# AWS_3_Modular_Symbols

March 5, 2026

## 1 Modular symbols

### 1.1 Question 1: Hecke operators

In this question, we will recreate the Hecke operator functionality on modular symbols in Magma. We will compute the Hecke operators on the space $\mathcal{M}_2(\Gamma_0(23), \mathbb{C})$. Start by defining the space of modular symbols:

[0]:
```
M := ModularSymbols(23);
```

In weight 2, there are two cusp forms of level $\Gamma_0(23)$ and one Eisenstein series. Confirm that the dimension of $M$ is correct.

[0]:
```
Dimension(M);
```

Next, pick a small prime $p$. Recall, the Hecke operator at $p$ on a modular symbol $\{\alpha, \beta\}$ is given by

$$T_p\{\alpha, \beta\} = \sum_{m \in \Theta_p} m \cdot \{\alpha, \beta\},$$

where $\Theta_p$ is an explicit set of matrices that can be found in Example 2.6.5. Write a function `ThetaPMatrices` that returns a sequence containing these matrices for a given prime $p$;

[0]:
```
ThetaPMatrices := function(p)
    // your code here...
    return p;
end function;
```

Now we will create a function `ActOnModularSymbol` to act with a matrix on a modular symbol. The action is given by

$$m \cdot \{\alpha, \beta\} = \{m \cdot \alpha, m \cdot \beta\}.$$

This function should extract the entries of a symbol, act on them, then package this back up into the resulting modular symbol. The function `ModularSymbolRepresentation` gives us the entries $\alpha, \beta$ of a modular symbol, and its coefficient. The result of `ModularSymbolRepresentation` can be turned back into a modular symbol with coersion; for example,

```
M ! < 1, [ [1, 0], [0, 1] ] >;
```

will return the modular symbol $\{\infty, 0\}$.

[0]:
```
ActOnModularSymbol := function(m,symb)
    symb_rep := ModularSymbolRepresentation(symb);
```

```
    // your code here...
    return symb;
end function;
```

Finally, we will write a function `ModularSymbolHeckeOperator`, which takes a space `M` of modular symbols and a prime number `p`, and return a matrix giving the action of $T_p$ on a basis of `M`. The code should do the following: 1) Extract a basis of `M`; 2) Get the list of matrices in $\Theta_p$; 3) Compute the action of $T_p$ on each element of the basis, using `ActOnModularSymbol`; 4) Use `Eltseq` to express the resulting modular symbol in terms of the basis; 5) Package these vectors up into a matrix.

Compare the output of your function to Magma's in-built function, `HeckeOperator`.

[0]:
```
```

**Extension:** Use the command `CommonEigenspaces` to find systems of eigenvalues of the Hecker operators up to some bound, say $p \leq 50$, and the spaces of eigenvectors that realise them. Define the cuspidal subspace `C` of `M` with `CuspidalSubspace` and intersect with the above (using `M meet C`) to get the cuspidal eigenforms. Can you find them in the LMFDB?

[0]:
```
```

### 1.2 Question 2: Reduction of modular symbols

An important part of the modular symbols algorithm is being able to recognise a given modular symbol as a linear combination of symbols in your chosen basis. This is used, for example, in computing the Hecke action. In the previous example, Magma uses the basis $\{x, 0\}$, with $x \in \{-\frac{1}{19}, -\frac{1}{17}, -\frac{1}{15}, -\frac{1}{11}, \infty\}$. But how can we tell what, e.g. $\{\frac{1}{3}, \frac{2}{5}\}$ is in terms of this basis?

[0]:
```
M!< 1, [[1,3],[2,5]]>;
```

We will spend this question working through the reduction algorithm to recreate Magma's result, following Section 3.5 of the notes. First, recall that we can split the modular symbol $\{\frac{1}{3}, \frac{2}{5}\}$ into two symbols, as $\{\frac{1}{3}, 0\} - \{\frac{2}{5}, 0\}$. So it suffices to treat symbols of the form $\{x, 0\}$ for rational $x$. Write a function `CosetReps` that takes an integer `N` and returns the coset representatives of $\Gamma_0(N)$. These can be obtained from the **projective line** over $\mathbb{Z}/N\mathbb{Z}$ as follows: an element $(c : d)$ corresponds to a matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

in $\mathrm{SL}_2(\mathbb{Z})$. We will also need to be able to identify the coset representative associated to a given matrix $M \in \mathrm{SL}_2(\mathbb{Z})$. Recall that $M$ is in the coset $\Gamma_0(N)x$ if $Mx^{-1} \in \Gamma_0(N)$.

[0]:
```
CosetReps := function(N)
    ZZ := Integers();
    PL, r := ProjectiveLine(quo<ZZ | N*ZZ>);
    // your code here...
    return N;
end function;

IdentifyCosetRep := function(reps,M)
```

```
    // your code here...
    return reps;
end function;
```

Now we only need to decompose the symbols $\{\frac{1}{3}, 0\}$ and $\{\frac{2}{5}, 0\}$ into sums of elements of the form $\gamma_i\{\infty, 0\}$, using Magma's `ContinuedFraction` and (3.5.2) in the notes. Verify that your expression matches `M!< 1, [[1,3],[2,5]]>;`

[0]:

### 1.3 Question 3: Coset representatives of $\Gamma_0(N)$

We will use Magma's group theory functionality to find the coset representatives of $\Gamma_0(11)$, and verify that they really do correspond to elements of the projective line. First, we need the group $\mathrm{SL}_2(\mathbb{Z})$. We have the presentation

$$\mathrm{SL}_2(\mathbb{Z}) \simeq \langle S, ST \mid S^4 = (ST)^6 = 1, S^2 = (ST)^3 \rangle,$$

where $S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, $T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. Use FPGroup to create this group.

[0]:

Magma can tell us about the subgroups of $\mathrm{SL}_2(\mathbb{Z})$. What is the index of $\Gamma_0(11)$?

[0]:

Use `LowIndexSubgroups` to find all subgroups of $\mathrm{SL}_2(\mathbb{Z})$ whose index matches that of $\Gamma_0(11)$. You should have 88 groups. Can you identify which of these 88 groups is $\Gamma_0(11)$?

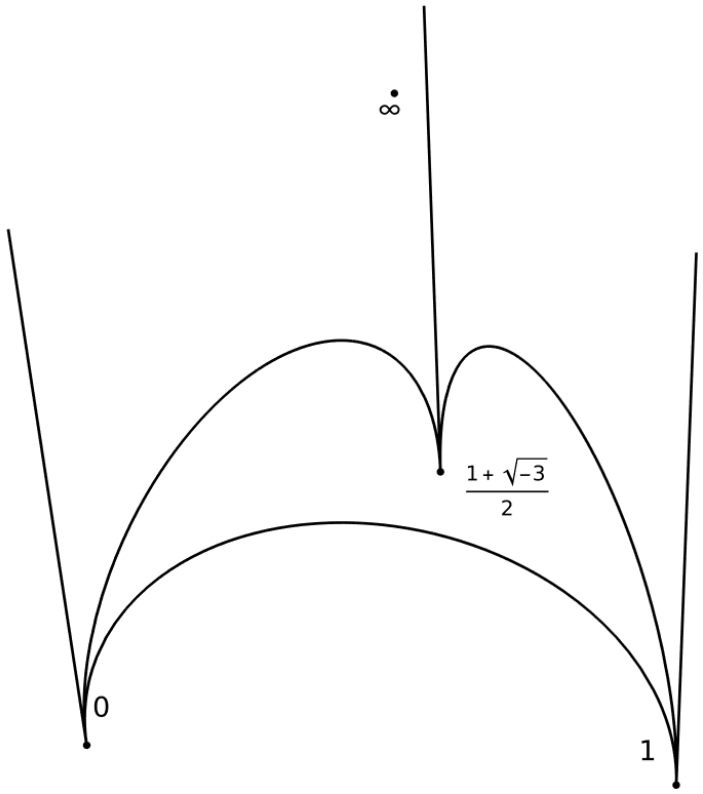
[0]:

Finally, look through the documentation to find a function that gives the coset representatives of $\Gamma_0(11)$ in $\mathrm{SL}_2(\mathbb{Z})$. Verify that these are in bijection with the set $\mathbb{P}^1(\mathbb{Z}/11\mathbb{Z})$.

[0]:

### 1.4 Question 4: Bianchi modular forms (hard extension problem)

Bianchi modular forms are modular forms for $\mathrm{PSL}_2(\mathbb{Z}_F)$, where $F$ is an imaginary quadratic field. Modular symbols algorithms have been developed for these groups, although the added complications of working over a number field mean the general algorithms are much more involved in this setting. Over $\mathbb{Q}(\sqrt{-3})$ the situation is relatively simple, since the field is Euclidean (and so the class number is 1, and we still have a simple continued fraction algorithm). Just as classical modular forms are functions on the upper half-plane $\mathbf{H}^2$, Bianchi modular forms are functions on the three-dimensional upper half-*space* $\mathbf{H}^3 \simeq \mathbb{C} \times \mathbb{R}_{>0}$. Below is a picture of the fundamental domain of $\mathrm{PSL}_2(\mathbb{Z}_F)$ on $\mathbf{H}^3$, a 3-dimensional region bounded by an ideal hyperbolic tetrahedron.




The paper *"Hyperbolic tessellations, modular symbols, and elliptic curves over complex quadratic fields"* (Compositio Mathematica, 1984) by John Cremona contains an explanation of how to compute modular symbols for this group, as well as tables of dimension data for some levels of small norm. Implement these methods and verify that your code produces the same dimension data as Cremona's table.

[0]: