_3(x_2) \sim x_2+x_2+x_2+3x_2+9x_2+8x_2+5x_2 = 28x_2.$$

Notice that

$$28 = 1^3+3^3 = \sigma_3(3).$$

Example 8.6.6. Next we compute $\mathbb{M}_2(\Gamma_0(11))$ explicitly. The Manin symbol generators are

$$x_0=(1,0),\quad x_1=(1,0),\quad x_2=(1,1),\quad x_3=(1,2),\quad x_4=(1,3),\quad x_5=(1,4),$$

$$x_6=(1,5),\quad x_7=(1,6),\quad x_8=(1,7),\quad x_9=(1,8),\quad x_{10}=(1,9),\quad x_{11}=(1,10).$$

The relation matrix is as follows, where the S relations are above the line, and the T relations are below it.

$$\left(\begin{array}{cccccccccccc}
1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0\\
\hline
1&1&0&0&0&0&0&0&0&0&0&1\\
0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&1&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&1&0&1&0&0
\end{array}\right)$$

In weight 2, two out of three T-relations are redundant, so we do not include them. The reduced row echelon form of the relation matrix is

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

From the echelon form we immediaely see that every symbol is equivalent to a combination of $x_1 = (1,0)$, $x_9 = (1,8)$, and $x_{10} = (1,9)$. (Notice that columns 1, 9, and 10 are the pivot columns, where we index columns starting at 0.) Explicitly, if (a,b,c) is the ith row of the following matrix, then $x_i = ax_1 + bx_9 + cx_{10}$:

$$\begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

To compute T_2, we apply each of the Heilbronn matrices of determinant 2 from Proposition 8.3.2 to x_1, then to x_9, and finally to x_{10}. The matrices are as in Example 8.6.5 above. We have

$$T_2(x_1) = 3(1,0) + (1,6) \sim 3x_1 - x_{10}.$$

Applying T_2 to $x_9 = (1,8)$, we get

$$T_2(x_9) = (1,3) + (1,4) + (1,5) + (1,10) \sim -2x_9$$

Applying T_2 to $x_{10} = (1,9)$, we get

$$T_2(x_{10}) = (1,4) + (1,5) + (1,7) + (1,10) \sim -x_1 - 2x_{10}.$$

Thus the matrix of T_2 with respect to this basis is

$$T_2 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & -2 & 0 \\ -1 & 0 & -2 \end{pmatrix},$$

where we write the matrix as an operator on the left on vectors written in terms of x_1, x_9, and x_{10}. The matrix T_2 has characteristic polynomial

$$(x-3)(x+2)^2.$$

The $(x-3)$ factor corresponds to the weight 2 Eisenstein series, and the $x+2$ factor corresponds to the elliptic curve $E = X_0(11)$, which has

$$a_2 = -2 = 2 + 1 - \#E(\mathbb{F}_2).$$

We have

$$\begin{aligned}
T_3(x_1) &= 4(1,0) + (1,4) + (1,6) + (1,8) \sim 4x_1 - x_{10} \\
T_3(x_9) &= (1,2) + (1,3) + (1,4) + (1,5) + (1,7) + 2(1,10) \sim -x_9 \\
T_3(x_{10}) &= (0,1) + (1,0) + (1,2) + (1,3) + (1,5) + (1,6) + (1,7) \sim -x_{10},
\end{aligned}$$

so

$$T_3 = \begin{pmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & -1 \end{pmatrix}.$$

The characteristic polynomial of T_3 is $(x-4)(x+1)^2$.

Example 8.6.7. In this example, we compute $\mathbb{M}_6(\Gamma_0(3))$, which illustrates both big weight and nontrivial level. We have the following generating Manin symbols:

$$\begin{aligned}
x_0 &= [XY^4, (0,1)], & x_1 &= [XY^4, (1,0)] \\
x_2 &= [XY^4, (1,1)], & x_3 &= [XY^4, (1,2)] \\
x_4 &= [XY^3, (0,1)], & x_5 &= [XY^3, (1,0)] \\
x_6 &= [XY^3, (1,1)], & x_7 &= [XY^3, (1,2)] \\
x_8 &= [X^2Y^2, (0,1)], & x_9 &= [X^2Y^2, (1,0)] \\
x_{10} &= [X^2Y^2, (1,1)], & x_{11} &= [X^2Y^2, (1,2)] \\
x_{12} &= [X^3Y, (0,1)], & x_{13} &= [X^3Y, (1,0)] \\
x_{14} &= [X^3Y, (1,1)], & x_{15} &= [X^3Y, (1,2)] \\
x_{16} &= [X^4Y, (0,1)], & x_{17} &= [X^4Y, (1,0)] \\
x_{18} &= [X^4Y, (1,1)], & x_{19} &= [X^4Y, (1,2)]
\end{aligned}$$

The relation matrix is already very large for $\mathcal{M}_6(\Gamma_0(3))$ follows, where the S

relations are before the line and the T relations after it:

$$
\left(\begin{array}{cccccccccccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&-1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&-1&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&-1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\
\hline
1&0&0&1&0&0&0&-4&0&0&0&6&0&0&0&-4&0&1&0&1\\
1&1&0&0&-4&0&0&0&6&0&0&0&-4&0&0&0&1&0&0&1\\
0&0&2&0&0&0&-4&0&0&0&6&0&0&0&-4&0&0&0&2&0\\
0&1&0&1&0&-4&0&0&0&6&0&0&0&-4&0&0&1&1&0&0\\
0&0&0&1&1&0&0&-3&0&0&0&3&0&-1&0&-1&0&1&0&0\\
1&0&0&0&-3&1&0&0&3&0&0&0&-1&0&0&-1&0&0&0&1\\
0&0&1&0&0&0&-2&0&0&0&3&0&0&0&-2&0&0&0&1&0\\
0&1&0&0&0&-3&0&1&0&3&0&0&-1&-1&0&0&1&0&0&0\\
0&0&0&1&0&0&0&-2&1&1&0&1&0&-2&0&0&0&1&0&0\\
1&0&0&0&-2&0&0&0&1&1&0&1&0&0&0&-2&0&0&0&1\\
0&0&1&0&0&0&-2&0&0&0&3&0&0&0&-2&0&0&0&1&0\\
0&1&0&0&0&-2&0&0&1&1&0&1&-2&0&0&0&1&0&0&0\\
0&0&0&1&0&-1&0&-1&0&3&0&0&1&-3&0&0&0&1&0&0\\
1&0&0&0&-1&0&0&-1&0&0&0&3&0&1&0&-3&0&0&0&1\\
0&0&1&0&0&0&-2&0&0&0&3&0&0&0&-2&0&0&0&1&0\\
0&1&0&0&-1&-1&0&0&3&0&0&0&-3&0&0&1&1&0&0&0\\
0&1&0&1&0&-4&0&0&0&6&0&0&0&-4&0&0&1&1&0&0\\
1&0&0&1&0&0&0&-4&0&0&0&6&0&0&0&-4&0&1&0&1\\
0&0&2&0&0&0&-4&0&0&0&6&0&0&0&-4&0&0&0&2&0\\
1&1&0&0&-4&0&0&0&6&0&0&0&-4&0&0&0&1&0&0&1
\end{array}\right)
$$

The reduced row echelon form of the relations matrix, with zero rows removed:

$$
\left(\begin{array}{cccccccccccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&3/16&-3/16\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&-1/16&1/16\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1/2&-5/16&-3/16\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&-1/2&3/16&5/16\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&-1/6&1/12&1/12\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1/6&-1/12&-1/12\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1/4&-1/4\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&-1/4&1/4\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&-1/16&1/16\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&3/16&-3/16\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1/2&3/16&5/16\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1/2&-5/16&-3/16
\end{array}\right)
$$

Since these relations are equivalent to the original relations, we see quite clearly how $x_0, \ldots, x_{15}$ can be expressed in terms of x_{16}, x_{17}, x_{18}, and x_{19}. Thus $\mathbb{M}_6(\Gamma_0(3))$ has dimension 4. For example,

$$x_{15} \sim \frac{1}{2}x_{17} - \frac{5}{16}x_{18} - \frac{3}{16}x_{19}.$$

Notice that the number of relations is already quite large. It is perhaps surprisingy how complicated the presentation is for $\mathbb{M}_6(\Gamma_0(3))$. Because there are denominators in the relations, the above calculation is only a computation of $\mathbb{M}_6(\Gamma_0(3), \mathbb{Q})$. Computing $\mathbb{M}_6(\Gamma_0(3), \mathbb{Z})$ requires computation of a $\mathbb{Z}$-basis for the kernel of the relation matrix, which could be accomplished via, e.g., Hermite normal form or LLL reduction.

As before, we find that with respect to the basis x_{16}, x_{17}, x_{18}, and x_{19}, that

$$T_2 = \begin{pmatrix} 33 & 0 & 0 & 0 \\ 3 & 6 & 12 & 12 \\ -3/2 & 27/2 & 15/2 & 27/2 \\ -3/2 & 27/2 & 27/2 & 15/2 \end{pmatrix}$$

Notice that there are denominators in the matrix for T_2 with respect to this basis. It is clear from the definition of T_2 acting on Manin symbols that T_2 preserves the $\mathbb{Z}$-module $\mathbb{M}_6(\Gamma_0(3))$, so there is some basis for $\mathbb{M}_6(\Gamma_0(3))$ such that T_2 is given by an integer matrix. Thus the characteristic polynomial f_2 of T_2 will have integer coefficients; indeed,

$$f_2 = (x - 33)^2 \cdot (x + 6)^2.$$

Note the factor of 33, which comes from the two images of the Eisenstein series E_4 of level 1. The factor $x + 6$ comes from a cusp form

$$g = q - 6q^2 + \cdots \in S_6(\Gamma_0(3)).$$

By computing more Hecke operators T_n, we can find more coefficients of g. For example, the charpoly of T_3 is $(x-1)(x-243)(x-9)^2$, and the matrix of T_5 is

$$T_5 = \begin{pmatrix} 3126 & 0 & 0 & 0 \\ 240 & 966 & 960 & 960 \\ -120 & 1080 & 1086 & 1080 \\ -120 & 1080 & 1080 & 1086 \end{pmatrix},$$

which has characteristic polynomial

$$f_5 = (x - 3126)^2 (x - 6)^2.$$

The matrix of T_7 is

$$T_7 = \begin{pmatrix} 16808 & 0 & 0 & 0 \\ 1296 & 5144 & 5184 & 5184 \\ -648 & 5832 & 5792 & 5832 \\ -648 & 5832 & 5832 & 5792 \end{pmatrix},$$

with characteristic polynomial

$$f_7 = (x - 16808)^2(x + 40)^2.$$

One can put this information together to deduce that

$$g = q - 6q^2 + 9q^3 + 4q^4 + 6q^5 - 54q^6 - 40q^7 + \cdots .$$

Example 8.6.8. The relation matrix for $\mathcal{M}_2(\Gamma_0(43))$ is

$$\left(\begin{array}{cccccccccccccccccccccccccccccccccccccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

Reducing, one computes a presentation for $\mathbb{M}_2(\Gamma_0(43))$, which has dimension 7. With respect to the symbols

$$x_1 = (1, 0), \quad x_{32} = (1, 31), \quad x_{33} = (1, 32),$$
$$x_{39} = (1, 38), \quad x_{40} = (1, 39), \quad x_{41} = (1, 40), \quad x_{42} = (1, 41),$$

the matrix of T_2 is

$$T_2 = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -2 & -1 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 2 & 1 \\ 0 & 0 & -1 & -1 & -1 & -2 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 & -1 \end{pmatrix},$$

which has characteristic polynomial

$$(x - 3)(x + 2)^2(x^2 - 2)^2.$$

There is one Eisenstein series, and three cusp forms with $a_2 = -2$ and $a_2 = \pm\sqrt{2}$.

Example 8.6.9. In this example we discuss computation of $\mathbb{M}_2(\Gamma_0(2004), \mathbb{Q})$, without explicitly writing down the matrices, which are huge. First we make a list of the

$$4032 = (2^2 + 2) \cdot (3 + 1) \cdot (167 + 1)$$

elements $(a, b) \in \mathbb{P}^1(\mathbb{Z}/2004\mathbb{Z})$ using Algorithm 8.6.1. This list looks like this:

$$x_0 = (0, 1), (1, 0), (1, 1), (1, 2), \ldots, (501, 7), (668, 1), (668, 3), (668, 5), x_{4032} = (1002, 1)$$

For each of the symbols x_i, we consider the S and T relations. Ignoring the redundant relations, we find 2016 S-relations and 1344 T-relations. It is simple to quotient out by the S-relations, e.g., by identifying x_i with $-x_iS = -x_j$ for some j (or setting $x_i = 0$ if $x_iS = x_i$). Once we've quotiented out by the S relations, we take the *image* of all of the 1344 T relations modulo the S-relations and quotient out by those relations. Because S and T do not commutate, we can not only quotient out by T relations $x_i + x_iT + x_iT^2 = 0$ where the x_i are the basis after quotienting out by the S relations. We find that the relation matrix has rank 3359, so $\mathbb{M}_2(\Gamma_0(2004), \mathbb{Q})$ has dimension 673.

If we instead compute the quotient $\mathbb{M}_2(\Gamma_0(2004), \mathbb{Q})^+$ of $\mathbb{M}_2(\Gamma_0(2004), \mathbb{Q})$ by the subspace of elements $x - \eta^*(x)$, we include relations $x_i + x_iI = 0$, where $I = \left(\begin{smallmatrix}-1 & 0\\ 0 & 1\end{smallmatrix}\right)$. There are now 2016 S relations, 2024 I relations, and 1344 T relations. Again, it is almost trivial to quotient out by the S relations by identifying x_i and $-x_iS$. We then take the image of all 2024 I relations modulo the S relations, and again directly quotient out by the I-relations by identifying $[x_i]$ with $-[x_iI] = -[x_j]$ for some j, where by $[x_i]$ we mean the class of x_i modulo the S relations. Finally, we quotient out by the 1344 T relations, which involves sparse Gauss elimination on a matrix with ??? columns and 1344 rows, and at most 3 nonzero entries per row. The dimension of $\mathbb{M}_2(\Gamma_0(2004), \mathbb{Q})^+$ is 331.

8.6.3 Refined Algorithm For Computing Presentation

Algorithm 8.6.10 (Compute Presentation). *This is an algorithm to compute $\mathbb{M}_k(\Gamma_0(N), \mathbb{Q})$ or $\mathbb{M}_k(\Gamma_0(N), \mathbb{Q})^{\pm}$, which only requires doing generic sparse linear algebra to deal with the three term T-relations.*

1. *Let $x_0, \ldots, x_n$ by a list of all Manin symbols.*
2. *Quotient out the two-term S relations and if the $\pm$ quotient is desired, by the two-term η relations. (See Algorithm 8.6.12 below.) Let $[x_i]$ denote the class of x_i after this quotienting process.*
3. *Create a sparse matrix A with m columns, whose rows encode the relations*

$$[x_i] + [x_iT] + [x_iT^2] = 0.$$

For example, there are about $n/3$ such rows (I'm unsure what the situation is for $k > 2$). The number of nonzero entries per row is at most $3(k-1)$.

Note that we must include rows for all i*, since even if* $[x_i] = [x_j]$*, it need not be the case that* $[x_i T] = [x_j T]$*, since the matrices* S *and* T *do not commute. However, we have an a priori formula for the dimension of the quotient by all these relations, so we could omit many relations and just check that there are enough at the end—if there aren't, we add in more.*

4. *Compute the reduced row echelon form of* A *using the multi-modular (sparse) Gaussian elimination algorithm (Algorithm 7.2.3). For* $k = 2$*, this is the echelon form of a matrix with size about* $n/3 \times n/4$*.*

5. *Use what we have done above to read off a sparse matrix* R *that expresses each of the* n *Manin symbols in terms of a basis of Manin symbols, modulo the relations.*

Remark 8.6.11. There is rumored to be a "geometric" way to compute a presentation for $\mathbb{M}_2(\Gamma_0(N))$ more directly, without resorting to general linear algebra techniques. I am unaware of such a method having ever been published, but it was sketched to me independently by Georg Weber in 1999 and Robert Pollack in 2004. The computations we do after computing a presentation for $\mathbb{M}_2(\Gamma_0(N))$ are usually significantly more time consuming than computation of a presentation in the first place, so it's unclear how useful this algorithm would be in practice. (I have not heard of a method for directly obtaining a presentation for $\mathbb{M}_k(\Gamma_0(N))$.)

Algorithm 8.6.12 (Quotient By 2-Term Relations). *This algorithm performs sparse Gauss elimination on a matrix all of whose columns have at most 2 nonzero entries. This algorithm is more subtle than just "identify symbols in pairs", since complicated relations can cause generators to surprisingly equal* 0*.*

```
INPUT:
    rels -- set of pairs ((i,s), (j,t)).  The pair represents
            the relation
                  s*x_i + t*x_j = 0.
    n -- int, the x_i are x_0, ..., x_{n-1}.
    F -- base field
OUTPUT:
    mod -- list such that mod[i] = (j,s), which means that
                 x_i is equivalent to s*x_j,
           where the x_j are a basis for the quotient.
EXAMPLE:
We quotient out by the relations
       3*x0 - x1 = 0,  x1 + x3 = 0,   x2 + x3 = 0,  x4 - x5 = 0
to get
    >>> Q = rings.RationalField()
    >>> rels = set([((0,3), (1,-1)), ((1,1), (3,1)), ((2,1),(3,1)), ((4,1),(5,-1))])
    >>> n = 6
    >>> sparse_2term_quotient(rels, n, Q)
    [(3, -1/3), (3, -1), (3, -1), (3, 1), (5, 1), (5, 1)]
"""
if not isinstance(rels, set):
    raise TypeError, "rels must be a set"
if not isinstance(n, int):
    raise TypeError, "n must be an int"
if not isinstance(F, rings.Ring):
    raise TypeError, "F must be a ring."

tm = misc.verbose()
free = range(n)
ONE = F(1)
ZERO = F(0)
coef = [ONE for i in xrange(n)]
related_to_me = [[] for i in xrange(n)]
for v0, v1 in rels:
    c0 = coef[v0[0]] * F(v0[1])
    c1 = coef[v1[0]] * F(v1[1])
    # Mod out by the relation
    #    c1*x_free[t[0]] + c2*x_free[t[1]] = 0.
    die = None
    if c0 == ZERO and c1 == ZERO:
        pass
    elif c0 == ZERO and c1 != ZERO:  # free[t[1]] --> 0
        die = free[v1[0]]
    elif c1 == ZERO and c0 != ZERO:
        die = free[v0[0]]
    elif free[v0[0]] == free[v1[0]]:
        if c0+c1 != 0:
            # all xi equal to free[t[0]] must now equal to zero.
            die = free[v0[0]]
    else:  # x1 = -c1/c0 * x2.
        x = free[v0[0]]
        free[x] = free[v1[0]]
        coef[x] = -c1/c0
        for i in related_to_me[x]:
            free[i] = free[x]
            coef[i] *= coef[x]
            related_to_me[free[v1[0]]].append(i)
        related_to_me[free[v1[0]]].append(x)
    if die != None:
        for i in related_to_me[die]:
            free[i] = 0
            coef[i] = ZERO
        free[die] = 0
        coef[die] = ZERO
```

8.7 Applications

8.7.1 Later in this Book

We now sketch some of the ways in which we will apply the modular symbols algorithms of this chapter later in this book.

Cuspidal modular symbols are in Hecke-equivariant duality with cuspidal modular forms, and as such we can compute modular forms by computing systems of eigenvalues for the Hecke operators acting on modular symbols. By the Atkin-Lehner-Li theory of newforms (see, e.g., 6.1.2), we can construct $S_k(N, \varepsilon)$ for any N, any ε, and $k \geq 2$ using this method. See Chapter 1 for more details.

Once we can compute spaces of modular symbols, we move to computing the corresponding modular forms. We define inclusion and trace maps from modular symbols of one level N to modular symbols of level a multiple or divisor of N. Using these we compute the quotient V of the new subspace of cuspidal modular symbols on which a "star involution" acts as $+1$. The Hecke operators act by diagonalizable commuting matrices on this space, and computing the simultaneous systems of Hecke eigenvalues is equivalent to computing corresponding newforms $\sum a_n q^n$. In this way, we obtain a list of *all* newforms (normalized eigenforms) in $S_k(N, \varepsilon)$ for any N, ε, and $k \geq 2$.

In Chapter 10, we compute with the period mapping from modular symbols to $\mathbb{C}$ attached to a newform $f \in S_k(N, \varepsilon)$. When $k = 2, \varepsilon = 1$ and f has rational Fourier coefficients, this gives a method to compute the period lattice associated to a modular elliptic curve attached to a newform (see Section 10.6). In general, computation of this map is important when finding equations for modular $\mathbb{Q}$-curves, CM curves, and curves with a given modular Jacobian. It is also important for computing special values of the L-function $L(f, s)$ at integer points in the critical strip.

8.7.2 Discussion of the Literature and Research

Modular symbols were introduced by Birch [Bir71] in connection with computations in support of the Birch and Swinnerton-Dyer conjecture. Manin [Man72] then made a systematic study of weight 2 modular symbols and used them to prove rationality results about special values of L-functions (note that "parabolic points" in the title of Manin's paper means "cusps"). Merel's paper [Mer94] builds on work of Šokurov (mainly [Šok80]), which developed a higher-weight generalization of Manin's work partly to understand rationality properties of special values of modular L-functions (Shimura simultaneously proved similar results via related cohomological methods). Cremona's book [Cre97a] discusses in detail how to compute the space of weight 2 modular symbols for $\Gamma_0(N)$, in connection with the problem of enumerating all elliptic curves of given conductor, and his article [Cre92] discusses the $\Gamma_1(N)$ case and computation of modular symbols with character.

There have been several recent Ph.D. thesis about modular symbols. Basmaji's thesis [Bas96], which is in German, contains a tricks to efficiently compute

Hecke operators T_p, with p very large, and also discusses how to compute spaces of half integral weight modular forms building on what one can get from modular symbols of integral weight. The author's Ph.D. thesis [Ste00] contains two chapters about higher-weight modular symbols, and an application to visibility of Shafarevich-Tate groups (see also [Aga00]). Martin's thesis [Mar01] is about an attempt to study an analogue of modular symbols for weight 1. Lemelin's thesis [Lem01] discusses modular symbols for quadratic imaginary fields in the context of p-adic analogues of the Birch and Swinnerton-Dyer conjecture. See also the survey paper [FM99], which discusses computation with of weight 2 modular symbols in the context of computing with modular abelian varieties.

There are analogues for modular symbols for groups besides finite-index subgroups of $\mathrm{SL}_2(\mathbb{Z})$, e.g., for groups of higher degree, e.g., GL_3. There has also been work on computing Hilbert modular forms, e.g., by Lassina Dembele [Dem04] (Hilbert modular forms are like classical modular forms, but are functions on a product of copies of $\mathfrak{h}$, and $\mathrm{SL}_2(\mathbb{Z})$ is replaced by a group of matrices with entries in a totally real field). I am *not* aware of any analogue of modular symbols for Siegel modular forms (these are like classical modular forms, except the upper half plane is replaced by a space of matrices).

Glenn Stevens (and recently Robert Pollack and Henri Darmon, see [DP04]) has been working for many years to develop an analogue of modular symbols in a rigid analytic context, which should be very helpful for questions about computing with over convergent p-adic modular forms, or proving results about p-adic L-functions.

Gabor Weise and Bas Edixhoven have been working on theory about mod p modular symbols, and computation of weight 1 modular symbols mod 2.

Finally we mention that Mazur uses the term "modular symbol" slightly differently in many of his papers. This is a dual notion, which attaches a "modular symbol" to a modular form or elliptic curve, and is really just an overloading of the terminology. See [MTT86] for an extensive discussion of modular symbols from this point of view, where they are used to construct p-adic L-functions.

8.8 Exercises

8.1 Compute $\mathbb{M}_3(\Gamma_1(3))$ explicitly. List each Manin symbol, the relations they satisfy, compute the quotient, etc. Find the matrix of T_2. (Check: The dimension of $\mathbb{M}_3(\Gamma_1(3))$ is 2, and the characteristic polynomial of T_2 is $(x-3)(x+3)$.)

8.2 Prove that the pairing 8.5.1 is well defined.

8.3 (a) Show that if $\eta = \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, then $\eta\Gamma\eta = \Gamma$ for $\Gamma = \Gamma_0(N)$ and $\Gamma = \Gamma_1(N)$.

(b) (*) Give an example of a finite index subgroup Γ such that $\eta\Gamma\eta \neq \Gamma$.

8.4 Suppose M is an integer multiple of N. Prove that the natural map $(\mathbb{Z}/M\mathbb{Z})^* \to (\mathbb{Z}/N\mathbb{Z})^*$ is surjective.

Chapter 9

Computing Spaces of Modular Forms

9.1 Decomposing Modular Forms Using Dirichlet Characters

The group $(\mathbb{Z}/N\mathbb{Z})^*$ acts on $M_k(\Gamma_1(N))$ through the *diamond-bracket operators* $\langle d\rangle$, as follows. For $[d] \in (\mathbb{Z}/N\mathbb{Z})^*$, define

$$f|\langle d\rangle = f|[\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)]_k,$$

where $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}_2(\mathbb{Z})$ is congruent to $\left(\begin{smallmatrix} d^{-1} & 0 \\ 0 & d \end{smallmatrix}\right) \pmod N$. Note that the map $\mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$ is surjective (see Exercise 5.2), so the matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ exists. To prove that $\langle d\rangle$ preserves $M_k(\Gamma_1(N))$, we prove the more general fact that $\Gamma_1(N)$ is normal in

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) : \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \pmod N \right\}.$$

This will imply that $\langle d\rangle$ preserves $M_k(\Gamma_1(N))$ since $\left(\begin{smallmatrix} a & b \\ c & d' \end{smallmatrix}\right) \in \Gamma_0(N)$.

Lemma 9.1.1. *The group $\Gamma_1(N)$ is a normal subgroup of $\Gamma_0(N)$, and the quotient $\Gamma_0(N)/\Gamma_1(N)$ is isomorphic to $(\mathbb{Z}/N\mathbb{Z})^*$.*

Proof. Consider the surjective homomorphism $r : \mathrm{SL}_2(\mathbb{Z}) \to \mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})$. Then $\Gamma_1(N)$ is the exact inverse image of the subgroup H of matrices of the form $\left(\begin{smallmatrix} 1 & * \\ 0 & 1 \end{smallmatrix}\right)$ and $\Gamma_0(N)$ is the inverse image of the subroup T of upper triangular matrices. It thus suffices to observe that H is normal in T, which is clear. Finally, the quotient T/H is isomorphic to the group of diagonal matrices in $\mathrm{SL}_2(\mathbb{Z}/N\mathbb{Z})^*$, which is isomorphic to $(\mathbb{Z}/N\mathbb{Z})^*$. □

The diamond bracket action is simply the action of $\Gamma_0(N)/\Gamma_1(N) \cong (\mathbb{Z}/N\mathbb{Z})^*$ on $M_k(\Gamma_1(N))$. Since $M_k(\Gamma_1(N))$ is a finite dimensional vector space over $\mathbb{C}$, the $\langle d \rangle$ action breaks $M_k(\Gamma_1(N))$ up as a direct sum of factors corresponding to the Dirichlet characters $D(N, \mathbb{C})$ of modulus N.

Proposition 9.1.2. *We have*

$$M_k(\Gamma_1(N)) = \bigoplus_{\varepsilon \in D(N,\mathbb{C})} M_k(N, \varepsilon),$$

where

$$M_k(N, \varepsilon) = \left\{ f \in \mathcal{M}_k(\Gamma_1(N)) : f|\langle d \rangle = \varepsilon(d) f \text{ all } d \in (\mathbb{Z}/N\mathbb{Z})^* \right\}.$$

Proof. The linear transformations $\langle d \rangle$, for the $d \in (\mathbb{Z}/N\mathbb{Z})^*$, all commute, since $\langle d \rangle$ acts through the abelian group $\Gamma_0(N)/\Gamma_1(N)$. Also, if e is the exponent of $(\mathbb{Z}/N\mathbb{Z})^*$, then $\langle d \rangle^e = \langle d^e \rangle = \langle 1 \rangle = 1$, so the matrix of $\langle d \rangle$ is diagonalizable. It is a standard fact from linear algebra that any commuting family of diagonalizable linear transformations is simultaneously diagonalizable (see Exercise 5.4), so there is a basis $f_1, \ldots, f_n$ for $M_k(\Gamma_1(N))$ so that all $\langle d \rangle$ act by diagonal matrices. The eigenvalues of the action of $(\mathbb{Z}/N\mathbb{Z})^*$ on a fixed f_i defines a Dirichlet character, i.e., each f_i has the property that $f_i|\langle d \rangle = \varepsilon_i(d)$, for all $d \in (\mathbb{Z}/N\mathbb{Z})^*$ and some Dirichlet character ε_i. The f_i for a given ε then span $M_k(N, \varepsilon)$, and taken together the $M_k(N, \varepsilon)$ must span $M_k(\Gamma_1(N))$. □

Definition 9.1.3 (Character of Modular Form). If $f \in M_k(N, \varepsilon)$, we say that f *has character* ε.

Remark 9.1.4. People sometimes write that f has "nebentypus character" ε.

The spaces $M_k(N, \varepsilon)$ are a direct sum of subspaces $S_k(N, \varepsilon)$ and $E_k(N, \varepsilon)$, where $S_k(N, \varepsilon)$ is the subspace of cusp forms, i.e., forms that vanish at *all* cusps (elements of $\mathbb{Q} \cup \{\infty\}$), and $E_k(N, \varepsilon)$ is the subspace of Eisenstein series, which is the unique subspace of $M_k(N, \varepsilon)$ that is invariant under all Hecke operators and is such that $M_k(N, \varepsilon) = S_k(N, \varepsilon) \oplus E_k(N, \varepsilon)$. The space $E_k(N, \varepsilon)$ can also be defined as the space spanned by all Eisenstein series of weight k and level N, as defined in Chapter 5. The space $E_k(N, \varepsilon)$ can also be defined using the Petersson inner product (see, e.g., [Lan95]).

The diamond bracket operators preserve the subspace of cusp forms, so the isomorphism of Proposition 9.1.2 restricts to an isomorphism of the corresponding cuspidal subspaces. SAGE implements dimension formulas for general spaces of cusp forms, which we can use to make a table giving the dimension of $S_k(\Gamma_1(N))$ and of the dimension of each subspace corresponding to a character. We do this first for $N = 13$.

```
sage: G = DirichletGroup(13)
sage: G
Group of Dirichlet characters of modulus 13 over Cyclotomic Field
      of order 12 and degree 4
sage: dimension_cusp_forms(Gamma1(13),2)
2
sage: [dimension_cusp_forms_eps(e,2) for e in G]
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]
```

Next we do this for $N = 100$.

```
sage: G = DirichletGroup(100)
sage: G
Group of Dirichlet characters of modulus 100 over Cyclotomic Field
      of order 20 and degree 8
sage: dimension_cusp_forms_gamma1(100)
231
sage: [dimension_cusp_forms_eps(e,2) for e in G]
[ 7, 0, 0, 13, 12, 0, 0, 13, 12, 0, 0, 9,  12, 0, 0, 13,
 12, 0, 0, 13,  6, 0, 0, 13, 12, 0, 0, 13, 12, 0, 0, 9,
 12, 0, 0, 13, 12, 0, 0, 13]
```

9.2 Atkin-Lehner-Li Theory

By Atkin-Lehner-Li theory (see [AL70, Li75]), we have a decomposition

$$S_k(\Gamma_1(N)) = \bigoplus_{M\mid N} \bigoplus_{d\mid N/M} \alpha_d(S_k(\Gamma_1(M))_{\mathrm{new}}). \tag{9.2.1}$$

Here $\alpha_d : S_k(\Gamma_1(M)) \to S_k(\Gamma_1(N))$ is the degeneracy map $f(q) \mapsto f(q^d)$, and $S_k(\Gamma_1(M))_{\mathrm{new}}$ is the largest $\mathbb{T}$-stable (or Petersson) complement of the image of all maps α_d from level properly dividing M. The analogue of (9.2.1) with Γ_1 replaced by Γ_0 is true; it is also true with character, as long as we omit the spaces $S_k(\Gamma_1(M), \varepsilon)$ for which $M \nmid \mathrm{cond}(\varepsilon)$.

Example 9.2.1. If N is prime and $k \leq 11$, then $S_k(\Gamma_1(N))_{\mathrm{new}} = S_k(\Gamma_1(N))$, since $S_k(\Gamma_1(1)) = 0$.

One can prove using the Petersson inner product that the Hecke operators T_n on $S_k(\Gamma_1(N))$, with $(n, N) = 1$, are diagonalizable. Another result of Atkin-Lehner-Li theory is that the ring of endomorphism of $S_k(\Gamma_1(N))_{\mathrm{new}}$ generated by all Hecke operators equals the ring generated by the Hecke operators T_n with $(n, N) = 1$. This statement need *not* be true if we do not restrict to the new subspace.

Example 9.2.2. We have

$$S_2(\Gamma_0(22)) = S_2(\Gamma_0(11)) \oplus \alpha_2(S_2(\Gamma_0(11))),$$

where each of the spaces $S_2(\Gamma_0(11))$ has dimension 1. Thus $S_2(\Gamma_0(22))_{\rm new} = 0$. The Hecke operator T_2 on $S_2(\Gamma_0(22))$ has characteristic polynomial x^2+2x+2, which is irreducible. Since α_2 commutes with all Hecke operators T_n, with $\gcd(n,2) = 1$, the subring $\mathbb{T}'$ of the Hecke algebra generated by operators T_n with n odd is isomorphic to $\mathbb{Z}$ (the 2×2 scalar matrices). Thus on the full space $S_2(\Gamma_0(22))$, we do not have $\mathbb{T}' = \mathbb{T}$. However, on the new subspace we do have this equality, since the new subspace has dimension 0.

Example 9.2.3. This example is similar to Example 9.2.2, except that there are newforms. We have

$$S_2(\Gamma_0(55)) = S_2(\Gamma_0(11)) \oplus \alpha_5(S_2(\Gamma_0(11))) \oplus S_2(\Gamma_0(55))_{\rm new},$$

where $S_2(\Gamma_0(11))$ has dimension 1 and $S_2(\Gamma_0(55))_{\rm new}$ has dimension 3. The Hecke operator T_5 on $S_2(\Gamma_0(55))_{\rm new}$ acts via the matrix

$$\begin{pmatrix} -2 & 2 & -1 \\ -1 & 1 & -1 \\ 1 & -2 & 0 \end{pmatrix}$$

with respect to some basis. This matrix has eigenvalues 1 and -1. Atkin-Lehner theory asserts that T_5 must be a linear combination of Hecke operators T_n, with $\gcd(n, 55) = 1$. Upon computing the matrix for T_2, we find by simple linear algebra that $T_5 = 2T_2 - T_4$.

Before moving on, we pause to say something about how the Atkin-Lehner-Li theorems are proved. A key result is to prove that if $f, g \in S_k(\Gamma_1(N))_{\rm new}$ and $a_n(f) = a_n(g)$ for all n with $\gcd(n, N) = 1$, then $f = g$. First, replace f and g by their difference $h = f - g$, and observe that $a_n(h) = 0$ for $\gcd(n, N) = 1$. Note that such an h "looks like" it is in the image of the maps α_d, for $d \mid N$. In fact it is—one shows that h is in the old subspace $S_k(\Gamma_1(N))_{\rm old}$ (this is the "crucial" Theorem 2 of [Li75]). But h is also new, since it is the difference of two newforms, so $h = 0$, hence $f = g$. The details involve introducing many maps between spaces of modular forms, and computing what they do to q-expansions.

Definition 9.2.4 (Newform). A *newform* is a $\mathbb{T}$-eigenform $f \in S_k(\Gamma_1(N))_{\rm new}$ that is normalized so that the coefficient of q is 1.

We now motivate this definition by explaining why any eigenform can be normalized so that the coefficient of q is 1, and how such an eigenform has the convenient properties that its Fourier coefficients are exactly the Hecke eigenvalues.

Proposition 9.2.5. *The coefficients of a normalized $\mathbb{T}$-eigenform are the eigenvalues.*

Proof. The Hecke algebra $\mathbb{T}_{\mathbb{Q}}$ on $S_k(\Gamma_1(N))$ contains the diamond bracket operators $\langle d\rangle$, since $T_{p^2} = T_p^2 - \langle p\rangle p^{k-1}$, so any $\mathbb{T}$-eigenform lies in a subspace $S_k(\Gamma_1(N), \varepsilon)$ for some Dirichlet character ε. The Hecke operators T_p, for p prime, act on $S_k(\Gamma_1(N), \varepsilon)$ by

$$T_p\left(\sum_{n=1}^{\infty} a_n q^n\right) = \sum_{n=1}^{\infty}\left(a_{np}q^n + \varepsilon(p)p^{k-1}a_n q^{np}\right),$$

and there is a similar formula for T_m with m composite. If $f = \sum_{n=1}^{\infty} a_n q^n$ is an eigenform for all T_p, with eigenvalues λ_p, then by the above formula

$$\lambda_p f = \lambda_p a_1 q + \lambda_p a_2 q^2 + \cdots = T_p(f) = a_p q + \text{higher terms}. \qquad (9.2.2)$$

Equating coefficients of q we see that if $a_1 = 0$, then $a_p = 0$ for all p, hence $a_n = 0$ for all n, because of the multiplicativity of Fourier coefficients and the recurrence

$$a_{p^r} = a_{p^{r-1}}a_p - \varepsilon(p)p^{k-1}a_{p^{r-2}}.$$

This would mean that $f = 0$, a contradiction. Thus $a_1 \neq 0$, and it makes sense to normalize f so that $a_1 = 1$. With this normalization, (9.2.2) implies that $\lambda_p = a_p$, as desired. □

Remark 9.2.6. We even have that the operators $\langle d\rangle$ on $S_k(\Gamma_1(N))$ lie in $\mathbb{Z}[\ldots, T_n, \ldots]$. It is enough to show $\langle p\rangle \in \mathbb{Z}[\ldots, T_n, \ldots]$ for primes p, since each $\langle d\rangle$ can be written in terms of the $\langle p\rangle$. Since $p \nmid N$, we have that

$$T_{p^2} = T_p^2 - \langle p\rangle p^{k-1},$$

so $\langle p\rangle p^{k-1} = T_p^2 - T_{p^2}$. By Dirichlet's theorem on primes in arithmetic progression, there is another prime q congruent to p mod N. Since p^{k-1} and q^{k-1} are relatively prime, there exist integers a and b such that $ap^{k-1} + bq^{k-1} = 1$. Then

$$\langle p\rangle = \langle p\rangle(ap^{k-1} + bq^{k-1}) = a({T_p}^2 - T_{p^2}) + b({T_q}^2 - T_{q^2}) \in \mathbb{Z}[\ldots, T_n, \ldots].$$

9.3 Computing Cuspforms Using Modular Symbols

There is an isomorphism

$$S_k(\Gamma_1(N), \varepsilon)_{\mathrm{new}} \cong \mathbb{S}_k(\Gamma_1(N), \varepsilon, \mathbb{C})^+_{\mathrm{new}}$$

of $\mathbb{T}$ modules. Thus finding the systems of $\mathbb{T}$-eigenvalues on cuspforms is the same as finding the systems of $\mathbb{T}$-eigenvalues on cuspidal modular symbols.

Our strategy to compute $S_k(\Gamma_1(N), \varepsilon)$ is to first reduce to computing spaces $S_k(\Gamma_1(N), \varepsilon)_{\mathrm{new}}$ using the Atkin-Lehner-Li decomposition (9.2.1). To compute $S_k(\Gamma_1(N), \varepsilon)_{\mathrm{new}}$ to a given precision, we compute the systems of eigenvalues of the Hecke operators T_p on $V = \mathbb{S}_k(\Gamma_1(N), \varepsilon, \mathbb{C})^+_{\mathrm{new}}$. Using Proposition 9.2.5, we then recover a basis of q-expansions for newforms. Note that we only need to compute Hecke eigenvalues T_p, for p prime, not the T_n for n composite, since the a_n can be quickly recovered in terms of the a_p using multiplicativity and the recurrence.

For many problems, one is really interested in the newforms, not just any basis for $S_k(\Gamma_1(N),\varepsilon)$. The are many other problems where just having a basis is enough, and knowing the newforms is not so important. Merel's paper [Mer94] culminates with the following algorithm to compute $S_k(\Gamma_1(N),\varepsilon)$ without finding any eigenspaces:

Algorithm 9.3.1 (Merel's Algorithm for Computing a Basis). *1. [Compute Modular Symbols] Using Algorithm 8.6.10, compute a presentation for $V = \mathbb{S}_k(\Gamma_1(N),\varepsilon)^+ \otimes \mathbb{Q}(\varepsilon)$, viewed as a $K = \mathbb{Q}(\varepsilon)$ vector space, along with an action of Hecke operators T_n.*

2. *[Basis for Linear Dual] Write down a basis for $V^* = \mathrm{Hom}(V, \mathbb{Q}(\varepsilon))$. E.g., if we identify V with K^n viewed as column vectors, then V^* is the space of row vectors of length n, and the pairing is the row $\times$ column product.*

3. *[Find Generator] Find $x \in V$ such that $\mathbb{T}x = V$ by choosing random x until we find one that generates. The set of x that fail to generate lie in a union of a finite number of proper subspace. (This can be seen by analyzing the structure of $S_k(\Gamma_1(N),\varepsilon)$ as a $\mathbb{T}$-module; see, e.g., my 252 notes.)*

4. *[Compute Basis] The set of power series*

$$f_i = \sum_{n=1}^{m} \psi_i(T_n(x))q^n + O(q^{m+1})$$

form a basis for $S_k(\Gamma_1(N),\varepsilon)$ to precision m.

In practice my experience is that my implementations of Algorithm 9.3.1 are significantly slower than the eigenspace algorithm that we will describe in the rest of this chapter. The theoretical complexity of Algorithm 9.3.1 *may* be better, because it is not necessary to factor any polynomials. Polynomial factorization is difficult from the analysis-of-complexity point of view, though usually fairly fast in practice. The eigenvalue algorithm only requires computing a few images $T_p(x)$ for p prime and x a Manin symbol on which T_p can easily be computed. The Merel algorithm involves computing $T_n(x)$ for all n, and a fairly easy x, which is potentially more work. (By "easy x", I mean that computing $T_n(x)$ is easier on x than on a completely random element of $\mathbb{S}_k(\Gamma_1(N),\varepsilon)^+$, e.g., x could be a Manin symbol.)

9.4 Computing Systems of Eigenvalues

In this section we describe an algorithm for computing the system of Hecke eigenvalues associated to a simple subspace of a space of modular symbols. This algorithm is vastly better than naively doing linear algebra over the number field generated by the eigenvalues. It only involves linear algebra over the base field, and also yields a very compact representation for the answer, which is much better than writing the eigenvalues in terms of a power basis for a number field.

9.4.1 Computing Projection Onto a Subspace

Suppose $V = \oplus W_i$ is the $\mathbb{T}$-simple decomposition of V and fix a factor W_j. Then there is a unique $\mathbb{T}$-equivariant homomorphism

$$\pi_j : V \to W_j$$

such that π_j restricted to W_j is the identity map. We compute π_j using the following algorithm.

Algorithm 9.4.1 (Projection Matrix). *INPUT: Decomposition $V = \oplus W_i$. OUTPUT: Matrix of Projection Onto a Factor W_j.*

1. *Let B be the matrix whose columns are got by concatenating together a basis for the factors W_i.*
2. *Compute $C = B^{-1}$ using, e.g., computation of the reduced row echelon form of the augmented matrix $[B|I]$, which is $[I|C]$.*
3. *The projection matrix onto W_j is the submatrix of C got from the rows corresponding to W_j, i.e., if the basis vectors for W_j appear as columns n through m of B, then the projection matrix is got from rows n through m of C.*

The algorithm works because the matrix of projection, written with respect to the basis of columns for B, is just given by an $m - n + 1$ row slice P of a diagonal matrix D with 1's in the n through m positions. Thus projection with respect to the standard basis is given by PC, which is just rows n through m of B^{-1}.

Note that we only have to do the work of inverting B once; we then get all projection maps π_i for all i by taking appropriate submatrices of B.

9.4.2 Systems of Eigenvalues

Algorithm 9.4.2 (System of Eigenvalues). *INPUT: A $\mathbb{T}$-simple subspace $W \subset V$ of modular symbols.*
OUTPUT: Maps ψ and e, where $\psi : \mathbb{T}_K \to W$ is a K-linear map and $e : W \cong L$ is an isomorphism of W with a number field L, such that $a_n = e(\psi(T_n))$ is the eigenvalue of the nth Hecke operator acting on a fixed $\mathbb{T}$-eigenvector in $W \otimes \overline{\mathbb{Q}}$. Thus $f = \sum_{n=1}^{\infty} i(\psi(T_n))q^n$ is a cuspidal modular eigenform.

1. *[Compute Projection] Using Algorithm 9.4.1, compute the $\mathbb{T}$-equivariant projection map $\pi : V \to W$. Remark: We can replace π by any K-vector space map $\varphi : V \to W'$ such that $\mathrm{Ker}(\pi) = \mathrm{Ker}(\varphi)$, where W' is any*

vector space, and the rest of the algorithm works. For example, one could find such a φ by finding the simple submodule of $V^* = \mathrm{Hom}(V, K)$ that is isomorphic to W, e.g., by applying Algorithm 7.4.8 to V^* with T replaced by the transpose of T. This is what Cremona means in his book when he talks about find "left eigenvectors".

2. *[Choose v] Choose a nonzero element $v \in V$ such that $\pi(v) \neq 0$ and computation of $T_n(v)$ is "easy", e.g., choose v to be a Manin symbol.*

3. *[Map From Hecke Ring] Let ψ be the map $\mathbb{T} \to W$, given by $\psi(t) = \pi(tv)$. Note that computation of ψ is relatively easy, because v was chosen so that tv is relatively easy to compute. In particular, if $t = T_p$, we do not need to compute the full matrix of T_p on V; instead we just compute $T_p(v)$. (We can even often compute eigenvalues for* all *the factors W_i just by computing one evaluation $T_p(v)$ for a single easy v!)*

4. *[Find Generator] Find a random $T \in \mathbb{T}$ such that the iterates*

$$\psi(T^0), \quad \psi(T), \quad \psi(T^2), \quad \ldots, \quad \psi(T^{d-1})$$

are a basis for W, where W has dimension d. For example, the T that was used to compute the decomposition $V = \oplus W_i$ earlier would work.

5. *[Characteristic Polynomial] Compute the characteristic polynomial f of $T|_W$, and let $L = K[x]/(f)$ be the number field generated by a root of f. Because of how we chose T in Step 4, the minimal and characteristic polynomials of $T|_W$ are equal, and both are irreducible, so L is an extension of K of degree $d = \dim(W)$. If in Step 4, we used the T used to compute the decomposition $V = \oplus W_i$ earlier, then we already know f.*

6. *[Field Structure] In this step we endow W with a field structure. Let $e : W \to L$ be the unique K-linear isomorphism such that*

$$e(\psi(T^i)) \equiv x^i \pmod{f}$$

for $i = 0, 1, 2, ..., \deg(f) - 1$. The map e is uniquely determined since the $\psi(T^i)$ are a basis for W. To compute e, we compute the change of basis matrix from the standard basis for W to the basis $\{\psi(T^i)\}$. This change of basis matrix is the inverse of the matrix whose rows are the $\psi(T^i)$ for $i = 0, ..., \deg(f) - 1$.

7. *[Hecke Eigenvalues] Finally note that we have*

$$a_n = e(\psi(T_n)) = e(\pi(T_n(v)))$$

for Hecke operators T_n, where the a_n are eigenvalues. Output the maps ψ and e and terminate.

One reason we separate ψ and e is that when $\dim(W)$ is large, the values $\psi(T_n)$ tend to take not too much space to store and are easier to compute, whereas each one of the values $e(\psi(n))$ are **huge**. John Cremona initially suggested to me the idea of separating these two maps. The function e typically involves large numbers if $\dim(W)$ is large, since e is got from the iterates of a single vector. For many applications, e.g., databases, it is better to store a matrix that defines e and the images under ψ of many T_n.

Remark 9.4.3. How can we find a minimal collection of information from which we can compute the map $n \mapsto \psi(T_n)$? Do we need the whole modular symbols presentation? No, we need only the image of each generating Manin symbol in M under projection. The Hecke operators are then given by the standard Manin symbols formulas, where we reduce all resulting Manin symbols to their image in M.

Example 9.4.4. The space $S_2(\Gamma_0(23))$ of cusp forms has dimension 2, and is spanned by two $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$-conjugate newforms, one of which is

$$f = \sum q + aq^2 + (-2a-1)q^3 + (-a-1)q^4 + 2aq^5 + \cdots,$$

where $a = (-1+\sqrt{5})/2$. We will use Algorithm 9.4.2 to compute a few of these coefficients.

The space $\mathbb{M}_2(\Gamma_0(23))^+$ of modular symbols has dimension 3. It has as basis the following basis of Manin symbols:

$$[(0,0)], \quad [(1,0)], \quad [(0,1)],$$

where we use square brackets to differentiate Manin symbols from vectors. The Hecke operator

$$T_2 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 0 & 2 \\ -1 & 1/2 & -1 \end{pmatrix}$$

has characteristic polynomial $(x-3)(x^2+x-1)$. The kernel of T_2-3 corresponds to the span of the Eisenstein series of level 23 and weight 2, and the kernel V of $T_2^2+T_2-1$ corresponds to $S_2(\Gamma_0(23))$. (We could also have computed V as the kernel of the boundary map $\mathbb{M}_2(\Gamma_0(23))^+ \to \mathbb{B}_2(\Gamma_0(23))^+$.) Each of the following steps corresponds to the same step of Algorithm 9.4.2.

1. [Compute Projection] Using the Algorithm 9.4.1, we compute projection onto V. The matrix whose first two columns are the echelon basis for V and whose last column is the echelon basis for the Eisenstein subspace is

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -2/11 \\ 0 & 1 & -3/11 \end{pmatrix}$$

and

$$B^{-1} = \begin{pmatrix} 2/11 & 1 & 0 \\ 3/11 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix},$$

so projection onto V is given by the first two rows:

$$\pi = \begin{pmatrix} 2/11 & 1 & 0 \\ 3/11 & 0 & 1 \end{pmatrix}.$$

2. [Choose v] Let $v = (0,1,0)^t$. Notice that $\pi(v) = (1,0)^t \neq 0$, and $v = [(1,0)]$ is a sum of only one Manin symbol, so it is easier to compute Hecke operators on v using Heilbronn matrices.

3. [Map From Hecke Ring] This step is purely conceptual, since no actual work needs to be done. We illustrate it by computing $\psi(T_1)$ and $\psi(T_2)$. We have
$$\psi(T_1) = \pi(v) = (1,0)^t,$$
and
$$\psi(T_2) = \pi(T_2(v)) = \pi((0,0,1/2)^t) = (0,1/2)^t.$$

4. [Find Generator] We have
$$\psi(T_2^0) = \psi(T_1) = (1,0)^t,$$
which is clearly independent from $\psi(T_2) = (0,1/2)^t$. Thus we find that the image of the powers of $T = T_2$ generate V.

5. [Characteristic Polynomial] It is easy to compute the characteristic polynomial of a 2×2 matrix. The matrix of $T_2|_V$ is $\left(\begin{smallmatrix} 0 & 2 \\ 1/2 & -1 \end{smallmatrix}\right)$, which has characteristic polynomial $f = x^2 + x - 1$. Of course, we already knew this because we computed V as the kernel of $T_2^2 + T_2 - 1$.

6. [Field Structure] We have
$$\psi(T_2^0) = \pi(v) = (1,0)^t \text{ and } \psi(T_2) = (0,1/2).$$
The matrix with rows the $\psi(T_2^i)$ is $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1/2 \end{smallmatrix}\right)$, which has inverse $e = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)$. The matrix e defines an isomorphism between V and the field
$$L = \mathbb{Q}[x]/(f) = \mathbb{Q}((-1+\sqrt{5})/2).$$
For example, $e((1,0)) = 1$ and $e((0,1)) = 2x$, where $x = (-1+\sqrt{5})/2$.

7. [Hecke Eigenvalues] We have $a_n = e(\Psi(T_n))$. For example,
$$\begin{aligned}
a_1 &= e(\Psi(T_1)) = e((1,0)) = 1 \\
a_2 &= e(\Psi(T_2)) = e((0,1/2)) = x \\
a_3 &= e(\Psi(T_3)) = e(\pi(T_3(v))) = e(\pi((0,-1,-1)^t)) = e((-1,-1)^t) = -1-2x \\
a_4 &= e(\Psi(T_4)) = e(\pi((0,-1,-1/2)^t)) = e((-1,-1/2)^t) = -1-x \\
a_5 &= e(\Psi(T_5)) = e(\pi((0,0,1)^t)) = e((0,1)^t) = 2x \\
a_{23} &= e(\Psi(T_{23})) = e(\pi((0,1,0)^t)) = e((1,0)^t) = 1 \\
a_{97} &= e(\Psi(T_{23})) = e(\pi((0,14,3)^t)) = e((14,3)^t) = 14+6x
\end{aligned}$$

It is difficult to appreciate this algorithm without seeing how big the coefficients of the power series expansion of a newform typically are, when the newform is defined over a large field. For such examples, please browse [Ste04].

Chapter 10

Special Values of L-functions and Periods

This chapter is about how to approximate the integration pairing, and the induced period mapping from modular symbols to a complex vector space.

Let Γ be a subgroup of $\mathrm{SL}_2(\mathbb{Z})$ that contains $\Gamma_1(N)$ for some N, and suppose

$$f = \sum_{n\geq 1} a_n q^n \in S_k(\Gamma).$$

is a newform. In this chapter we describe how to approximately compute the complex period mapping

$$\Phi_f : \mathbb{M}_k(\Gamma) \to \mathbb{C},$$

given by

$$\Phi_f(P\{\alpha,\beta\}) = \langle f,\, P\{\alpha,\beta\}\rangle = \int_\alpha^\beta f(z)P(z,1)dz,$$

as in Section 8.5. As an application, we approximate the special values $L(f,j)$, for $j = 1, 2, \ldots, k-1$ using (8.5.3) from page 116. We also compute the period lattice attached to a modular abelian variety, which is an important step, e.g., in enumeration of $\mathbb{Q}$-curves [cite Gonzalez, Lario, etc.] or computation of a curve whose Jacobian is a modular abelian variety A_f [cite X. Wang and Ph.D. thesis from Essen].

The algorithms that we describe in this chapter are a generalization of the ones in [Cre97a] to other congruence subgroups, newforms of degree bigger than 1, and weight bigger than 2.

10.1 The Period Mapping and Complex Torus Attached to a Newform

Fix a newform $f \in S_k(\Gamma)$, where $\Gamma_1(N) \subset \Gamma$ for some N. Let $f_1, \ldots, f_d$ be the $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$-conjugates of f, where $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts via its action on the Fourier coefficients, which are algebraic integers. Let

$$V_f = \mathbb{C}f_1 \oplus \cdots \oplus \mathbb{C}f_d \subset S_k(\Gamma)$$

be the subspace of cusp forms spanned by the $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$-conjugates of f. The integration pairing induces a $\mathbb{T}$-equivariant homomorphism

$$\Phi_f : \mathbb{M}_k(\Gamma) \to V_f^* = \mathrm{Hom}_{\mathbb{C}}(V_f, \mathbb{C}),$$

from modular symbols to the $\mathbb{C}$-linear dual V_f^* of V_f. Here $\mathbb{T}$ acts on V_f^* via $(\varphi.t)(x) = \varphi(tx)$, and this homomorphism is $\mathbb{T}$-stable by Theorem 8.5.4. The *complex torus attached to* f is the quotient

$$A_f(\mathbb{C}) = V_f^*/\Phi_f(\mathbb{S}_k(\Gamma, \mathbb{Z})).$$

Note that $\mathbb{S}_k(\Gamma, \mathbb{Z}) = \mathbb{S}_k(\Gamma)$, and we include the $\mathbb{Z}$ in the notation here just to emphasize that these are integral modular symbols.

When $k = 2$, we can also construct A_f as a quotient of the modular Jacobian $\mathrm{Jac}(X_\Gamma)$, so A_f is an abelian variety canonically defined over $\mathbb{Q}$.

In general, we have an exact sequence

$$0 \to \mathrm{Ker}(\Phi_f) \to \mathbb{S}_k(\Gamma) \to V_f^* \to A_f(\mathbb{C}) \to 0.$$

When $k = 2$, we have an exact sequence

$$0 \to B \to \mathrm{Jac}(X_\Gamma) \to A_f \to 0,$$

where $X_\Gamma = \Gamma\backslash\mathfrak{h}^*$ is the modular curve associated to Γ and B is some abelian variety. We have

$$\mathrm{H}_1(\mathrm{Jac}(X_\Gamma), \mathbb{Z}) \cong \mathrm{H}_1(X_\Gamma, \mathbb{Z}) \cong \mathbb{S}_2(\Gamma),$$

so the induced map on homology is

$$0 \to \mathrm{H}_1(B, \mathbb{Z}) \to \mathbb{S}_2(\Gamma) \to \mathrm{H}_1(A_f, \mathbb{Z}).$$

Thus we can identify the homology of A_f with a quotient of modular symbols.

Remark 10.1.1 (Warnings). In the literature, the notation A_f is sometimes used for the abelian subvariety of $C \subset \mathrm{Jac}(X_\Gamma)$ attached to f. Here C is the abelian variety dual of our A_f. Also, f could be a newform for a different group Γ', and then the corresponding abelian variety A_f could be different, so A_f depends on the choice of Γ. For example, any newform for $\Gamma_0(N)$ is also a newform for $\Gamma_1(N)$, but the corresponding A_f's need not be equal.

Remark 10.1.2. When $k > 2$, it is my understanding that the complex torus $A_f(\mathbb{C})$ is an abelian variety over $\mathbb{C}$. This additional abelian variety structure comes somehow from the Petersson inner product. I believe Shimura proves this in [Shi59].

10.2 Extended Modular Symbols

In this section, we extend the notion of modular symbols to allows symbols of the form $P\{w, z\}$ where w and z are arbitrary elements of $\mathfrak{h}^* = \mathfrak{h} \cup \mathbb{P}^1(\mathbb{Q})$.

Definition 10.2.1 (Extended Modular Symbols). The abelian group $\overline{\mathbb{M}}_k$ of *extended modular symbols of weight* k is the $\mathbb{Z}$-span of symbols $P\{w, z\}$, with $P \in V_{k-2}$ a homogenous polynomial of degree $k-2$ with integer coefficients, modulo the relations

$$P \cdot (\{w, y\} + \{y, z\} + \{z, w\}) = 0$$

and modulo any torsion.

Fix a finite-index subgroup $\Gamma \subset \mathrm{SL}_2(\mathbb{Z})$. Just as for usual modular symbols, $\overline{\mathbb{M}}_k$ is equipped with an action of Γ, and we define the space of extended modular of weight k for Γ to be the biggest quotient

$$\overline{\mathbb{M}}_k(\Gamma) = (\overline{\mathbb{M}}_k / \{\gamma x - x : \gamma \in \Gamma, x \in \overline{\mathbb{M}}_k\}) / \operatorname{tor}$$

of $\overline{\mathbb{M}}_k(\Gamma)$ that is torsion free and fixed by Γ.

The integration pairing extends naturally to a pairing

$$\bigl(S_k(\Gamma) \oplus \overline{S}_k(\Gamma)\bigr) \times \overline{\mathbb{M}}_k(\Gamma) \to \mathbb{C}, \tag{10.2.1}$$

where we recall that $\overline{S}_k(\Gamma)$ denotes the space of antiholomorphic cusp forms. Moreover, if

$$\iota : \mathbb{M}_k(\Gamma) \hookrightarrow \overline{\mathbb{M}}_k(\Gamma)$$

is the natural embedding, then ι respects (10.2.1) in the sense that for all $f \in S_k(\Gamma) \oplus \overline{S}_k(\Gamma)$ and $x \in \mathbb{M}_k(\Gamma)$, we have

$$\langle f, x\rangle = \langle f, \iota(x)\rangle.$$

As we will see soon, it is often useful to replace $x \in \mathbb{M}_k(\Gamma)$ first by $\iota(x)$, and then by an equivalent sum $\sum y_i$ of symbols $y_i \in \overline{\mathbb{M}}_k(N, \varepsilon)$ such that $\langle f, \sum y_i\rangle$ is easier to compute numerically than $\langle f, x\rangle$.

For any Dirichlet character ε modulo N we also define $\overline{\mathbb{M}}_k(\Gamma_1(N), \varepsilon)$ to be the quotient of $\overline{\mathbb{M}}_k(\Gamma_1(N), \mathbb{Z}[\varepsilon])$ by the relations $\gamma(x) - \varepsilon(\gamma)x$, for all $\gamma \in \Gamma_0(N)$, and modulo any torsion. (Recall that if $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, then $\varepsilon(\gamma) = \varepsilon(d)$.)

10.3 Numerically Approximating Period Integrals

In this section we assume Γ is a congruence subgroup of $\mathrm{SL}_2(\mathbb{Z})$ that contains $\Gamma_1(N)$ for some N. Suppose $\alpha \in \mathfrak{h}$, so $\mathrm{Im}(\alpha) > 0$ and m is an integer such that $0 \le m \le k-2$, and consider the extended modular symbol $X^m Y^{k-2-m}\{\alpha, \infty\}$.

Given an arbitrary cusp form $f = \sum_{n=1}^{\infty} a_n q^n \in S_k(\Gamma)$, we find that

$$\begin{aligned} \Phi_f(X^m Y^{k-2-m}\{\alpha,\infty\}) &= \langle f,\, X^m Y^{k-2-m}\{\alpha,\infty\}\rangle && (10.3.1)\\ &= \int_{\alpha}^{i\infty} f(z) z^m dz && (10.3.2)\\ &= \sum_{n=1}^{\infty} a_n \int_{\alpha}^{i\infty} e^{2\pi i n z} z^m dz. && (10.3.3)\end{aligned}$$

The reversal of summation and integration is justified because the imaginary part of α is positive so that the sum converges absolutely. This is made explicit in the following lemma, which one proves by repeated integration by parts.

Lemma 10.3.1.

$$\int_{\alpha}^{i\infty} e^{2\pi i n z} z^m dz \;=\; e^{2\pi i n \alpha} \sum_{s=0}^{m} \left(\frac{(-1)^s \alpha^{m-s}}{(2\pi i n)^{s+1}} \prod_{j=(m+1)-s}^{m} j \right). \qquad (10.3.4)$$

In practice we will be interested in computing the period map Φ_f when $f \in S_k(\Gamma)$ is a newform. Since f is a newform, there is a Dirichlet character ε such that $f \in S_k(\Gamma_1(N), \varepsilon)$. The period map $\Phi_f : \mathbb{M}_k(\Gamma) \to \mathbb{C}$ then factors through the quotient $\mathbb{M}_k(\Gamma_1(N), \varepsilon)$, so it suffices to compute the period map on modular symbols in $\mathbb{M}_k(\Gamma_1(N), \varepsilon)$.

The following proposition is a higher weight analogue of [Cre97a, Prop. 2.1.1(5)].

Proposition 10.3.2. *For any $\gamma \in \Gamma_0(N)$, $P \in V_{k-2}$ and $\alpha \in \mathfrak{h}^*$, we have the following relation in $\overline{\mathbb{M}}_k(\Gamma_1(N), \varepsilon)$:*

$$\begin{aligned} P\{\infty, \gamma(\infty)\} &= P\{\alpha, \gamma(\alpha)\} + (P - \varepsilon(\gamma)\gamma^{-1}P)\{\infty, \alpha\} && (10.3.5)\\ &= \varepsilon(\gamma)(\gamma^{-1}P)\{\alpha, \infty\} - P\{\gamma(\alpha), \infty\}. && (10.3.6)\end{aligned}$$

Proof. By definition, if $x \in \mathbb{M}_k(N, \varepsilon)$ is a modular symbol and $\gamma \in \Gamma_0(N)$ then $\gamma x = \varepsilon(\gamma) x$. Thus $\varepsilon(\gamma)\gamma^{-1} x = x$, so

$$\begin{aligned} P\{\infty, \gamma(\infty)\} &= P\{\infty, \alpha\} + P\{\alpha, \gamma(\alpha)\} + P\{\gamma(\alpha), \gamma(\infty)\}\\ &= P\{\infty, \alpha\} + P\{\alpha, \gamma(\alpha)\} + \varepsilon(\gamma)\gamma^{-1}(P\{\gamma(\alpha), \gamma(\infty)\})\\ &= P\{\infty, \alpha\} + P\{\alpha, \gamma(\alpha)\} + \varepsilon(\gamma)(\gamma^{-1}P)\{\alpha, \infty\}\\ &= P\{\alpha, \gamma(\alpha)\} + P\{\infty, \alpha\} - \varepsilon(\gamma)(\gamma^{-1}P)\{\infty, \alpha\}\\ &= P\{\alpha, \gamma(\alpha)\} + (P - \varepsilon(\gamma)\gamma^{-1}P)\{\infty, \alpha\}. \end{aligned}$$

The second equality in the statement of the proposition now follows easily. ∎

In the classical case of weight two and trivial character, the "error term" $(P - \varepsilon(\gamma)\gamma^{-1}P)\{\infty, \alpha\}$ vanishes. In general this term does not vanish. However, we can suitably modify the formulas found in [Cre97a, 2.10], and still obtain an algorithm for computing period integrals.

Algorithm 10.3.3 (Period Integrals). *INPUT: A matrix $\gamma \in \Gamma_0(N)$, a polynomial $P \in V_{k-2}$ and a cuspidal modular form $f \in S_k(\Gamma_1(N), \varepsilon)$ presented as a q-expansion to some precision.*
OUTPUT: The period integral $\langle g, P\{\infty, \gamma(\infty)\}\rangle$, computed to some precision.

1. Write $\gamma = \left(\begin{smallmatrix} a & b \\ cN & d \end{smallmatrix}\right) \in \Gamma_0(N)$, with $a, b, c, d \in \mathbb{Z}$, and set $\alpha = \frac{-d+i}{cN}$ in Proposition 10.3.2.
2. Replacing γ by $-\gamma$ if necessary, we find that the imaginary parts of α and $\gamma(\alpha) = \frac{a+i}{cN}$ are both equal to the positive number $\frac{1}{cN}$.
3. Use (10.3.3) and Lemma 10.3.1 to compute the period integrals of Proposition 10.3.2.

Remark 10.3.4. I have not specified the precision of the output in terms of the input, which is a *major* problem with this algorithm.

It would be nice to know that the modular symbols of the form $P\{\infty, \gamma(\infty)\}$, for $P \in V_{k-2}$ and $\gamma \in \Gamma_0(N)$ generate a large subspace of $\mathbb{M}_k(\Gamma_1(N), \varepsilon) \otimes \mathbb{Q}$. When $k = 2$ and $\varepsilon = 1$, Manin proved in [Man72], that the map $\Gamma_0(N) \to H_1(X_0(N), \mathbb{Z})$ sending γ to $\{0, \gamma(0)\}$ is a surjective group homomorphism. When $k > 2$, I have not found any similar group-theoretic statement. However, we have the following theorem.

Theorem 10.3.5. *Any element of $\mathbb{S}_k(\Gamma_1(N), \varepsilon)$ can be written in the form*

$$\sum_{i=1}^{n} P_i\{\infty, \gamma_i(\infty)\}$$

for some $P_i \in V_{k-2}$ and $\gamma_i \in \Gamma_0(N)$. Moreover, P_i and γ_i can be chosen so that $\sum P_i = \sum \varepsilon(\gamma_i)\gamma_i^{-1}(P_i)$, so the error term in (10.3.6) vanishes.

The author and Helena Verrill prove this theorem in [SV01]. See also [[what that Edixhoven student is writing up...]] The condition that the error term vanishes, means that one can replace ∞ by any α in the expression for the modular symbol and obtain an equivalent modular symbol. For this reason, we call such modular symbols *transportable*, as illustrated in Figure 10.3.1.

Note that in general not every element of the form $P\{\infty, \gamma(\infty)\}$ must lie in $\mathbb{S}_k(N, \varepsilon)$. However, if $\gamma P = P$ then $P\{\infty, \gamma(\infty)\}$ does lie in $\mathbb{S}_k(N, \varepsilon)$. It would be interesting to know under what circumstances $\mathbb{S}_k(N, \varepsilon)$ is generated by symbols of the form $P\{\infty, \gamma(\infty)\}$ with $\gamma P = P$. This sometimes fails for k odd; for example, when $k = 3$ the condition $\gamma P = P$ implies that $\gamma \in \Gamma_0(N)$ has an eigenvector with eigenvalue 1, hence is of finite order. When k is even the author can see no obstruction to generating $\mathbb{S}_k(N, \varepsilon)$ using such symbols.

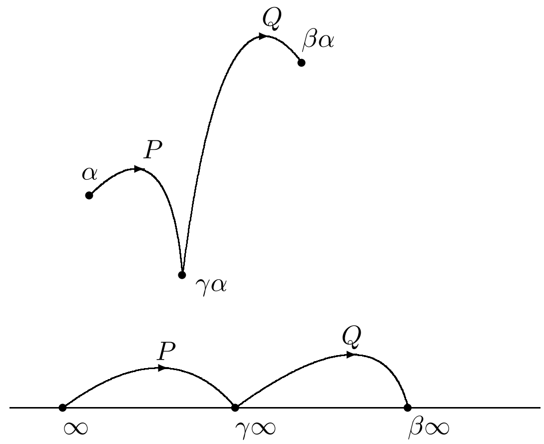

Figure 10.3.1: "Transporting" a transportable modular symbol.

10.4 Speeding Convergence Using the Atkin-Lehner Operator

Let $w_N = \left(\begin{smallmatrix} 0 & -1 \\ N & 0 \end{smallmatrix}\right) \in M_2(\mathbb{Z})$. Consider the Atkin-Lehner involution W_N on $M_k(\Gamma_1(N))$, which is defined by

$$\begin{aligned} W_N(f) &= N^{(2-k)/2} \cdot f|_{[w_N]_k} \\ &= N^{(2-k)/2} \cdot f\left(-\frac{1}{Nz}\right) \cdot N^{k-1} \cdot (Nz)^{-k} \\ &= N^{-k/2} \cdot z^{-k} \cdot f\left(-\frac{1}{Nz}\right). \end{aligned}$$

Here we take the positive square root if k is odd. Then $W_N^2 = (-1)^k$ is an involution when k is even.

There is an operator on modular symbols, which we also denote W_N, which is given by

$$\begin{aligned} W_N(P\{\alpha,\beta\}) &= N^{(2-k)/2} \cdot w_N(P)\{w_N(\alpha), w_N(\beta)\} \\ &= N^{(2-k)/2} \cdot P(-Y, NX)\left\{-\frac{1}{\alpha N}, -\frac{1}{\beta N}\right\}, \end{aligned}$$

and one has that if $f \in S_k(\Gamma_1(N))$ and $x \in \mathbb{M}_k(\Gamma_1(N))$, then

$$\langle W_N(f), x\rangle = \langle f, W_N(x)\rangle.$$

If ε is a Dirichlet character mod N, then the operator W_N sends $S_k(\Gamma_1(N),\varepsilon)$ to $S_k(\Gamma_1(N),\overline{\varepsilon})$. Thus if $\varepsilon^2 = 1$, then W_N preserves $S_k(\Gamma_1(N),\varepsilon)$. In particular, W_N acts on $S_k(\Gamma_0(N))$.

The follow proposition shows how to compute the pairing $\langle f, P\{\infty, \gamma(\infty)\}\rangle$ under certain restrictive assumptions. It generalizes a result of [Cre97b] to higher weight.

Proposition 10.4.1. *Let $f \in S_k(\Gamma_1(N), \varepsilon)$ be a cusp form which is an eigenform for the Atkin-Lehner operator W_N having eigenvalue $w \in \{\pm 1\}$ (thus $\varepsilon^2 = 1$ and k is even). Then for any $\gamma \in \Gamma_0(N)$ and any $P \in V_{k-2}$, with the property that $\gamma P = \varepsilon(\gamma)P$, we have the following formula, valid for any $\alpha \in \mathfrak{h}$:*

$$\langle g, P\{\infty, \gamma(\infty)\}\rangle = \Big\langle g, \quad w\frac{P(Y,-NX)}{N^{k/2-1}}\{w_N(\alpha), \infty\} + \left(P - w\frac{P(Y,-NX)}{N^{k/2-1}}\right)\left\{i/\sqrt{N}, \infty\right\} - P\left\{\gamma(\alpha), \infty\right\}\Big\rangle.$$

Here $w_N(\alpha) = -\dfrac{1}{N\alpha}$.

Proof. By Proposition 10.3.2 our condition on P implies that $P\{\infty, \gamma(\infty)\} = P\{\alpha, \gamma(\alpha)\}$. The steps of the following computation are described below.

$$\begin{aligned}
&\Big\langle f, \quad P\{\alpha, \gamma(\alpha)\}\Big\rangle \\
&\quad = \Big\langle f, \quad P\{\alpha, i/\sqrt{N}\} + P\{i/\sqrt{N}, W(\alpha)\} + P\{W(\alpha), \gamma(\alpha)\}\Big\rangle \\
&\quad = \Big\langle f, \quad w\frac{W(P)}{N^{k/2-1}}\{W(\alpha), i/\sqrt{N}\} + P\{i/\sqrt{N}, W(\alpha)\} + P\{W(\alpha), \gamma(\alpha)\}\Big\rangle \\
&\quad = \Big\langle f, \quad \left(w\frac{W(P)}{N^{k/2-1}} - P\right)\{W(\alpha), i/\sqrt{N}\} + P\{W(\alpha), \infty\} - P\{\gamma(\alpha), \infty\}\Big\rangle \\
&\quad = \Big\langle f, \quad w\frac{W(P)}{N^{k/2-1}}\{W(\alpha), \infty\} + \left(P - w\frac{W(P)}{N^{k/2-1}}\right)\{i/\sqrt{N}, \infty\} - P\{\gamma(\alpha), \infty\}\Big\rangle.
\end{aligned}$$

For the first step, we break the path into three paths. In the second step, we apply the W-involution to the first term, and use that the action of W is compatible with the pairing $\langle\, ,\, \rangle$ and that f is an eigenvector with eigenvalue w. The third step involves combining the first two terms and breaking up the third. In the final step, we replace $\{W(\alpha), i/\sqrt{N}\}$ by $\{W(\alpha), \infty\} + \{\infty, i/\sqrt{N}\}$ and regroup. $\square$

A good choice for α is $\alpha = \gamma^{-1}\left(\frac{b}{d} + \frac{i}{d\sqrt{N}}\right)$, so that $W(\alpha) = \frac{c}{d} + \frac{i}{d\sqrt{N}}$. This maximizes the minimum of the imaginary parts of α and $W(\alpha)$, which results in series that converge more quickly.

Let $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \Gamma_0(N)$. The polynomial

$$P(X,Y) = (cX^2 + (d-a)XY - bY^2)^{\frac{k-2}{2}}$$

satisfies $\gamma(P) = P$. We obtained this formula by viewing V_{k-2} as the $(k-2)$th symmetric product of the two-dimensional space on which $\Gamma_0(N)$ acts naturally. For example, observe that since $\det(\gamma) = 1$ the symmetric product of two eigenvectors for γ is an eigenvector in V_2 having eigenvalue 1. For the same reason, if $\varepsilon(\gamma) \neq 1$, there need not be a polynomial $P(X,Y)$ such that $\gamma(P) = \varepsilon(\gamma)P$. One remedy is to choose another γ so that $\varepsilon(\gamma) = 1$.

Since the imaginary parts of the terms $i/\sqrt{N}$, α and $W(\alpha)$ in the proposition are all relatively large, the sums appearing at the beginning of Section 10.3 converge quickly if d is small. It is **extremely** important to choose γ in Proposition 10.4.1 with d small, otherwise the series will converge very slowly.

Remark 10.4.2. There should be a generalization of Proposition 10.4.1 without the restrictions that $\varepsilon^2 = 1$ and k is even. I would love to include something like this in the final version of this book. Student project?

10.4.1 Another Atkin-Lehner Trick

Suppose E is an elliptic curve and let $L(E, s)$ be the corresponding L-function. Let $\varepsilon \in \{\pm 1\}$ be the root number of E, i.e., the sign of the functional equation for $L(E, s)$, so $\Lambda(E, s) = \varepsilon\Lambda(E, 2-s)$, where $\Lambda(E, s) = N^{s/2}(2\pi)^{-s}\Gamma(s)L(E, s)$. Let $f = f_E$ be the modular form associated to E. If $W_N(f) = wf$, then $\varepsilon = -w$ (see Exercise 10.2). We have

$$\begin{aligned}
L(E,1) &= -2\pi \int_0^{i\infty} f(z)\,dz \\
&= -2\pi i\,\langle f,\, \{0,\infty\}\rangle \\
&= -2\pi i\,\left\langle f, \{0, i/\sqrt{N}\} + \{i/\sqrt{N}, \infty\}\right\rangle \\
&= -2\pi i\,\left\langle wf, \{w_N(0), w_N(i/\sqrt{N})\} + \{i/\sqrt{N}, \infty\}\right\rangle \\
&= -2\pi i\,\left\langle wf, \{\infty, i/\sqrt{N}\} + \{i/\sqrt{N}, \infty\}\right\rangle \\
&= -2\pi i\,(w-1)\left\langle f, \{\infty, i/\sqrt{N}\}\right\rangle
\end{aligned}$$

If $w = 1$, then $L(E, 1) = 0$. If $w = -1$, then

$$L(E,1) = 4\pi i \left\langle f, \{\infty, i/\sqrt{N}\}\right\rangle = 2\sum_{n=1}^{\infty} \frac{a_n}{n} e^{-2\pi n/\sqrt{N}}. \tag{10.4.1}$$

For much more about computing with L-functions of elliptic curves, including a trick for computing ε quickly without directly computing W_N, see [Coh93, §7.5] and [Cre97a, §2.11]. One can also find higher derivatives $L^{(r)}(E, 1)$ by a formula similar to (10.4.1) (see [Cre97a, §2.13]).

10.5 Computing the Period Mapping

Fix a newform $f = \sum a_n q^n \in S_k(\Gamma)$, where $\Gamma_1(N) \subset \Gamma$ for some N. Let $I = I_f \subset \mathbb{T}$ be the kernel of the ring homomorphism $\mathbb{T} \to K_f = \mathbb{Q}(a_2, \ldots)$ that sends T_n to a_n. Let Θ_f be the *rational period mapping* associated to f and Φ_f the period mapping associated to the $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$-conjugates of f, so we have a commutative diagram

$$\begin{tikzcd} \mathbb{M}_k(\Gamma)_{\mathbb{Q}} \arrow[rr, "\Phi_f"] \arrow[dr, "\Theta_f"'] & & \mathrm{Hom}_{\mathbb{C}}(S_k(\Gamma)[I], \mathbb{C}) \\ & \dfrac{\mathbb{M}_k(\Gamma)_{\mathbb{Q}}}{\mathrm{Ker}(\Phi_f)} \arrow[ur, hook, "i_f"] & \end{tikzcd}$$

Recall that the cokernel of Φ_f is the complex torus $A_f(\mathbb{C})$.

The Hecke algebra $\mathbb{T}$ acts on the linear dual

$$\mathbb{M}_k(\Gamma)^* = \mathrm{Hom}(\mathbb{M}_k(\Gamma), \mathbb{Q})$$

by $(t.\varphi)(x) = \varphi(tx)$. Since f is a newform, if $\theta_1, \ldots, \theta_d$ is a basis for $\mathbb{M}_k(\Gamma)^*_{\mathbb{Q}}[I]$, then

$$\mathrm{Ker}(\Phi_f) = \mathrm{Ker}(\theta_1) \oplus \cdots \oplus \mathrm{Ker}(\theta_d).$$

Thus we can compute $\mathrm{Ker}(\Phi_f)$, hence Θ_f, so to compute Φ_f we only need to compute i_f.

Let $g_1, \ldots, g_d$ be a basis for the $\mathbb{Q}$-vector space

$$S_k(\Gamma; \mathbb{Q})[I] = S_k(\Gamma) \cap \mathbb{Q}[[q]],$$

i.e., the space of cusp forms with rational Fourier expansion. We will compute Φ_f with respect to the basis of $\mathrm{Hom}_{\mathbb{Q}}(S_k(\Gamma; \mathbb{Q})[I], \mathbb{C})$ dual to this basis. Choose elements $x_1, \ldots, x_d \in \mathbb{M}_k(\Gamma)$ with the following properties:

1. Using Proposition 10.3.2 or Proposition 10.4.1 it is possible to compute the period integrals $\langle g_i, x_j \rangle$, $i, j \in \{1, \ldots d\}$ efficiently.

2. The $2d$ elements $v + \eta(v)$ and $v - \eta(v)$ for $v = \Theta_f(x_1), \ldots, \Theta_f(x_d)$ span a space of dimension $2d$ (i.e., they span $\mathbb{M}_k(\Gamma)/\mathrm{Ker}(\Phi_f)$).

Given this data, we can compute

$$i_f(v + \eta(v)) = 2\mathrm{Re}(\langle g_1, x_i \rangle, \ldots, \langle g_d, x_i \rangle)$$

and

$$i_f(v - \eta(v)) = 2i\mathrm{Im}(\langle g_1, x_i \rangle, \ldots, \langle g_d, x_i \rangle).$$

We break the integrals into real and imaginary parts because this increases the precision of our answers. Since the vectors $v_n + \eta(v_n)$ and $v_n - \eta(v_n)$, $n = 1, \ldots, d$ span $\mathbb{M}_k(N, \varepsilon)_{\mathbb{Q}}/\mathrm{Ker}(\Phi_f)$, we have computed i_f.

Remark 10.5.1. We want to find symbols x_i satisfying the conditions of Proposition 10.4.1. This is usually possible when d is very small, but in practice I have had problems doing this when d is large.

Remark 10.5.2. The above strategy was motivated by [Cre97a, §2.10].

Remark 10.5.3. The following idea just occured to me. We could use that $\langle T_n(g), x\rangle = \langle g, T_n(x)\rangle$ for any Hecke operator T_n, so that we only need to compute the period integrals $\langle g, x_i\rangle$. Then we obtain all pairings $\langle T_n(g), x_i\rangle = \langle g, T_n(x_i)\rangle$. Since the $T_n(g)$ span the simple $\mathbb{T}$-module $S_k(\Gamma;\mathbb{Q})[I]$, this must give all pairings. However, it requires computing only $2d$ pairings instead of $2d^2$ pairings, which is potentially a huge savings when d is large.

10.6 Computing Elliptic Curves of Given Conductor

10.6.1 Using Modular Symbols

Using modular symbols and the period map, we can compute all elliptic curves over $\mathbb{Q}$ of conductor N, up to isogeny. The algorithm in this section gives all *modular elliptic curves*, i.e., elliptic curves attached to modular forms, of conductor N. Fortunately, it is now known by [Wil95, BCDT01, TW95] that every elliptic curve over $\mathbb{Q}$ is modular, so the procedure described in this section gives all elliptic curves, up to isogeny, of given conductor. I think this algorithm was first introduced by Tingly (??), and later refined by Cremona [Cre97a].

Algorithm 10.6.1 (Elliptic Curves of Conductor N). *INPUT: A positive integer N.*
OUTPUT: A list of Weierstrass equations for the elliptic curves of conductor N, up to isogeny.

1. [Compute Modular Symbols] Compute $\mathbb{M}_2(\Gamma_0(N))$ using Section 8.6.
2. [Find Rational Eigenspaces] Find the two-dimensional eigenspaces V in $\mathbb{M}_2(\Gamma_0(N))_{\mathrm{new}}$ that correspond to elliptic curves. Do *not* use the decomposition algorithms from Section 7.4, which are too complicated, and give way more information than we need. Instead, for the first few primes $p \nmid N$, compute all eigenspaces $\mathrm{Ker}(T_p - a)$, where a runs through integers with $-2\sqrt{p} < a < 2\sqrt{p}$. Intersect these eigenspaces to find the eigenspaces that correspond to elliptic curves. To find just the new ones, either compute the degeneracy maps to lower level, or find all the rational eigenspaces of all levels that strictly divide N and exclude them.
3. [Find Rational Newforms] Using Algorithm 9.4.2, find each rational newform $f = \sum_{n=1}^{\infty} a_n q^n \in \mathbb{Z}[[q]]$ associated to each eigenspace V found in Step 2.
4. [Find Each Curve] For each rational eigenvector f found in Step 3, do the following:

(a) [Period Lattice] Compute the corresponding period lattice $\Lambda = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ by computing the image of Φ_f, as described in Section 10.5.

(b) [Compute τ] Let $\tau = \omega_1/\omega_2$. If $\mathrm{Im}(\tau) < 0$, swap ω_1 and ω_2, so $\mathrm{Im}(\tau) > 0$. By successively applying generators of $\mathrm{SL}_2(\mathbb{Z})$, we find an $\mathrm{SL}_2(\mathbb{Z})$ equivalent element τ' in the standard fundamental domain, so $|\mathrm{Re}(\tau')| \leq 1/2$ and $|\tau| \geq 1$.

(c) [c-invariants] Compute the invariants c_4 and c_6 of the lattice Λ using the following rapidly convergent series:

$$c_4 = \left(\frac{2\pi}{\omega_2}\right)^4 \cdot \left(1 + 240 \sum_{n=1}^{\infty} \frac{n^3 q^n}{1-q^n}\right)$$
$$c_6 = \left(\frac{2\pi}{\omega_2}\right)^6 \cdot \left(1 - 504 \sum_{n=1}^{\infty} \frac{n^5 q^n}{1-q^n}\right),$$

where $q = e^{2\pi i \tau'}$, where τ' is as in Step 4b. A theorem of Edixhoven (that the Manin constant is an integer) implies that the invariants c_4 and c_6 of Λ are integers, so it is only necessary to compute Λ to large precision to determine them.

(d) [Elliptic Curve] An elliptic curve with invariants c_4 and c_6 is

$$E: \quad y^2 = x^3 - \frac{c_4}{48}x - \frac{c_6}{864}.$$

(e) [Prove Correctness] Compute the conductor of E. If the conductor of E is not N, then recompute c_4 and c_6 using a larger precision everywhere (e.g., more terms of f, reals to larger precision, etc.) If the conductor is N, compute the coefficients b_p of the modular form $g = g_E$ attached to the elliptic curve E, for $p \leq \#\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})/6$. Verify that $a_p = b_p$, where a_p are the coefficients of f. If this equality holds, then E must be isogenous to the elliptic curve attached to f, by the Sturm bound (Theorem 11.1.2) and Faltings's isogeny theorem. If the equality fails for some p, recompute c_4 and c_6 to larger precision.

There are a couple of tricks to optimize the above algorithm. For example, one can work separately with $\mathbb{M}_k(\Gamma_0(N))^+_{\mathrm{new}}$ and $\mathbb{M}_k(\Gamma_0(N))^-_{\mathrm{new}}$ and get enough information to find E, up to isogeny (see [Cre97b]).

Once we have one curve from each isogeny class of curves of conductor N, we can find each curve in each isogeny class, hence all curves of conductor N. If $E/\mathbb{Q}$ is an elliptic curve, then any curve isogenous to E is isogenous via a chain of isogenous of prime degree. There is an *a priori* bound on the degrees of these isogenous due to Mazur. Also, there are various methods for finding all isogenous from E of a given fixed degree. See [Cre97a, §3.8] for more details.

10.6.2 Finding Curves by Finding S-Integral Points

Cremona and others have recently been systematically developing an alternative complementary approach to the problem of computing all elliptic curves of given conductor (see [CL04]). Instead of computing all curves of given conductor, we instead consider the seemingly more difficult problem of find all curves with good reduction outside a finite set S of primes. Since one can compute the conductor of a curve using Tate's algorithm [Tat75, Cre97a, §3.2], if we know all curves with good reduction outside S, we can find all curves of conductor N by letting S be the set of prime divisors of N.

There is a strategy for finding all curves with good reduction outside S. It is not a provably-correct algorithm, in the sense that it is always guarenteed to terminate (the modular symbols method above *is* an algorithm), but in practice it often works, and I think one conjectures that it always does. Also, this strategy makes sense over any number field, whereas the modular symbols method does not, though there are generalizations of modular symbols to other number fields.

Fix a finite set S of primes of a number field K. It is a theorem of Shafarevich that there are only finitely many elliptic curves with good reduction outside S (see [Sil92, §IX.6]). His proof uses that the group of S-units in K is finite, and Siegel's theorem that there are only finitely many S-integral points on an elliptic curve. One can make all this explicit, and sometimes in practice one can compute all these S-integral points.

The problem of finding all elliptic curves with good reduction outside of S can be broken into several subproblems, the main ones being:

1. Determine the following finite subgroup of $K^*/(K^*)^m$:
$$K(S,m) = \{x \in K^*/(K^*)^m \,:\, m \mid \mathrm{ord}_{\mathfrak{p}}(x) \text{ all } \mathfrak{p} \notin S\}.$$
2. Find all S-integral points on certain elliptic curves $y^2 = x^3 + k$.

In [CL04], there is one example, where he finds all curves of conductor $N = 2^8 \cdot 17^2 = 73984$ by finding all curves with good reduction outside $\{2, 17\}$. He finds 32 curves of conductor 73984 that divide into 16 isogeny classes. He remarks that $\dim S_2(\Gamma_0(N)) = 9577$, and his modular symbols program was not able to find these curves at this high of level (presumably due to memory constraints?).

10.7 Examples

10.7.1 Jacobians of genus-two curves

The author is among the the six authors of [FpS$^+$01], who gather empirical evidence for the BSD conjecture for Jacobian of genus two curves. Of the 32 Jacobians considered, all but four are optimal quotients of $J_0(N)$ for some N. The methods of this section can be used to compute Ω_f^+ for the Jacobians of these 28 curves. Using explicit models for the genus two curves, the authors of [FpS$^+$01] computed the measure of A with respect to a basis for the Néron differentials of A. In all 28 cases our answers agreed to the precision computed. Thus in these cases we have numerically verified that the Manin constant equals 1.

The first example considered in [FpS+01] is the Jacobian $A = J_0(23)$ of the modular curve $X_0(23)$. This curve has as a model

$$y^2 + (x^3 + x + 1)y = -2x^5 - 3x^2 + 2x - 2$$

from which one can compute the BSD $\Omega_A = 2.7328....$ The following is an integral basis of cusp forms for $S_2(23)$.

$$\begin{aligned} g_1 &= q - q^3 - q^4 - 2q^6 + 2q^7 + \cdots \\ g_2 &= q^2 - 2q^3 - q^4 + 2q^5 + q^6 + 2q^7 + \cdots \end{aligned}$$

The space $\mathbb{M}_2(23;\mathbb{Q})$ of modular symbols has dimension five and is spanned by $\{-1/19,0\}$, $\{-1/17,0\}$, $\{-1/15,0\}$, $\{-1/11,0\}$ and $\{\infty,0\}$. The submodule $\mathbb{S}_2(23;\mathbb{Z})$ has rank four and has as basis the first four of the above five symbols. Choose $\gamma_1 = \left(\begin{smallmatrix} 8 & 1 \\ 23 & 3 \end{smallmatrix}\right)$ and $\gamma_2 = \left(\begin{smallmatrix} 6 & 1 \\ 23 & 4 \end{smallmatrix}\right)$ and let $x_i = \{\infty, \gamma_i(\infty)\}$. Using the W_N-trick (see Section 10.4) we compute the period integrals $\langle g_i, x_j\rangle$ using 97 terms of the q-expansions of g_1 and g_2, and obtain

$$\begin{array}{ll} \langle g_1, x_1\rangle \sim -1.3543 + 1.0838i, & \langle g_1, x_2\rangle \sim -0.5915 + 1.6875i \\ \langle g_2, x_1\rangle \sim -0.5915 - 0.4801i, & \langle g_2, x_2\rangle \sim -0.7628 + 0.6037i \end{array}$$

Using 97 terms we already obtain about 14 decimal digits of accuracy, but we do not reproduce them all here. We next find that

$$\langle g_1, x_1 + x_1^*\rangle \sim 2\mathrm{Re}(-1.3543 + 1.0838i) = 2.7086,$$

and so on. Upon writing each generator of $\mathbb{S}_2(23)$ in terms of $x_1 + x_1^*$, $x_1 - x_1^*$, $x_2 + x_2^*$ and $x_2 - x_2^*$ we discover that the period mapping with respect to the basis dual to g_1 and g_2 is (approximately)

$$\begin{array}{lcll} \{-1/19,0\} & \mapsto & (\ \ 0.5915 - 1.6875i, & 0.7628 - 0.6037i) \\ \{-1/17,0\} & \mapsto & (-0.5915 - 1.6875i, & -0.7628 - 0.6037i) \\ \{-1/15,0\} & \mapsto & (-1.3543 - 1.0838i, & -0.5915 + 0.4801i) \\ \{-1/11,0\} & \mapsto & (-1.5256, & 0.3425) \end{array}$$

Working in $\mathbb{S}_2(23)$ we find $\mathbb{S}_2(23)^+$ is spanned by $\{-1/19,0\} - \{-1/17,0\}$ and $\{-1/11,0\}$. There is only one real component so

$$\Omega_I^+ \sim \left| \begin{array}{cc} 1.1831 & 1.5256 \\ -1.5256 & 0.3425 \end{array} \right| = 2.7327...$$

To greater precision we find that $\Omega_f^+ \sim 2.7327505324965$. This agrees with the value in [FpS+01]; since the Manin constant is an integer, it must equal 1.

10.7.2 Level one cusp forms

In the following two sections we consider several specific examples of tori attached to modular forms of weight greater than two.

Table 10.7.1: Volumes associated to level one cusp forms.

k	Ω^+	Ω^-
12	0.002281474899	$0.000971088287i$
16	0.003927981492	$0.000566379403i$
18	0.000286607497	$0.023020042428i$
20	0.008297636952	$0.0005609325015i$
22	0.002589288079	$0.0020245743816i$
24	0.000000002968	$0.0000000054322i$
26	0.003377464512	$0.3910726132671i$
28	0.000000015627	$0.0000000029272i$

Let $k \geq 12$ be an even integer. Associated to each Galois conjugacy class of normalized eigenforms f, there is a torus A_f over $\mathbb{R}$. The real and minus volume of the first few of these tori are displayed in Table 10.7.1. For weights 24 and 28 we give Ω^-/i so that the columns will line up nicely. In each case, 97 terms of the q-expansion were used.

The volumes appear to be *much* smaller than the volumes of weight two abelian varieties. The dimension of each A_f is 1, except for weights 24 and 28 when the dimension is 2.

10.7.3 CM elliptic curves of weight greater than two

Let f be a rational newform with "complex multiplication", in the sense that "half" of the Fourier coefficients of f are zero. For our purposes, it is not necessary to define complex multiplication any more precisely. Experimentally, it appears that the associated elliptic A_f has rational j-invariant. As evidence for this we present Table 10.7.2, which includes the analytic data about every rational CM form of weight four and level ≤ 197. The computations of Table 10.7.2 were done using at least 97 terms of the q-expansion of f. The rationality of j could probably be proved by observing that the CM forces A_f to have extra automorphisms.

In these examples, the invariants c_4 and c_6 are mysterious (to me); in contrast, in weight 2 the invariants of an elliptic curve are known to be integers (see [Cre97a, 2.14]).

10.8 Exercises

10.1 Let $f \in S_k(\Gamma_1(N))$ be a newform, and let V_f be the subspace spanned by the $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ conjugates of f. Let $V_f^\perp$ be the Petersson complement of V_f in $S_k(\Gamma_1(N))$.

(a) Show that Atkin-Lehner-Li theory and properties of the Petersson innner product imply that $V_f^\perp$ is stable under the full Hecke algebra

Table 10.7.2: CM elliptic curves of weight > 2.

E	j	Ω^+	Ω^-	c_4	c_6
9k4A	0	0.2095	$0.1210i$	0.0000	−56626421686.2951
32k4A	1728	0.2283	$0.2283i$	−3339814.8874	0.0000
64k4D	1728	0.1614	$0.1614i$	53437038.1988	0.0000
108k4A	0	0.0440	$0.0762i$	−14699.2655	24463608892439.7456
108k4C	0	0.0554	$0.0960i$	1608.7743	6115643810955.1724
121k4A	-2^{15}	0.0116	$0.0385i$	85659519816.8841	25723073306989527.1216
144k4E	0	0.0454	$0.0262i$	81.1130	−549788016394046.1396
27k6A	0	0.0110	$0.0191i$	0.0000	97856189971744203.7795
32k6A	1728	0.0199	$0.0199i$	−58095643136.7658	8.0094

$\mathbb{T} \subset S_k(\Gamma_1(N))$.

(b) (*) Give an example of $f \in S_2(\Gamma_1(N))$ that shows that $V_f^\perp$ need not be $\mathbb{T}$-stable if f is not a newform. [Hint: Argue that if $V_f^\perp$ is $\mathbb{T}$-stable for any f, then every element of $\mathbb{T}$ is diagonalizable. An example of a space where T_3 is not diagonalizable is $S_2(\Gamma_1(81))$ (you may assume this).]

10.2 Suppose $f \in S_2(\Gamma_0(N))$ is a newform and that $W_N(f) = wf$. Let $\Lambda(E,s) = N^{s/2}(2\pi)^{-s}\Gamma(s)L(E,s)$. Prove that

$$\Lambda(E,s) = -w\Lambda(E, 2-s).$$

[Hint: Show that $\Lambda(f,s) = \int_{0,\infty} f(iy/\sqrt{N})y^{s-1}\,dy$, then substitute $1/y$ for y. If you get completely stuck, see any of many standard references, e.g., [Cre97a, §2.8].]

Chapter 11

Congruences

11.1 Congruences Between Modular Forms

In this section we develop theory for determining when modular forms are congruent, which is extremely import for computing with modular forms.

Let Γ be an arbitrary congruence subgroup of $\mathrm{SL}_2(\mathbb{Z})$, and suppose $f \in M_k(\Gamma)$ is a modular form of integer weight k for Γ. Since $\left(\begin{smallmatrix} 1 & N \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma$ for some integer N, the form f has a Fourier expansion in nonnegative powers of $q^{1/N}$. For a rational number n, let $a_n(f)$ be the coefficient of q^n in the Fourier expansion of f. Put

$$\mathrm{ord}_q(f) = \min\{n \in \mathbb{Q} : a_n \neq 0\},$$

where by convention we take $\min \emptyset = +\infty$, so $\mathrm{ord}_q(0) = +\infty$.

11.1.1 The j-invariant

Let

$$j = \frac{1}{q} + 744 + 196884q + \cdots$$

be the j-function, which is a weight 0 modular function that is holomorphic except for a simple pole at ∞ and has integer Fourier coefficients (see, e.g., [Ser73, §VIII.3.3]).

Lemma 11.1.1. *Suppose g is a weight 0 level 1 modular function that is holomorphic except possibly with a pole of order n at ∞. Then g is a polynomial in j of degree at most n. Moreover, the coefficients of this polynomial lie in the ideal I generated by the coefficients $a_m(g)$ with $m \leq 0$.*

Proof. If $n = 0$, then $g \in M_0(\mathrm{SL}_2(\mathbb{Z})) = \mathbb{C}$, so g is constant with constant term in I, so the statement is true. Next suppose $n > 0$ and the lemma has been proved for all functions with smaller order poles. Let $\alpha = a_n(g)$, and note that

$$\mathrm{ord}_q(g - \alpha j^n) = \mathrm{ord}_q\left(g - \alpha \cdot \left(\frac{1}{q} + 744 + 196884q + \cdots\right)^n\right) > -n.$$

Thus by induction $h = g - \alpha j^n$ is a polynomial in j of degree $< n$ with coefficients in the ideal generated by the coefficients $a_m(g)$ with $m < 0$. It follows that $g = \alpha \cdot j^n - h$ satisfies the conclusion of the lemma. $\square$

11.1.2 Congruences for Modular Forms

If $\mathcal{O}$ is the ring of integers of a number field, $\mathfrak{m}$ is a maximal ideal of $\mathcal{O}$, and $f = \sum a_n q^n \in \mathcal{O}[[q^{1/N}]]$ for some integer N, let

$$\mathrm{ord}_{\mathfrak{m}}(f) = \mathrm{ord}_q(f \mod \mathfrak{m}) = \min\{n \in \mathbb{Q} : a_n \not\in \mathfrak{m}\}.$$

Note that $\mathrm{ord}_{\mathfrak{m}}(fg) = \mathrm{ord}_{\mathfrak{m}}(f) + \mathrm{ord}_{\mathfrak{m}}(g)$. The following theorem was first proved in [Stu87], and our proof is an expanded version of the one in [Stu87].

Theorem 11.1.2 (Sturm). *Let $\mathfrak{m}$ be a prime ideal in the ring of integers $\mathcal{O}$ of a number field K, and let Γ be a congruence subgroup of $\mathrm{SL}_2(\mathbb{Z})$ of index m and level N. Suppose $f \in M_k(\Gamma, \mathcal{O})$ is a modular form and*

$$\mathrm{ord}_{\mathfrak{m}}(f) > \frac{km}{12}$$

or $f \in S_k(\Gamma, \mathcal{O})$ is a cusp form and

$$\mathrm{ord}_{\mathfrak{m}}(f) > \frac{km}{12} - \frac{m-1}{N}.$$

Then $f \equiv 0 \pmod{\mathfrak{m}}$.

Proof. **Case 1: First we assume $\Gamma = \mathrm{SL}_2(\mathbb{Z})$.**
Let

$$\Delta = q + 24q^2 + \cdots \in S_{12}(\mathrm{SL}_2(\mathbb{Z}), \mathbb{Z})$$

be the Δ function. Since $\mathrm{ord}_{\mathfrak{m}}(f) > k/12$, we have $\mathrm{ord}_{\mathfrak{m}}(f^{12}) > k$. We have

$$\mathrm{ord}_q(f^{12} \cdot \Delta^{-k}) = 12 \cdot \mathrm{ord}_q(f) - k \cdot \mathrm{ord}_q(\Delta) \geq -k, \tag{11.1.1}$$

since f is holomorphic at infinity and Δ has a zero of order 1. Also

$$\mathrm{ord}_{\mathfrak{m}}(f^{12} \cdot \Delta^{-k}) = \mathrm{ord}_{\mathfrak{m}}(f^{12}) - k \cdot \mathrm{ord}_{\mathfrak{m}}(\Delta) > k - k = 0. \tag{11.1.2}$$

Combining (11.1.1) and (11.1.2), we see that

$$f^{12} \cdot \Delta^{-k} = \sum_{n \geq -k} b_n q^n,$$

with $b_n \in \mathcal{O}$ and $b_n \in \mathfrak{m}$ if $n \leq 0$.

By Lemma 11.1.1,

$$f^{12} \cdot \Delta^{-k} \in \mathfrak{m}[j]$$

is a polynomial in j of degree at most k with coefficients in $\mathfrak{m}$. Thus

$$f^{12} \in \mathfrak{m}[j] \cdot \Delta^k,$$

so since the coefficients of Δ are integers, every coefficient of f^{12} is in $\mathfrak{m}$. Thus $\mathrm{ord}_{\mathfrak{m}}(f^{12}) = +\infty$, hence $\mathrm{ord}_{\mathfrak{m}}(f) = +\infty$, so $f = 0$, as claimed.

Case 2: Γ Arbitrary

Let N be such that $\Gamma(N) \subset \Gamma$, so also $f \in M_k(\Gamma(N))$. If $g \in M_k(\Gamma(N))$ is arbitrary, then because $\Gamma(N)$ is a normal subgroup of $\mathrm{SL}_2(\mathbb{Z})$, we have that for any $\gamma \in \Gamma(N)$ and $\delta \in \mathrm{SL}_2(\mathbb{Z})$, that

$$(g|[\delta]_k)|[\gamma]_k = g|[\delta\gamma]_k = g|[\gamma'\delta]_k = g|[\gamma']_k|[\delta]_k = g|[\delta]_k,$$

where $\delta' \in \mathrm{SL}_2(\mathbb{Z})$. Thus for any $\delta \in \mathrm{SL}_2(\mathbb{Z})$, we have that $g|[\delta]_k \in M_k(\Gamma(N))$, so $\mathrm{SL}_2(\mathbb{Z})$ acts on $M_k(\Gamma(N))$.

It is a standard (but nontrivial) fact about modular forms, which comes from the geometry of the modular curve $X(N)$ over $\mathbb{Q}(\zeta_N)$ and $\mathbb{Z}[\zeta_N]$, that $M_k(\Gamma(N))$ has a basis with Fourier expansions in $\mathbb{Z}[\zeta_N][[q^{1/N}]]$, and that the action of $\mathrm{SL}_2(\mathbb{Z})$ on $M_k(\Gamma(N))$ preserves

$$M_k(\Gamma(N), \mathbb{Q}(\zeta_N)) = M_k(\Gamma(N)) \cap (\mathbb{Q}(\zeta_N)[[q^{1/N}]]),$$

and the cuspidal subspace $S_k(\Gamma(N), \mathbb{Q}(\zeta_N))$. In particular, for any $\gamma \in \mathrm{SL}_2(\mathbb{Z})$,

$$f|[\gamma]_k \in M_k(\Gamma(N), K(\zeta_N)),$$

Moreover, the denominators of $f|[\gamma]_k$ are bounded, since f is an $\mathcal{O}[\zeta_N]$-linear combination of a basis for $M_k(\Gamma(N), \mathbb{Z}[\zeta_N])$, and the denominators of $f|[\gamma]_k$ divide the product of the denominators of the images of each of these basis vectors under $[\gamma]_k$.

Let $L = K(\zeta_N)$. Let $\mathfrak{M}$ be a prime of $\mathcal{O}_L$ that divides $\mathfrak{m}\mathcal{O}_L$. We will now show that for each $\gamma \in \mathrm{SL}_2(\mathbb{Z})$, the Chinese remainder theorem implies that there is an element $A_\gamma \in L^*$ such that

$$A_\gamma \cdot f|[\gamma]_k \in M_k(\Gamma(N), \mathcal{O}_L) \qquad \text{and} \qquad \mathrm{ord}_{\mathfrak{M}}(A_\gamma \cdot f|[\gamma]_k) < \infty. \tag{11.1.3}$$

First find $A \in L^*$ such that $A \cdot f|[\gamma]_k$ has coefficients in $\mathcal{O}_L$. Choose $\alpha \in \mathfrak{M}$ with $\alpha \notin \mathfrak{M}^2$, and find a negative power α^t such that $\alpha^t \cdot A \cdot f|[\gamma]_k$ has $\mathfrak{M}$-integral coefficients and finite valuation. This is possible because we assumed that f is nonzero. Use the Chinese remainder theorem to find $\beta \in \mathcal{O}_L$ such that $\beta \equiv 1 \pmod{\mathfrak{M}}$ and $\beta \equiv 0 \pmod{\wp}$ for each prime $\wp \neq \mathfrak{M}$ that divides (α). Then for some s we have

$$\beta^s \cdot \alpha^t \cdot A \cdot f|[\gamma]_k = A_\gamma \cdot f|[\gamma]_k \in M_k(\Gamma(N), \mathcal{O}_L),$$

and $\mathrm{ord}_{\mathfrak{M}}(A_\gamma \cdot f|[\gamma]_k) < \infty$.

Write

$$\mathrm{SL}_2(\mathbb{Z}) = \bigcup_{i=1}^{m} \Gamma\gamma_i$$

with $\gamma_1 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, and let

$$F = f \cdot \prod_{i=2}^{m} A_{\gamma_i} \cdot f|[\gamma_i]_k.$$

Then $F \in M_{km}(\mathrm{SL}_2(\mathbb{Z}))$ and since $\mathfrak{M} \cap \mathcal{O}_K = \mathfrak{m}$, we have $\mathrm{ord}_{\mathfrak{M}}(f) = \mathrm{ord}_{\mathfrak{m}}(f)$, so

$$\mathrm{ord}_{\mathfrak{M}}(F) \geq \mathrm{ord}_{\mathfrak{M}}(f) = \mathrm{ord}_{\mathfrak{m}}(f) > \frac{km}{12}.$$

Thus we can apply case 1 to conclude that

$$\mathrm{ord}_{\mathfrak{M}}(F) = +\infty.$$

Thus

$$\infty = \mathrm{ord}_{\mathfrak{M}}(F) = \mathrm{ord}_{\mathfrak{m}}(f) + \sum_{i=2}^{m} \mathrm{ord}_{\mathfrak{M}}(A_{\gamma_i} f|[\gamma]_k), \tag{11.1.4}$$

so $\mathrm{ord}_{\mathfrak{m}}(f) = +\infty$, because of (11.1.3).

We next obtain a better bound when f is a cusp form. Since $|[\gamma]_k$ preserves cusp forms, $\mathrm{ord}_{\mathfrak{M}}(A_{\gamma_i} f|[\gamma]_k) \geq \frac{1}{N}$ for each i. Thus

$$\mathrm{ord}_{\mathfrak{M}}(F) \geq \mathrm{ord}_{\mathfrak{M}}(f) + \frac{m-1}{N} = \mathrm{ord}_{\mathfrak{m}}(f) + \frac{m-1}{N} > \frac{km}{12},$$

since now we are merely assuming that

$$\mathrm{ord}_{\mathfrak{m}}(f) > \frac{km}{12} - \frac{m-1}{N}.$$

Thus we again apply case 1 to conclude that $\mathrm{ord}_{\mathfrak{M}}(F) = +\infty$, and using (11.1.4) conclude that $\mathrm{ord}_{\mathfrak{m}}(f) = +\infty$.

$\square$

Corollary 11.1.3. *Let $\mathfrak{m}$ be a prime ideal in the ring of integers $\mathcal{O}$ of a number field. Suppose $f, g \in M_k(\Gamma, \mathcal{O})$ are modular forms and*

$$a_n(f) \equiv a_n(g) \pmod{\mathfrak{m}}$$

for all

$$n \leq \begin{cases} \frac{km}{12} - \frac{m-1}{N} & \text{if } f - g \in S_k(\Gamma, \mathcal{O}), \\ \frac{km}{12} & \text{otherwise,} \end{cases}$$

where $m = [\mathrm{SL}_2(\mathbb{Z}) : \Gamma]$. Then $f \equiv g \pmod{\mathfrak{m}}$.

Buzzard proved the following corollary, which is extremely useful in practical computations. It asserts that the Sturm bound for modular forms with character is the same as the Sturm bound for $\Gamma_0(N)$.

Corollary 11.1.4 (Buzzard). *Let $\mathfrak{m}$ be a prime ideal in the ring of integers $\mathcal{O}$ of a number field. Suppose $f, g \in M_k(\Gamma_1(N), \varepsilon, \mathcal{O})$ are modular forms with Dirichlet character $\varepsilon : (\mathbb{Z}/N\mathbb{Z})^* \to \mathbb{C}^*$ and assume that*

$$a_n(f) \equiv a_n(g) \pmod{\mathfrak{m}} \qquad \text{for all} \qquad n \leq \frac{km}{12},$$

where

$$m = [\mathrm{SL}_2(\mathbb{Z}) : \Gamma_0(N)] = \#\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z}) = N \cdot \prod_{p \mid N} \left(1 + \frac{1}{p}\right).$$

Then $f \equiv g \pmod{\mathfrak{m}}$.

Proof. Let $h = f - g$ and let $r = km/12$, so $\mathrm{ord}_{\mathfrak{m}}(h) > r$. Let s be the order of the Dirichlet character ε. Then $h^s \in M_{ks}(\Gamma_0(N))$ and

$$\mathrm{ord}_{\mathfrak{m}}(h^s) > sr = \frac{ksm}{12}.$$

By Theorem 11.1.2, we have $\mathrm{ord}_{\mathfrak{m}}(h^s) = \infty$, so $\mathrm{ord}_{\mathfrak{m}}(h) = \infty$. It follows that $f \equiv g \pmod{\mathfrak{m}}$. □

11.1.3 Congruence for Newforms

Sturm's paper [Stu87] also applies some results of Asai on q-expansions at various cusps to obtain a more refined result for newforms.

Theorem 11.1.5 (Sturm). *Let N be a square-free positive integer, and suppose f and g are two newforms in $S_k(\Gamma_1(N), \varepsilon, \mathcal{O})$, where $\mathcal{O}$ is the ring of integers of a number field, and suppose that $\mathfrak{m}$ is a maximal ideal of $\mathcal{O}$. Let I be an arbitrary subset of the prime divisors of N. If $a_p(f) = a_p(g)$ for all $p \in I$, and*

$$a_p(f) \equiv a_p(g) \pmod{\mathfrak{m}}$$

for all primes

$$p \leq \frac{k \cdot [\mathrm{SL}_2(\mathbb{Z}) : \Gamma_0(N)]}{12 \cdot 2^{\#I}},$$

then $f \equiv g \pmod{\mathfrak{m}}$.

[BS02] also contains a result about congruences between newforms, which does not require that the level be square free. Recall (see Definition 4.5.5) that the conductor of a Dirichlet character ε is the largest divisor c of N such that ε factors through $(\mathbb{Z}/c\mathbb{Z})^\times$.

Theorem 11.1.6. *Let $N > 4$ be any integer, and suppose f and g are two normalized eigenforms in $S_k(\Gamma_1(N), \varepsilon, \mathcal{O})$, where $\mathcal{O}$ is the ring of integers of a number field, and suppose that $\mathfrak{m}$ is a maximal ideal of $\mathcal{O}$. Let I be the set of prime divisors of N that do not divide $\frac{N}{\mathrm{cond}(\varepsilon)}$. If*

$$a_p(f) \equiv a_p(g) \pmod{\mathfrak{m}}$$

for all primes $p \in I$ and for all primes

$$p \leq \frac{k \cdot [\mathrm{SL}_2(\mathbb{Z}) : \Gamma_0(N)]}{12 \cdot 2^{\#I}},$$

then $f \equiv g \pmod{\mathfrak{m}}$.

For the proof, see Lemma 1.4 and Corollary 1.7 in [BS02, §1.3].

11.2 Generating the Hecke Algebra as a $\mathbb{Z}$-module

The following theorem appeared in [LS02, Appendix], except that we give a better bound here.

Theorem 11.2.1. *Suppose Γ is a congruence subgroup that contains $\Gamma_1(N)$ and let*

$$r = \frac{km}{12} - \frac{m-1}{N}, \tag{11.2.1}$$

where $m = [\mathrm{SL}_2(\mathbb{Z}) : \Gamma]$. Then the Hecke algebra $\mathbb{T} = \mathbb{Z}[\ldots, T_n, \ldots] \subset \mathrm{End}(S_k(\Gamma))$ is generated as a $\mathbb{Z}$-module by the Hecke operators T_n for $n \leq r$.

Proof. For any ring R, let $S_k(N, R) = S_k(N, \mathbb{Z}) \otimes R$, where $S_k(N, \mathbb{Z}) \subset \mathbb{Z}[[q]]$ is the submodule of cusp forms with integer Fourier expansion at the cusp ∞, and let $\mathbb{T}_R = \mathbb{T} \otimes_{\mathbb{Z}} R$. For any ring R, there is a perfect pairing

$$S_k(N, R) \otimes_R \mathbb{T}_R \to R$$

given by $\langle f, T \rangle \mapsto a_1(T(f))$ (this is true for $R = \mathbb{Z}$, hence for any R).

Let M be the submodule of $\mathbb{T}$ generated by $T_1, T_2, \ldots, T_r$, where r is the largest integer $\leq \frac{kN}{12} \cdot [\mathrm{SL}_2(\mathbb{Z}) : \Gamma]$. Consider the exact sequence of additive abelian groups

$$0 \to M \xrightarrow{i} \mathbb{T} \to \mathbb{T}/M \to 0.$$

Let p be a prime and use that tensor product is right exact to obtain an exact sequence

$$M \otimes \mathbb{F}_p \xrightarrow{\bar{i}} \mathbb{T} \otimes \mathbb{F}_p \to (\mathbb{T}/M) \otimes \mathbb{F}_p \to 0.$$

Suppose that $f \in S_k(N, \mathbb{F}_p)$ pairs to 0 with each of $T_1, \ldots, T_r$. Then

$$a_m(f) = a_1(T_m f) = \langle f, T_m \rangle = 0$$

in $\mathbb{F}_p$ for each $m \leq r$. By Theorem 11.1.2, it follows that $f = 0$. Thus the pairing restricted to the image of $M \otimes \mathbb{F}_p$ in $\mathbb{T}_{\mathbb{F}_p}$ is nondegenerate, so because (11.2.1) is perfect, it follows that

$$\dim_{\mathbb{F}_p} \bar{i}(M \otimes \mathbb{F}_p) = \dim_{\mathbb{F}_p} S_k(N, \mathbb{F}_p).$$

Thus $(\mathbb{T}/M) \otimes \mathbb{F}_p = 0$. Repeating the argument for all primes p shows that $\mathbb{T}/M = 0$, as claimed. $\square$

Remark 11.2.2. In general, the conclusion of Theorem 11.2.1 is not true if one considers only T_n where n runs over the *primes* less than the bound. Consider, for example, $S_2(11)$, where the bound is 1 and there are no primes ≤ 1. However, the Hecke algebra is generated as an *algebra* by operators T_p with $p \leq r$.

Chapter 12

Miscellaneous

12.1 Computing Widths of Cusps

Let Γ be a congruence subgroup of level N. Suppose $\alpha \in C(\Gamma)$ is a cusp, and choose $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\gamma(\infty) = \alpha$. The minimal h such that $\left(\begin{smallmatrix} 1 & h \\ 0 & 1 \end{smallmatrix}\right) \in \gamma^{-1}\Gamma\gamma$ is called the *width* of the cusp α for the group Γ. In this section we discuss how to compute h.

Algorithm 12.1.1 (Width of Cusp). *Given a congruence subgroup Γ of level N and a cusp α for Γ, this algorithm computes the width h of α. We assume that Γ is given by congruence conditions, e.g., $\Gamma = \Gamma_0(N)$ or $\Gamma_1(N)$.*

1. [Find γ] Use the extended Euclidean algorithm to find $\gamma \in \mathrm{SL}_2(\mathbb{Z})$ such that $\gamma(\infty) = \alpha$, as follows. If $\alpha = \infty$ set $\gamma = 1$; otherwise, write $\alpha = a/b$, find c, d such that $ad - bc = 1$, and set $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$.
2. [compute Conjugate Matrix] Compute the following matrix in $M_2(\mathbb{Z}[x])$:
$$\delta(x) = \gamma \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \gamma^{-1}.$$
Note that the entries of $\delta(x)$ are constant or linear in x.
3. [Solve] The congruence conditions that define Γ give rise to four linear congruence conditions on x. Use techniques from elementary number theory (or enumeration) to find the smallest simultaneous positive solution h to these four equations.

Example 12.1.2. 1. Suppose $\alpha = 0$ and $\Gamma = \Gamma_0(N)$ or $\Gamma_1(N)$. Then $\gamma = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ has the property that $\gamma(\infty) = \alpha$. Next, the congruence condition is
$$\delta(x) = \gamma \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \gamma^{-1} = \begin{pmatrix} 1 & 0 \\ x & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & * \\ 0 & 1 \end{pmatrix} \pmod{N}.$$
Thus the smallest positive solution is $h = N$, so the width of 0 is N.

2. Suppose $N = pq$ where p, q are distinct primes, and let $\alpha = 1/p$. Then $\gamma = \left(\begin{smallmatrix} 1 & 0 \\ p & 1 \end{smallmatrix}\right)$ sends ∞ to α. The congruence condition for $\Gamma_0(pq)$ is

$$\delta(x) = \gamma \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \gamma^{-1} = \begin{pmatrix} 1 - px & x \\ -p^2x & px + 1 \end{pmatrix} \equiv \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \pmod{pq}.$$

   Since $p^2x \equiv 0 \pmod{pq}$, we see that $x = q$ is the smallest solution. Thus $1/p$ has width q, and symmetrically $1/q$ has width p.

Remark 12.1.3. For $\Gamma_0(N)$, once we enforce that the bottom left entry is $0 \pmod{N}$, and use that the determinant is 1, the coprimality from the other two congruences is automatic. So there is one congruence to solve in the $\Gamma_0(N)$ case. There are 2 congruences in the $\Gamma_1(N)$ case.

Bibliography

[Aga00] A. Agashe, *The Birch and Swinnerton-Dyer formula for modular abelian varieties of analytic rank* 0, Ph.D. thesis, University of California, Berkeley (2000).

[AL70] A. O. L. Atkin and J. Lehner, *Hecke operators on* $\Gamma_0(m)$, Math. Ann. **185** (1970), 134–160.

[Bas96] Jacques Basmaji, *Ein algorithmus zur berechnung von hecke-operatoren und anwendungen auf modulare kurven*, `http://modular.fas.harvard.edu/scans/papers/basmaji/` (1996).

[BCDT01] C. Breuil, B. Conrad, F. Diamond, and R. Taylor, *On the modularity of elliptic curves over* **Q**: *wild 3-adic exercises*, J. Amer. Math. Soc. **14** (2001), no. 4, 843–939 (electronic). MR 2002d:11058

[BCP97] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3–4, 235–265, Computational algebra and number theory (London, 1993). MR 1 484 478

[Bir71] B. J. Birch, *Elliptic curves over* **Q**: *A progress report*, 1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969), Amer. Math. Soc., Providence, R.I., 1971, pp. 396–400.

[BS02] K. Buzzard and W. A. Stein, *A mod five approach to modularity of icosahedral Galois representations*, Pacific J. Math. **203** (2002), no. 2, 265–282. MR 2003c:11052

[Buz96] Kevin Buzzard, *On the eigenvalues of the Hecke operator* T_2, J. Number Theory **57** (1996), no. 1, 130–132. MR 96m:11033

[CL04] J. Cremona and M. P. Lingham, *Finding all elliptic curves with good reduction outside a given set of primes*, In Progress (2004).

[CO77] H. Cohen and J. Oesterlé, *Dimensions des espaces de formes modulaires*, 69–78. Lecture Notes in Math., Vol. 627. MR 57 #12396

[Coh93] H. Cohen, *A course in computational algebraic number theory*, Springer-Verlag, Berlin, 1993. MR 94i:11105

[Cre92] J. E. Cremona, *Modular symbols for $\Gamma_1(N)$ and elliptic curves with everywhere good reduction*, Math. Proc. Cambridge Philos. Soc. **111** (1992), no. 2, 199–218.

[Cre97a] ______, *Algorithms for modular elliptic curves*, second ed., Cambridge University Press, Cambridge, 1997, http://www.maths.nott.ac.uk/personal/jec/book/.

[Cre97b] ______, *Computing periods of cusp forms and modular elliptic curves*, Experiment. Math. **6** (1997), no. 2, 97–107.

[CWZ01] Janos A. Csirik, Joseph L. Wetherell, and Michael E. Zieve, *On the genera of $X_0(N)$*, See http://www.csirik.net/papers.html (2001).

[Dem04] Lassina Dembele, *Quaternionic modular symbols and computing Hilbert modular forms,* http://modular.fas.harvard.edu/mcs/archive/spring2004/dembele.html.

[DI95] F. Diamond and J. Im, *Modular forms and modular curves*, Seminar on Fermat's Last Theorem, Providence, RI, 1995, pp. 39–133.

[DP04] H. Darmon and R. Pollack, *The efficient calculation of Stark-Heegner points via overconvergent modular symbols.*

[FJ02] D. W. Farmer and K. James, *The irreducibility of some level 1 Hecke polynomials*, Math. Comp. **71** (2002), no. 239, 1263–1270 (electronic). MR 2003e:11046

[FM99] G. Frey and M. Müller, *Arithmetic of modular curves and applications*, Algorithmic algebra and number theory (Heidelberg, 1997), Springer, Berlin, 1999, pp. 11–48.

[FpS$^+$01] E. V. Flynn, F. Leprévost, E. F. Schaefer, W. A. Stein, M. Stoll, and J. L. Wetherell, *Empirical evidence for the Birch and Swinnerton-Dyer conjectures for modular Jacobians of genus 2 curves*, Math. Comp. **70** (2001), no. 236, 1675–1697 (electronic). MR 1 836 926

[GH81] M. J. Greenberg and J. R. Harper, *Algebraic topology*, Benjamin/Cummings Publishing Co. Inc. Advanced Book Program, Reading, Mass., 1981, A first course. MR 83b:55001

[Gor93] Daniel M. Gordon, *Discrete logarithms in* GF(p) *using the number field sieve*, SIAM J. Discrete Math. **6** (1993), no. 1, 124–138. MR 94d:11104

[Gor04] D. Gordon, *Discrete logarithm problem,* http://www.win.tue.nl/~henkvt/content.html.

[Hij74] H. Hijikata, *Explicit formula of the traces of Hecke operators for* $\Gamma_0(N)$, J. Math. Soc. Japan **26** (1974), no. 1, 56–82.

[Kna92] A. W. Knapp, *Elliptic curves*, Princeton University Press, Princeton, NJ, 1992.

[Knu] Donald E. Knuth, *The art of computer programming. Vol. 2*, third ed., Addison-Wesley Publishing Co., Reading, Mass., Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[Lan95] S. Lang, *Introduction to modular forms*, Springer-Verlag, Berlin, 1995, With appendixes by D. Zagier and W. Feit, Corrected reprint of the 1976 original.

[Lem01] Dominic Lemelin, *Mazur-tate type conjectures for elliptic curves defined over quadratic imaginary fields.*

[Li75] W-C. Li, *Newforms and functional equations*, Math. Ann. **212** (1975), 285–315.

[LS02] Joan-C. Lario and René Schoof, *Some computations with Hecke rings and deformation rings*, Experiment. Math. **11** (2002), no. 2, 303–311, With an appendix by Amod Agashe and William Stein. MR MR1959271 (2004b:11072)

[Man72] J. I. Manin, *Parabolic points and zeta functions of modular curves*, Izv. Akad. Nauk SSSR Ser. Mat. **36** (1972), 19–66. MR 47 #3396

[Mar01] Franois Martin, *Périodes de formes modulaires de poids 1*, `http://modular.fas.harvard.edu/scans/papers/diderot/` (2001).

[Maz73] B. Mazur, *Courbes elliptiques et symboles modulaires*, Séminaire Bourbaki, 24ème année (1971/1972), Exp. No. 414, Springer, Berlin, 1973, pp. 277–294. Lecture Notes in Math., Vol. 317. MR 55 #2930

[Mer94] L. Merel, *Universal Fourier expansions of modular forms*, On Artin's conjecture for odd 2-dimensional representations, Springer, 1994, pp. 59–94.

[Miy89] T. Miyake, *Modular forms*, Springer-Verlag, Berlin, 1989, Translated from the Japanese by Yoshitaka Maeda.

[MTT86] B. Mazur, J. Tate, and J. Teitelbaum, *On p-adic analogues of the conjectures of Birch and Swinnerton-Dyer*, Invent. Math. **84** (1986), no. 1, 1–48. MR MR830037 (87e:11076)

[Nec94] V. I. Nechaev, *On the complexity of a deterministic algorithm for a discrete logarithm*, Mat. Zametki **55** (1994), no. 2, 91–101, 189. MR 96a:11145

[RS01] K. A. Ribet and W. A. Stein, *Lectures on Serre's conjectures*, Arithmetic algebraic geometry (Park City, UT, 1999), IAS/Park City Math. Ser., vol. 9, Amer. Math. Soc., Providence, RI, 2001, pp. 143–232. MR 2002h:11047

[Sch95] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Théor. Nombres Bordeaux **7** (1995), no. 1, 219–254, Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993). MR MR1413578 (97i:11070)

[Ser73] J-P. Serre, *A Course in Arithmetic*, Springer-Verlag, New York, 1973, Translated from the French, Graduate Texts in Mathematics, No. 7.

[Shi59] G. Shimura, *Sur les intégrales attachées aux formes automorphes*, J. Math. Soc. Japan **11** (1959), 291–311.

[Shi94] ______, *Introduction to the arithmetic theory of automorphic functions*, Princeton University Press, Princeton, NJ, 1994, Reprint of the 1971 original, Kan Memorial Lectures, 1.

[Sho97] Victor Shoup, *Lower bounds for discrete logarithms and related problems*, Advances in cryptology—EUROCRYPT '97 (Konstanz), Lecture Notes in Comput. Sci., vol. 1233, Springer, Berlin, 1997, pp. 256–266. MR 98j:94023

[Sil92] J. H. Silverman, *The arithmetic of elliptic curves*, Springer-Verlag, New York, 1992, Corrected reprint of the 1986 original.

[SJ05] William Stein and David Joyner, *Sage: System for algebra and geometry experimentation*, Communications in Computer Algebra (SIGSAM Bulletin) (July 2005), `http://sage.sourceforge.net/`.

[Šok80] V. V. Šokurov, *Shimura integrals of cusp forms*, Izv. Akad. Nauk SSSR Ser. Mat. **44** (1980), no. 3, 670–718, 720. MR MR582162 (82b:10029)

[Ste99] W. A. Stein, `HECKE`*: The modular symbols calculator*, Software (available online) (1999).

[Ste00] ______, *Explicit approaches to modular abelian varieties*, Ph.D. thesis, University of California, Berkeley (2000).

[Ste03] William Stein, *Modular abelian varieties,* `http://modular.fas.harvard.edu/edu/fall2003/252/`.

[Ste04] W. A. Stein, *The Modular Forms Database*, `http://modular.fas.harvard.edu/Tables` (2004).

[Stu87] J. Sturm, *On the congruence of modular forms*, Number theory (New York, 1984–1985), Springer, Berlin, 1987, pp. 275–280.

[SV01] W. A. Stein and H. A. Verrill, *Cuspidal modular symbols are transportable*, LMS J. Comput. Math. **4** (2001), 170–181 (electronic). MR 1 901 355

[Tat75] J. Tate, *Algorithm for determining the type of a singular fiber in an elliptic pencil*, Modular functions of one variable, IV (Proc. Internat. Summer School, Univ. Antwerp, Antwerp, 1972), Springer, Berlin, 1975, pp. 33–52. Lecture Notes in Math., Vol. 476. MR 52 #13850

[TW95] R. Taylor and A. J. Wiles, *Ring-theoretic properties of certain Hecke algebras*, Ann. of Math. (2) **141** (1995), no. 3, 553–572.

[Wil95] A. J. Wiles, *Modular elliptic curves and Fermat's last theorem*, Ann. of Math. (2) **141** (1995), no. 3, 443–551.