# MODULAR SYMBOLS

## 0.1 Introduction

This chapter, which was written by William Stein (`was@math.harvard.edu`) with the help of feedback from Kevin Buzzard, describes how to compute with modular symbols using MAGMA. Modular symbols provide a presentation for certain homology groups, and as such they can be used to compute an eigenform basis for spaces of cusp forms $S_k(N, \varepsilon)$, where $k \geq 2$ is an integer and $\varepsilon$ is an *arbitrary* Dirichlet character. Their generality makes modular symbols a natural tool in applications ranging from verification of modularity of Galois representations to elliptic curve computations.

Our implementation of modular symbols algorithms in MAGMA was deeply influenced by [Cre92, Cre97, Mer94 ]. The algorithms for computing arithmetic invariants of modular abelian varieties are based on [Ste00]. Those unfamiliar with modular symbols might wish to consult [Ste02] and the references contained therein and peruse [FM99].

### 0.1.1 Modular Symbols

The *modular group* $\mathrm{SL}_2(\mathbf{Z})$ is the group of $2 \times 2$ integer matrices with determinant 1. For each positive integer $N$ let $\Gamma_0(N)$ denote the subgroup of $\mathrm{SL}_2(\mathbf{Z})$ of matrices that are upper triangular modulo $N$. As explained in the survey paper [DI95], there is an algebraic curve $X_0(N)$ over $\mathbf{Q}$ attached to $\Gamma_0(N)$. The Riemann surface attached to $X_0(N)$ is a compactified quotient of the upper half plane by the action of $\Gamma_0(N)$ via linear fractional transformations. Modular symbols provide an explicit computable presentation for certain "(co-)homology groups" attached to modular curves $X_0(N)$.

Let $\mathbf{P^1}(\mathbf{Q})$ denote the set $\mathbf{Q} \cup \{\infty\}$, and fix a field $F$. Let $\mathbf{M}$ denote the $F$-vector space generated by the formal symbols $\{a, b\}$, with $a, b \in \mathbf{P^1}(\mathbf{Q})$, modulo the relations $\{a, b\} + \{b, c\} + \{c, a\} = 0$ for all $a, b, c \in \mathbf{Q}$. (The symbol $\{a, b\}$ can be visualized as the homology class of a geodesic path from $a$ to $b$ in the upper half plane.) Fix a positive integer $k$. A weight-$k$ symbol is a formal product $X^i Y^{k-2-i}\{a, b\}$, where $X^i Y^{k-2-i} \in F[X, Y]$. Denote by $\mathbf{M_k}$ the formal $F$-vector space with basis the set of all weight-$k$ modular symbols (thus $\mathbf{M_k} \approx \mathbf{M} \otimes \mathrm{Sym}^{\mathbf{k-2}}(\mathbf{F} \times \mathbf{F})$). The group $\mathrm{GL}_2(\mathbf{Q})$ acts on the left on $\mathbf{M_k}$; the matrix $g = \begin{pmatrix} u & v \\ w & z \end{pmatrix}$ in $\mathrm{GL}_2(\mathbf{Q})$ acts by

$$g(X^i Y^{k-2-i}\{a, b\}) = (zX - vY)^i(-wX + uY)^{k-2-i} \left\{ \frac{ua + v}{wa + z}, \frac{ub + v}{wb + z} \right\}.$$

A *mod N Dirichlet character* $\varepsilon$ is a homomorphism

$$\varepsilon : (\mathbf{Z}/N\mathbf{Z})^* \to F^*.$$

The vector space $\mathbf{M_k}(\mathbf{N}, \varepsilon; \mathbf{F})$ of *modular symbols of weight k, level N and character* $\varepsilon$ *over* $F$ is the quotient of $\mathbf{M_k}$ by the subspace generated by all $x - \varepsilon(u)g(x)$, for $x$ in $\mathbf{M_k}$ and $g = \begin{pmatrix} u & v \\ w & z \end{pmatrix} \in \Gamma_0(N)$. We denote the equivalence class that defines a modular symbol by giving a representative element.

The space of modular symbols is a finite-dimensional vector space, and there is a natural finite presentation for it in terms of Manin symbols.

## 0.2 Basics

### 0.2.1 Verbose Output

The verbosity level for modular symbols computations can be set using the command `SetVerbose("ModularSymbols",n)`, where `n` is 0 (silent), 1 (verbose), or 2 (very verbose). (The verbose flag for modular symbols was called `ModularForms` in Magma version 2.7.)

### 0.2.2 Categories

Spaces of modular symbols belong to the category `ModSym`, and groups of Dirichlet characters form a category `GrpDrch`. The category `SetCsp` has exactly one object `Cusps()`, which is the set $\mathbf{P^1}(\mathbf{Q}) = \mathbf{Q} \cup \{\infty\}$ introduced above. The element $\infty$ of $\mathbf{P^1}(\mathbf{Q})$ is entered using the expression `Cusps()!Infinity()`.

**Example H0E1**

We compute a basis for the space of modular symbols of weight 2, level 11 and trivial character.

```
> M := ModularSymbols(11,2); M;
Full modular symbols space for Gamma_0(11) of weight 2 and dimension 3
over Rational Field
> Type(M);
ModSym
> Basis(M);
[
    {-1/7, 0},
    {-1/5, 0},
    {oo, 0}
]
> M!<1,[1/5,1]>;
{-1/5, 0}
> // the modular symbols {1/5,1} and {-1/5,0} are equal.
> Type(M!<1,[1/5,1]>);
ModSymElt
```

Using `SetVerbose`, we can see how the computation progresses.

```
> SetVerbose("ModularSymbols",2);
> M := ModularSymbols(11,2);
Computing space of modular symbols of level 11 and weight 2....
I.      Manin symbols list.
(0 s)
II.     2-term relations.
(0.019 s)
III.    3-term relations.
Computing quotient by 4 relations.
(0.009 s)
(total time to create space = 0.029 s)
> SetVerbose("ModularSymbols",0);
```

Modular symbols can be input using `Cusps()`.

```
> M := ModularSymbols(11,2);
> P := Cusps(); P;
Set of all cusps
> Type(P);
SetCsp
> oo := P!Infinity();
> M!<1,[oo,P!0]>;       // note that 0 must be coerced into P.
{oo, 0}
> M!<1,[1/5,1]> + M!<1,[oo,P!0]>;
{-1/5, 0} + {oo, 0}
```

Modular symbols are also defined over finite fields.

```
> M := ModularSymbols(11,2,GF(7)); M;
Full modular symbols space for Gamma_0(11) of weight 2 and dimension 3
over Finite field of size 7
> BaseField(M);
Finite field of size 7
> 7*M!<1,[1/5,1]>;
0
```

## 0.3 Creation Functions

### 0.3.1 Ambient Spaces

An ambient space of modular symbols is created by specifying a character, weight, and optional sign. The signature `ModularSymbols(eps, k, sign)` is the most general. (The level is the modulus of the character `eps`.)

**Warning:** Certain functions, such as `DualVectorSpace`, may fail when given as input a space of modular symbols over a field of positive characteristic, because the Hecke operators $T_p$, with $p$ prime to the level, need not be semisimple.

`ModularSymbols(N)`

The space of modular symbols of level $N$, weight 2, and trivial character over the rational numbers.

`ModularSymbols(N, k)`

The space of modular symbols of level $N$, weight $k$, and trivial character over the rational numbers.

`ModularSymbols(N, k, F)`

The space of modular symbols of level $N$, weight $k$, and trivial character over the field $F$.

`ModularSymbols(N, k, sign)`

The space of modular symbols of level $N$, weight $k$, trivial character, and given sign over the rational numbers.

`ModularSymbols(N, k, F, sign)`

The space of modular symbols of level $N$, weight $k$, trivial character, and given sign, over the field F.

`ModularSymbols(eps, k)`

The space of modular symbols of weight $k$ and character $\varepsilon$. Note that $\varepsilon$ determines the level and the base field, so they do not need to be specified.

`ModularSymbols(eps, k, sign)`

The space of modular symbols of weight $k$ and character $\varepsilon$. The level and base field are specified as part of $\varepsilon$. The third argument "sign" allows for working in certain quotients. The possible values are $-1$, 0, and $+1$, which correspond to the $-1$ quotient, full space, and $+1$ quotient, respectively. The $+1$ quotient of $M$ is $M/(* - 1)M$, where $*$ is `StarInvolution(M)`.

`DisownChildren(M)`

Ambient modular symbols spaces $M$ can create circular references, which cause the memory manager to never de-allocate $M$. Calling `DisownChildren` forces the

circular references created by $M$ to be deleted. In practice, if you are doing a computation involving many different spaces of modular symbols, it's probably best to use a scripting language such as Perl, Python, or even MAGMA, and start a separate MAGMA process for each computation.

**Example H0E2**

We create spaces of modular symbols in several different ways.

```
> M37 := ModularSymbols(37); M37;
Full modular symbols space for Gamma_0(37) of weight 2 and dimension 5
over Rational Field
> Basis(M37);
[
    {-1/29, 0},
    {-1/22, 0},
    {-1/12, 0},
    {-1/18, 0},
    {oo, 0}
]
```

As `M37` is a space of modular symbols, it is not incorrect that its dimension is different than that of the three-dimensional space of modular forms $M_2(\Gamma_0(37))$. We have

$$\dim \mathbf{M_2(\Gamma_0(37))} = \mathbf{2} \times (\text{dim cusp forms}) + \mathbf{1} \times (\text{dim Eisenstein series}) = \mathbf{5}.$$

```
> MF := ModularForms(Gamma0(37),2);
> 2*Dimension(CuspidalSubspace(MF)) + Dimension(EisensteinSubspace(MF));
5
```

Next we decompose `M37` with respect to the Hecke operators $T_2$, $T_3$, and $T_5$.

```
> D := Decomposition(M37,5); D;
[
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 1
     over Rational Field,
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
     over Rational Field
]
```

The first factor corresponds to the standard Eisenstein series, and the second corresponds to an elliptic curve:

```
> E := EllipticCurve(D[2]); E;
Elliptic Curve defined by y^2 + y = x^3 + x^2 - 23*x - 50 over
Rational Field
> Rank(E);
0
```

We now create the space $\mathbf{M_{12}(1)}$ of weight 12 modular symbols of level 1.

```
> M12 := ModularSymbols(1,12); M12;
Full modular symbols space for Gamma_0(1) of weight 12 and dimension 3
over Rational Field
> Basis(M12);
[
    X^10*{0, oo},
    X^8*Y^2*{0, oo},
    X^9*Y*{0, oo}
]

> DimensionCuspFormsGamma0(1,12);
1
> R<z>:=PowerSeriesRing(Rationals());
> Delta(z)+ O(z^7);
z - 24*z^2 + 252*z^3 - 1472*z^4 + 4830*z^5 - 6048*z^6 + O(z^7)
```

As a module, cuspidal modular symbols equal cuspforms with multiplicity two.

```
> M12 := ModularSymbols(1,12);
> HeckeOperator(CuspidalSubspace(M12),2);
[-24   0]
[  0 -24]
>  qExpansionBasis(CuspidalSubspace(M12),7);
[
    q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 + O(q^7)
]
```

For efficiency purposes, since one is often interested only in $q$-expansion, it is possible to work in the quotient of the space of modular symbols by all relations $*x = x$ (or $*x = -x$), where $*$ is `StarInvolution(M)`. In either of these quotients (except possibly in characteristic $p > 0$) the cusp forms appear with multiplicity one instead of two.

```
> M12plus := ModularSymbols(1,12,+1);
> Basis(M12plus);
[
    X^10*{0, oo},
    X^8*Y^2*{0, oo}
]
> CuspidalSubspace(M12plus);
Modular symbols space for Gamma_0(1) of weight 12 and dimension 1 over
Rational Field
> qExpansionBasis(CuspidalSubspace(M12),7);
[
    q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 + O(q^7)
]
```

The following is an example of how to create Dirichlet characters in MAGMA, and how to create a space of modular symbols with nontrivial character. For more details, see Section 0.3.4.

```
> G<a,b,c> := DirichletGroup(16*7,CyclotomicField(EulerPhi(16*7)));
```

```
> Order(a);
2
> Conductor(a);
4
> Order(b);
4
> Conductor(b);
16
> Order(c);
6
> Conductor(c);
7
> eps := a*b*c;
> M := ModularSymbols(eps,2); M;
Full modular symbols space of level 112, weight 2, character a*b*c,
and dimension 32 over Cyclotomic Field of order 48 and degree 16
> BaseField(M);
Cyclotomic Field of order 48 and degree 16
```

---

### 0.3.2 Labels

It is also possible to create many spaces of modular symbols for $\Gamma_0(N)$ by passing a "descriptive label" as an argument to `ModularSymbols`. The most specific label is a string of the form `[Level]k[Weight][IsogenyClass]`, where `[Level]` is the level, `[Weight]` is the weight, `[IsogenyClass]` is a letter code: `A`, `B`, ..., `Z`, `AA`, `BB`, ..., `ZZ`, `AAA`, ..., and `k` is a place holder to separate the level and the weight. If the label is `[Level][IsogenyClass]`, then the weight $k$ is assumed equal to 2. If the label is `[Level]k[Weight]` then the cuspidal subspace of the full ambient space of modular symbols of weight $k$ and level $N$ is returned. The following are valid labels: `11A`, `37B`, `3k12A`, `11k4`. The ordering used on isogenies classes is `lt`; see the documentation for `SortDecomposition`.

**Note:** There is currently no intrinsic that, given a space of modular symbols, returns its label.

**Warning:** For 146 of the levels between 56 and 450, our ordering of weight 2 rational newforms disagrees with the ordering used in [Cre97]. Fortunately, it is easy to create a space of modular symbols from one of Cremona's labels using the associated elliptic curve.

```
> E  := EllipticCurve(CremonaDatabase(),"56A");
> M1 := ModularSymbols(E);
```

Observe that Cremona's `"56A"` is different from ours.

```
> M2 := ModularSymbols("56A");
> M2 eq M1;
false
```

`ModularSymbols(s, sign)`

`ModularSymbols(s)`

The space of modular symbols described by a label.

**Example H0E3**

The cusp form $\Delta(q)$ is related to the space of modular symbols whose label is `"1k12A"`.

```
> Del := ModularSymbols("1k12A"); Del;
Modular symbols space for Gamma_0(1) of weight 12 and dimension 2 over
Rational Field
> qEigenform(Del,5);
q - 24*q^2 + 252*q^3 - 1472*q^4 + O(q^5)
```

Next, we create the space corresponding to the first newform on $\Gamma_0(11)$ of weight 4.

```
> M := ModularSymbols("11k4A"); M;
Modular symbols space for Gamma_0(11) of weight 4 and dimension 4 over
Rational Field
> AmbientSpace(M);
Full modular symbols space for Gamma_0(11) of weight 4 and dimension 6
over Rational Field
> qEigenform(M,5);
q + a*q^2 + (-4*a + 3)*q^3 + (2*a - 6)*q^4 + O(q^5)
> Parent($1);
Power series ring in q over Univariate Quotient Polynomial Algebra in
a over Rational Field with modulus a^2 - 2*a - 2
```

We next create the $+1$ quotient of the cuspidal subspace of weight-4 modular symbols of level 37.

```
> M := ModularSymbols("37k4",+1); M;
Modular symbols space for Gamma_0(37) of weight 4 and dimension 9 over
Rational Field
> AmbientSpace(M);
Full modular symbols space for Gamma_0(37) of weight 4 and dimension
11 over Rational Field
> Factorization(CharacteristicPolynomial(HeckeOperator(M,2)));
[
    <x^4 + 6*x^3 - x^2 - 16*x + 6, 1>,
    <x^5 - 4*x^4 - 21*x^3 + 74*x^2 + 102*x - 296, 1>
]
```

### 0.3.3 Creation of Elements

Suppose $M$ is a space of weight $k$ modular symbols over a field $F$. A modular symbol $P(X,Y)\{\alpha,\beta\}$ is input as `M!<P(X,Y),[alpha,beta]>`, where $P(X,Y) \in F[X,Y]$ is homogeneous of degree $k-2$, and $\alpha, \beta \in \mathbf{P^1}(\mathbf{Q})$. Here is an example:

**Example H0E4**

First create the space $M = \mathbf{M_4}(\mathbf{\Gamma_0}(\mathbf{3}); \mathbf{F_7})$.

```
> F7 := GF(7);
> M := ModularSymbols(3,4,F7);
> R<X,Y> := PolynomialRing(F7,2);
```

Now we input $(X^2 - 2XY)\{0, 1\}$.

```
> M!<X^2-2*X*Y,[Cusps()|0,1]>;
6*Y^2*{oo, 0}
```

Note that $(X^2 - 2XY)\{0, 1\} = 6Y^2\{\infty, 0\}$ in $M$.
When $k = 2$, simply enter `M!<1,[alpha,beta]>`.

```
> M := ModularSymbols(11,2);
> M!<1,[Cusps()|0,Infinity()]>;
-1*{oo, 0}
> M![<1,[Cusps()|0,Infinity()]>, <1,[Cusps()|0,1/11]>];
-2*{oo, 0}
```

---

Any space $M$ of modular symbols is finitely generated. One proof of this uses that every modular symbol is a linear combination of *Manin symbols*. Let $\mathbf{P^1}(\mathbf{Z}/\mathbf{NZ})$ be the set of pairs $(\overline{u}, \overline{v}) \in \mathbf{Z}/N\mathbf{Z} \times \mathbf{Z}N\mathbf{Z}$ such that $\mathrm{GCD}(u, v, N) = 1$, where $u$ and $v$ are lifts of $\overline{u}$ and $\overline{v}$ to $\mathbf{Z}$. A Manin symbols $\langle P(X, Y), (\overline{u}, \overline{v})\rangle$ is a pair consisting of a homogeneous polynomial $P(X, Y) \in F[X, Y]$ of degree $k - 2$ and an element $(\overline{u}, \overline{v}) \in \mathbf{P^1}(\mathbf{Z}/\mathbf{NZ})$. The modular symbol associated to $\langle P(X, Y), (\overline{u}, \overline{v})\rangle$ is constructed as follows. Choose lifts $u, v$ of $\overline{u}, \overline{v}$ such that $\mathrm{GCD}(u, v) = 1$. Then there is a matrix $g = \begin{pmatrix} w & z \\ u & v \end{pmatrix}$ in $\mathrm{SL}_2(\mathbf{Z})$ whose lower two entries are $u$ and $v$. The modular symbol is then $g(P(X, Y)\{0, \infty\})$. The intrinsic `ConvertFromManinSymbol` computes the modular symbol attached to a Manin symbol. Ever modular symbol can be written as a linear combination of Manin symbols using the intrinsic `ManinSymbol`.

**Example H0E5**

In this example, we convert between Manin and modular symbols representations of a few elements of a space of modular symbols of weight 4 over $\mathbf{F}_5$.

```
> F5 := GF(5);
> M := ModularSymbols(6,4,F5);
> R<X,Y> := PolynomialRing(F5,2);
> ConvertFromManinSymbol(M,<X^2+Y^2,[1,4]>);
(3*X^2 + 3*X*Y + 2*Y^2)*{-1/2, 0} + (X^2 + 4*X*Y + 4*Y^2)*{-1/3, 0} +
(X^2 + X*Y + 4*Y^2)*{1/3, 1/2}
> ManinSymbol(M.1-3*M.2);
[
<X^2, (0 1)>,
<2*X^2, (1 2)>
```

```
]
```

Thus the element `M.1-3*M.2` of $M$ corresponds to the sum of Manin symbols $\langle X^2, (0,1)\rangle + 2\langle X^2, (1,2)\rangle$.

---

`M ! x`

The coercion of $x$ into $M$. Here $x$ can be either a modular symbol that lies in a subspace of $M$, a 2-tuple that describes a modular symbol, a sequence of such 2-tuples, or anything that can be coerced into `VectorSpace(M)`. If $x$ is a valid sequence of such 2-tuples, then `M!x` is the sum of the coercions into $M$ of the elements of the sequence $x$.

`ConvertFromManinSymbol(M, x)`

The modular symbol associated to the 2-tuple $x = \langle P(X,Y), [u,v]\rangle$, where $P(X,Y) \in F[X,Y]$ is homogeneous of degree $k-1$, $F$ is the base field of $M$, and $[u,v]$ is a sequence of 2 integers that defines an element of $\mathbf{P^1}(\mathbf{Z}/N\mathbf{Z})$, where $N$ is the level of $M$.

`ManinSymbol(x)`

An expression for $x$ in terms of Manin symbols, which are represented as 2-tuples $\langle P(X,Y), [u,v]\rangle$.

**Example H0E6**

```
> M := ModularSymbols(14,2); M;
Full modular symbols space for Gamma_0(14) of weight 2 and dimension 5
over Rational Field
> Basis(M);
[
    {oo, 0},
    {-1/8, 0},
    {-1/10, 0},
    {-1/12, 0},
    {-1/2, -3/7}
]
> M!<1,[1,0]>;
0
> M!<1,[0,1/11]>;
{-1/10, 0} + -1*{-1/12, 0}
> M![<1,[0,1/2]>, <-1,[0,1/7]>];              // sequences are added
{-1/8, 0} + -1*{-1/12, 0} + -1*{-1/2, -3/7}
> M!<1,[0,1/2]> - M!<1,[0,1/7]>;
{-1/8, 0} + -1*{-1/12, 0} + -1*{-1/2, -3/7}
> M!<1,[Cusps()|Infinity(),0]>;               // Infinity() is in Cusps().
{oo, 0}
```

We can also coerce sequences into the underlying vector space of M.

```
> VectorSpace(M);
Full Vector space of degree 5 over Rational Field
Mapping from:  Full Vector space of degree 5 over Rational Field to
ModSym:  M given by a rule [no inverse]
Mapping from:  ModSym:  M to Full Vector space of degree 5 over Rational
Field given by a rule [no inverse]
> Eltseq(M.3);
[ 0, 0, 1, 0, 0 ]
> M![ 0, 0, 1, 0, 0 ];
{-1/10, 0}
> M.3;
{-1/10, 0}
```

The "polynomial coefficients" of the modular symbols are homogeneous polynomials in 2 variables of degree $k-2$.

```
> M := ModularSymbols(1,12);
> Basis(M);
[
    X^10*{0, oo},
    X^8*Y^2*{0, oo},
    X^9*Y*{0, oo}
]
> R<X,Y> := PolynomialRing(Rationals(),2);
> M!<X^9*Y,[Cusps()|0,Infinity()]>;
X^9*Y*{0, oo}
> M!<X^7*Y^3,[Cusps()|0,Infinity()]>;
-25/48*X^9*Y*{0, oo}
> Eltseq(M!<X*Y^9,[1/3,1/2]>);
[ -19171, -58050, -30970 ]
> M![1,2,3];
X^10*{0, oo} + 2*X^8*Y^2*{0, oo} + 3*X^9*Y*{0, oo}
> ManinSymbol(M![1,2,3]);
[
    <X^10, (0 1)>,
    <2*X^8*Y^2, (0 1)>,
    <3*X^9*Y, (0 1)>
]
```

---

### 0.3.4 Dirichlet Characters

Let $R$ be a ring. Then a *Dirichlet character over $R$ of modulus $N$* is a homomorphism

$$\varepsilon : (\mathbf{Z}/N\mathbf{Z})^* \to R^*,$$

where $R^*$ is the group of invertible elements of $R$. We extend $\varepsilon$ to a set theoretic map on the whole of $\mathbf{Z}$ by defining $\varepsilon(x) = 0$ if $\gcd(x, N) \neq 1$. The *conductor* of $\varepsilon$ is the smallest

positive integer $M$ such that the homomorphism $(\mathbf{Z}/N\mathbf{Z})^* \to R^*$ factors through $(\mathbf{Z}/M\mathbf{Z})^*$ via the natural map $(\mathbf{Z}/N\mathbf{Z})^* \to (\mathbf{Z}/M\mathbf{Z})^*$.

The following functions support computations involving Dirichlet characters.

`DirichletGroup(N)`

The group of Dirichlet characters modulo $N$ with image in `RationalField()`. This is a group of exponent at most 2.

`DirichletGroup(N,R)`

The group of Dirichlet characters modulo $N$ with image in the ring $R$.

`DirichletGroup(N,R,z,r)`

The group of Dirichlet characters mod $N$ with image in the order-$r$ cyclic subgroup of the ring $R$ generated by the root of unity $z$. Here $z$ must be an element of $R$ of exact order $r$. The reason $r$ must be input is that, for certain rings, it is not possible to determine the order of a general element.

`Elements(G)`

The Dirichlet characters in $G$.

`KroneckerCharacter(D)`

The Kronecker character $n \mapsto (D/n)$, where $D$ is a fundamental discriminant or 1. Thus `Evaluate(KroneckerCharacter(D),n)` equals `KroneckerSymbol(D,n)`.

`KroneckerCharacter(D, R)`

The Kronecker character $n \mapsto (D/n)$ over the ring $R$.

`Random(G)`

A random element of $G$.

`Ngens(G)`

The number of generators of $G$.

`AssignNames(~G, S)`

Assign names to the generators of G.

`Evaluate(x,n)`

Evaluate $x$ at the integer $n$.

`IsEven(x)`

True if and only if `Evaluate(x,-1)` is equal to 1. Note that in characteristic 0, the space of modular forms of weight $k$ and character $x$ is zero if $x$ is even and $k$ is odd.

`IsOdd(x)`

True if and only if `Evaluate(x,-1)` is equal to $-1$. Note that in characteristic 0, the space of modular forms of weight $k$ and character $x$ is zero if $x$ is odd and $k$ is even.

`IsTrivial(x)`

True if and only if $x$ has order 1.

`BaseExtend(G, R)`

Base extension of $G$ to $R$.

`BaseExtend(G, R, z)`

Base extension of $G$ to $R$ that identifies the distinguished root of unity of the base ring of $G$ with $z$.

`x * y`

The product of $x$ and $y$. This is a Dirichlet character of modulus equal to the least common multiple of the moduli of $x$ and $y$. The base rings and chosen roots of unity of the parents of $x$ and $y$ are equal.

`x ^ n`

The Dirichlet character $x$ raised to the power of $n$.

**Example H0E7**

We begin by constructing the group of characters $(\mathbf{Z}/5\mathbf{Z})^* \to \mathbf{Q}^*$.

```
> G<a> := DirichletGroup(5);  G;  // The default base field is Q.
Group of Dirichlet characters of modulus 5 over Rational Field
> #G;
2
> [Evaluate(a,n) :  n in [1..5]];
[ 1, -1, -1, 1, 0 ]
> Eltseq(a);
[ 2 ]
> a eq G![2];
true
> IsEven(a);
true
> IsOdd(a);
false
> IsTrivial(a);
false
```

Next we create a character by building it up "locally".

```
> G1<a4> := DirichletGroup(4);
> Conductor(a4);
4
> G2<a5> := DirichletGroup(25);
> Conductor(a5);
```

```
5
> eps := a4*a5;
> Modulus(eps);
100
> Conductor(eps);
20
> Evaluate(eps,7) eq Evaluate(a4,7)*Evaluate(a5,7);
true
```

Characters can be constructed over various fields.

```
> G<a> := DirichletGroup(7,GF(7));
> #G;
6
> Evaluate(a,2);
2
>
> G<a3,a5> := DirichletGroup(15,CyclotomicField(EulerPhi(15)));
> G;
Group of Dirichlet characters of modulus 15 over Cyclotomic Field of
order 8 and degree 4
> #G;
8
> Conductor(a3);
3
> Conductor(a5);
5
> Order(a5);
4
> Evaluate(a5,2);
zeta_8^2
```

If $D$ is a fundamental discriminant, then `KroneckerCharacter(D)` is the quadratic Dirichlet character corresponding to the quadratic field $\mathbf{Q}(\sqrt{D})$. The following code verifies that `KroneckerCharacter` and `KroneckerSymbol` agree in the case $D = 209$.

```
> chi := KroneckerCharacter(209);
> for n in [1..209] do
>    assert Evaluate(chi,n) eq KroneckerSymbol(209,n);
> end for;
```

If $E$ is an elliptic curve with newform $f_E$, then the twist $E_D$ corresponds to $f_E$ twisted by this character, as illustrated below.

```
> E := EllipticCurve(CremonaDatabase(),"11A");
> f := qEigenform(E,8); f;
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
> chi := KroneckerCharacter(-7);
> qEigenform(QuadraticTwist(E,-7),8);
q - 2*q^2 + q^3 + 2*q^4 - q^5 - 2*q^6 + O(q^8)
> R<q> := Parent(f);
```

```
> &+[Evaluate(chi,n)*Coefficient(f,n)*q^n :  n in [1..7]] + O(q^8);
q - 2*q^2 + q^3 + 2*q^4 - q^5 - 2*q^6 + O(q^8)
```

## 0.4 Bases

`Basis(M)`

Basis for $M$.

`IntegralBasis(M)`

First suppose that $M$ equals `AmbientSpace(M)`. Then this intrinsic returns a basis $x_1, \ldots, x_n$ for $M$ such that $\mathbf{Z}x_1 + \cdots + \mathbf{Z}x_n$ is the $\mathbf{Z}$-submodule of $M$ generated by all modular symbols $X^i \cdot Y^{k-2-i}\{\alpha, \beta\}$ with $i = 0, \ldots, k-2$ and $\alpha, \beta \in \mathbf{P^1}(\mathbf{Q})$. If $M$ is not `AmbientSpace(M)`, then this intrinsic returns a $\mathbf{Z}$-basis for $M \cap (\mathbf{Z}x_1 + \cdots + \mathbf{Z}x_n)$, where $x_1, \ldots, x_n$ is an integral basis for `AmbientSpace(M)`. The base field of $M$ must be $\mathbf{Q}$.

**Example H0E8**

```
> M := ModularSymbols(1,12);
> Basis(M);
[
    X^10*{0, oo},
    X^8*Y^2*{0, oo},
    X^9*Y*{0, oo}
]
> IntegralBasis(M);
[
    1/48*X^9*Y*{0, oo},
    1/14*X^8*Y^2*{0, oo},
    X^10*{0, oo}
]
```

`IntegralBasis(M)` is a basis for the $\mathbf{Z}$-module spanned by the following symbols:

```
> R<X,Y> := PolynomialRing(Rationals(),2);
> [M!<X^i*Y^(10-i),[Cusps()|0,Infinity()]> :  i in [0..10]];
[
    -X^10*{0, oo},
    X^9*Y*{0, oo},
    -X^8*Y^2*{0, oo},
    -25/48*X^9*Y*{0, oo},
    9/14*X^8*Y^2*{0, oo},
    5/12*X^9*Y*{0, oo},
    -9/14*X^8*Y^2*{0, oo},
    -25/48*X^9*Y*{0, oo},
    X^8*Y^2*{0, oo},
```

```
    X^9*Y*{0, oo},
    X^10*{0, oo}
]
```

We can also compute an integral basis of a subspace.

```
> C := CuspidalSubspace(M);
> IntegralBasis(C);
[
    1/48*X^9*Y*{0, oo},
    1/14*X^8*Y^2*{0, oo}
]
```

In Remark 3 on page 69 of [Mer94], Merel says "it would be interesting to find a basis in terms of Manin symbols" for the **Z**-module of Eisenstein symbols (see Section 0.8 for the definition of `EisensteinSubspace`). Here are the first few examples in the case of level 1:

```
> M := ModularSymbols(1,12);
> E := EisensteinSubspace(M);
> IntegralBasis(E);
[
    691*X^10*{0, oo} + 1620*X^8*Y^2*{0, oo}
]
> ManinSymbol(IntegralBasis(E)[1]);
[
    <691*X^10, (0 1)>,
    <1620*X^8*Y^2, (0 1)>
]
```

To more easily compute several examples, we define a function:

```
> function EisZ(k)
>    E := EisensteinSubspace(ModularSymbols(1,k));
>    B := IntegralBasis(E);
>    return [ManinSymbol(z) :  z in B];
> end function;
> EisZ(12);
[
    [
        <691*X^10, (0 1)>,
        <1620*X^8*Y^2, (0 1)>
    ]
]
> EisZ(16);
[
    [
        <16380*X^12*Y^2, (0 1)>,
        <3617*X^14, (0 1)>
    ]
]
> EisZ(18);
```

```
[
    [
        <43867*X^16, (0 1)>,
        <270000*X^14*Y^2, (0 1)>
    ]
]
> EisZ(20);
[
    [
        <174611*X^18, (0 1)>,
        <1349460*X^16*Y^2, (0 1)>
    ]
]
> EisZ(22);
[
    [
        <748125*X^18*Y^2, (0 1)>,
        <77683*X^20, (0 1)>
    ]
]
```

Send me an email if you determine the basis in general. In each example above the coefficient of $X^{k-2}$ is, up to sign, `Numerator(Bernoulli(k)/k)`.

---

## 0.5 Associated Vector Space

The functions `VectorSpace`, `DualVectorSpace`, and `Lattice` return the underlying vector space, dual vector space, and lattice associated to a space of modular symbols. A space of modular symbols is represented internally as a subspace of a vector space, and a subspace of the linear dual of the vector space. To carry along the subspace of the linear dual is useful in many computations; one example is efficient computation of Hecke operators. When the base field is $\mathbf{Q}$, the lattice comes from the natural integral structure on modular symbols.

`VectorSpace(M)`

The vector space $V$ underlying $M$, the map $V \to M$, and the map $M \to V$.

`DualVectorSpace(M)`

The subspace of the linear dual of `VectorSpace(AmbientSpace(M))` that is isomorphic to $M$ as a module over the Hecke algebra.

`Lattice(M)`

The lattice generated by the integral modular symbols in the vector space representation of $M$. This is the lattice generated by all modular symbols $X^iY^{k-2-i}\{a, b\}$. The base field of $M$ must be `RationalField()`.

**Example H0E9**

```
> M := ModularSymbols(DirichletGroup(11).1,3); M;
Full modular symbols space of level 11, weight 3, character $.1, and
dimension 4 over Rational Field
> VectorSpace(M);
Full Vector space of degree 4 over Rational Field
Mapping from:  Full Vector space of degree 4 over Rational Field to
ModSym:  M given by a rule [no inverse]
Mapping from:  ModSym:  M to Full Vector space of degree 4 over Rational
Field given by a rule [no inverse]
> Basis(VectorSpace(CuspidalSubspace(M)));
[
    ( 0  1  0 -1),
    ( 0  0  1 -1)
]
> Basis(VectorSpace(EisensteinSubspace(M)));
[
    (   1    0 -2/3 -1/3),
    ( 0  1 -5 -2)
]
> Lattice(CuspidalSubspace(M));
Lattice of rank 2 and degree 4
Basis:
( 0  1 -1  0)
( 0  1  1 -2)
Basis Denominator:  2
Mapping from:  Lattice of rank 2 and degree 4 to Modular symbols space
of level 11, weight 3, character $.1, and dimension 2 over Rational
Field given by a rule [no inverse]
> Basis(Lattice(EisensteinSubspace(M)));
[
    (   0  1/2 -5/2   -1),
    (   3 -1/2  1/2    0)
]
```

## 0.6 Degeneracy Maps

Consider an ambient space $M_1$ of modular symbols of level $N_1$, and suppose $M_2$ is an ambient space of modular symbols of level a multiple $N_2$ of $N_1$ whose weight equals the weight of $M_1$ and whose character is induced by the character of $M_1$. Then for each divisor $d$ of $N_2/N_1$ there are natural maps $\alpha_d : M_1 \to M_2$ and $\beta_d : M_2 \to M_1$ such that $\beta_d \circ \alpha_d$ is multiplication by $d^{k-2} \cdot [\Gamma_0(N_1) : \Gamma_0(N_2)]$, where $k$ is the common weight of $M_1$ and $M_2$. On cuspidal parts, the map $\beta_d$ is dual to the map $f(q) \to f(q^d)$ on modular forms. Use the function `DegeneracyMap` to compute the maps $\alpha_d$ and $\beta_d$.

Given a space $M$ of modular symbols and a positive integer $N$ that is a multiple of the level of $M$, the images of $M$ under the degeneracy maps generate a modular symbols space of level $N$. The constructor `ModularSymbols(M,N)` computes this space.

Let $M$ be a space of modular symbols of level $N$, and let $N'$ be a multiple of $N$. The subspace

$$\sum_{d|\frac{N'}{N}} \alpha_d(M) \subset \mathbf{M_k}(\mathbf{N}', \varepsilon)$$

is stable under the Hecke operators. Here is how to create this subspace using MAGMA:

```
> M := ModularSymbols(11,2); M;
Full modular symbols space for Gamma_0(11) of weight 2 and dimension 3
over Rational Field
> M33 := ModularSymbols(M,33); M33;
Modular symbols space for Gamma_0(33) of weight 2 and dimension 6 over
Rational Field
```

`DegeneracyMap(M1, M2, d)`

The degeneracy map $M_1 \to M_2$ associated to $d$. Let $N_i$ be the level of $M_i$ for $i = 1, 2$. Suppose that $d$ is a divisor of either the numerator or denominator of the rational number $N_1/N_2$, written in reduced form. If $N_1 \mid N_2$, then this intrinsic returns $\alpha_d : M_1 \to M_2$, or if $N_2 \mid N_1$, then this intrinsic returns $\beta_d : M_1 \to M_2$. It is an error if neither divisibility holds.

`DegeneracyMatrix(M1, M2, d)`

The matrix of `DegeneracyMap(M1,M2,d)` with respect to `Basis(M1)` and `Basis(M2)`. Both `IsAmbient(M1)` and `IsAmbient(M2)` must be true.

`ModularSymbols(M, N')`

The modular symbols space of level $N'$ associated to $M$. Let $N$ be the level of $M$. If $N \mid N'$, then this intrinsic returns the modular symbols space

$$\sum_{d|\frac{N'}{N}} \alpha_d(M).$$

If $N' \mid N$, then this intrinsic returns the modular symbols space

$$\sum_{d \mid \frac{N}{N'}} \beta_d(M).$$

In this latter case, if `Conductor(DirichletCharacter(M))` does not divide $N'$, then the 0 space is returned.

`M1 !! M2`

The modular symbols subspace of $M_1$ associated to $M_2$. Let $N_1$ be the level of $M_1$. If `ModularSymbols(M2,N1)` is defined, let $M_3$ be this modular symbols space, otherwise terminate with an error. If $M_3$ is contained in $M_1$, return $M_3$, otherwise terminate with an error.

**Example H0E10**

We compute degeneracy maps $\alpha_2$ and $\beta_2$.

```
> M15 := ModularSymbols(15);
> M30 := ModularSymbols(30);
> alp_2 := DegeneracyMap(M15,M30,2);
> alp_2(M15.1);
2*{oo, 0} + -1*{-1/28, 0} + -1*{-1/2, -7/15}
> beta_2 := DegeneracyMap(M30,M15,2);
> beta_2(alp_2(M15.1));
3*{oo, 0}
> M15.1;
{oo, 0}
```

We can consider the space generated by the image of a space of modular symbols of level 11 in spaces of higher level.

```
> X11 := ModularSymbols("11k2A");
> qEigenform(X11,6);
q - 2*q^2 - q^3 + 2*q^4 + q^5 + O(q^6)
> ModularSymbols(X11,33);
Modular symbols space for Gamma_0(33) of weight 2 and dimension 4 over
Rational Field
> X33 := ModularSymbols(X11,33);
> qExpansionBasis(X33,6);
[
    q - 2*q^2 + 2*q^4 + q^5 + O(q^6),
    q^3 + O(q^6)
]
> Factorization(CharacteristicPolynomial(HeckeOperator(X33,3)));
[
    <x^2 + x + 3, 2>
]
> ModularDegree(X33);
```

```
3
```

We can also construct the space generated by the images of `X11` at higher level using the `!!` operator.

```
> M44 := ModularSymbols(44,2);
> A := M44!!X11; A;
Modular symbols space for Gamma_0(44) of weight 2 and dimension 6 over
Rational Field
> X11!!A;       // back to the original space
Modular symbols space for Gamma_0(11) of weight 2 and dimension 2 over
Rational Field
```

---

## 0.7 Decomposition

The functions `Decomposition` and `NewformDecomposition` express a space of modular symbols as a direct sum of Hecke-stable subspaces.

In the intrinsics below, the `Proof` parameter affects the internal characteristic polynomial computations. If `Proof` is set to `false` and this causes a characteristic polynomial computation to fail, then the sum of the dimensions of the spaces returned by `Decomposition` will be less than the dimension of `M`. Thus setting `Proof` equal to `false` is usually safe.

`Decomposition(M, bound : parameters)`

`Proof` BoolElt *Default : true*

The decomposition of $M$ with respect to the Hecke operators $T_p$ with $p$ coprime to the level of $M$ and $p \leq$ `bound`. If `bound` is too small, the constituents of the decomposition are not guaranteed to be "irreducible", in the sense that they can not be decomposed further into kernels and images of Hecke operators $T_p$ with $p$ prime to the level of $M$. When `Decomposition` is called, the result is cached, so each successive call results in a possibly more refined decomposition.
Important Note: In some cases `NewformDecomposition` is significantly faster than `Decomposition`.

`NewformDecomposition(M : parameters)`

`Proof` BoolElt *Default : true*

Unsorted decomposition of $M$ into factors corresponding to the Galois conjugacy classes of newforms of level some divisor of the level of $M$. We require that `IsCuspidal(M)` is `true`.

`AssociatedNewSpace(M)`

The space of modular symbols corresponding to the Galois-conjugacy class of newforms associated to $M$. The level of the newforms is allowed to be a proper divisor of the level of $M$. The space $M$ must have been created using `NewformDecomposition`.

`SortDecomposition(D)`

Sort the sequence $D$ of spaces of modular symbols with respect to the `lt` comparison operator.

`IsIrreducible(M)`

True if and only if Decomposition(M) has cardinality 1.

`M1 lt M2`

The ordering determined as follows:

(1) This rule applies only if `NewformDecomposition` was used to construct both of $M_1$ and $M_2$: If `Level(AssociatedNewSpace(M1))` and `Level(AssociatedNewSpace(M2))` are not equal then the $M_i$ with larger associated level is first.

(2) The smaller dimension is first.

(3) The following applies when the weight is 2 and the character is trivial: Order by $W_q$ eigenvalues, starting with the *smallest* $p \mid N$, with the eigenvalue $+1$ being less than the eigenvalue $-1$.

(4) Order by abs(trace($a_p$)), with $p$ not dividing the level, and with positive trace being smaller in the event that the two absolute values are equal.

Rule (3) is included so that our ordering extends the one used in (most of!) [Cre97].

**Example H0E11**

First, we compute the decomposition of the space of modular symbols of weight 2 and level 37.

```
> M := ModularSymbols(37,2); M;
Full modular symbols space for Gamma_0(37) of weight 2 and dimension 5
over Rational Field
> D := Decomposition(M,2); D;
[
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 1
     over Rational Field,
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
     over Rational Field
]
> IsIrreducible(D[2]);
true
> C := CuspidalSubspace(M); C;
Modular symbols space for Gamma_0(37) of weight 2 and dimension 4 over
Rational Field
> N := NewformDecomposition(C); N;
[
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(37) of weight 2 and dimension 2
```

```
     over Rational Field
]
```

Next, we use `NewformDecomposition` to decompose a space having plentiful old subspaces.

```
> M := ModularSymbols(90,2); M;
Full modular symbols space for Gamma_0(90) of weight 2 and dimension
37 over Rational Field
> D := Decomposition(M,11); D;
[
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 11
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 4
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 4
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 8
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 6
     over Rational Field
]
> C := CuspidalSubspace(M); C;
Modular symbols space for Gamma_0(90) of weight 2 and dimension 22
over Rational Field
> N := NewformDecomposition(C); N;
[
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 4
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 4
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 8
     over Rational Field
]
```

The above decomposition uses all of the Hecke operator; it suggests that the decomposition `D` is not as fine as possible. Indeed, `D[7]` breaks up further:

```
> Decomposition(D[7],11);
[
    Modular symbols space for Gamma_0(90) of weight 2 and dimension 6
```

```
    over Rational Field
]
> Decomposition(D[7],19);
[
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 4
     over Rational Field,
     Modular symbols space for Gamma_0(90) of weight 2 and dimension 2
     over Rational Field
]
```

The function `AssociatedNewSpace` allows us to see where each of these subspace comes from. By definition they each arise by taking images under the degeneracy maps from a single Galois-conjugacy class of newforms of *some* level dividing 90.

```
> [Level(AssociatedNewSpace(A)) : A in N];
[ 90, 90, 90, 45, 30, 15 ]
> A := N[4];
> qEigenform(AssociatedNewSpace(A),7);
q + q^2 - q^4 - q^5 + O(q^7)
> qExpansionBasis(A,7);
[
    q - 2*q^4 - q^5 + O(q^7),
    q^2 + q^4 + O(q^7)
]
```

---

## 0.8 Subspaces

The following functions compute the cuspidal, Eisenstein, and new subspaces, along with the complement of a subspace.

`CuspidalSubspace(M)`

The cuspidal subspace of $M$. This is the kernel of `BoundaryMap(M)`.

`IsCuspidal(M)`

True if and only if M is contained in the cuspidal subspace of the ambient space.

`EisensteinSubspace(M)`

The Eisenstein subspace of $M$. This is the complement in $M$ of the cuspidal subspace of $M$.

`IsEisenstein(M)`

True if and only if M is contained in the Eisenstein subspace of the ambient space.

`NewSubspace(M)`

The new subspace of $M$. This is the intersection of `NewSubspace(M,p)` as $p$ varies over all prime divisors of the level of $M$. Note that $M$ is required to be cuspidal.

`IsNew(M)`

True if and only if $M$ is contained in the new cuspidal subspace of the ambient space.

`NewSubspace(M, p)`

The $p$-new subspace of $M$. This is the kernel of the degeneracy map from $M$ to the space of modular symbols of level equal to the level of $M$ divided by $p$ and character the restriction of the character of $M$. If the character of $M$ does not restrict, then `NewSubspace(M,p)` is equal to $M$. Note that $M$ is required to be cuspidal.

`Kernel(I, M)`

The kernel of $I$ on $M$. Let $T_p$ denote the $p$th Hecke operator (see Section 0.9). This is the subspace of $M$ obtained by intersecting the kernels of the operators $f_n(T_{p_n})$, where $I$ is a sequence $[\langle p_1, f_1(x)\rangle, ..., \langle p_n, f_n(x)\rangle]$ of pairs consisting of a prime number and a polynomial. Only primes $p_i$ which do not divide the level of $M$ are used.

`Complement(M)`

The space of modular symbols complementary to $M$ in the ambient space of $M$. Thus the ambient space of M is equal to the direct sum of $M$ and `Complement(M)`.

`BoundaryMap(M)`

A matrix that represents the boundary map from $M$ to the vector space whose basis consists of the weight $k$ cusps. (Note: At present there is no intrinsic that lists these cusps.)

**Example H0E12**

First we compute the cuspidal subspace of the space of modular symbols for $\Gamma_0(11)$.

```
> M := ModularSymbols(11,2); M;
Full modular symbols space for Gamma_0(11) of weight 2 and dimension 3
over Rational Field
> IsCuspidal(M);
false
> C := CuspidalSubspace(M); C;
Modular symbols space for Gamma_0(11) of weight 2 and dimension 2 over
Rational Field
> IsCuspidal(C);
true
```

Next we compute the Eisenstein subspace.

```
> IsEisenstein(C);
false
> E := EisensteinSubspace(M); E;
Modular symbols space for Gamma_0(11) of weight 2 and dimension 1 over
```

```
Rational Field
> IsEisenstein(E);
true
> E + C eq M;
true
```

The Eisenstein subspace is the complement of the cuspidal subspace, and conversely.

```
> E eq Complement(C);
true
> C eq Complement(E);
true
```

---

**Example H0E13**

```
> M := ModularSymbols("37B"); M;
Modular symbols space for Gamma_0(37) of weight 2 and dimension 2 over
Rational Field
> BoundaryMap(M);
[0 0]
[0 0]
> A := AmbientSpace(M);
> BoundaryMap(A);
[ 0  0]
[ 0  0]
[ 0  0]
[ 0  0]
[ 1 -1]
```

Observe that the Eisenstein subspace of $A$ is not in the kernel of the boundary map.

```
> Basis(VectorSpace(EisensteinSubspace(A)));
[
    (0 0 0 1 3)
]
```

---

## 0.9 Operators

Each space $\mathbf{M}$ of modular symbols comes equipped with a commuting family $T_1, T_2, T_3, \ldots$ of linear operators acting on it called the Hecke operators.

The Hecke operators are defined recursively, as follows. First, $T_1 = 1$. When $n = p$ is prime,

$$T_p(x) = \left[ \begin{pmatrix} p & 0 \\ 0 & 1 \end{pmatrix} + \sum_{r \bmod p} \begin{pmatrix} 1 & r \\ 0 & p \end{pmatrix} \right] x,$$

where the first matrix is omitted if $p$ divides the level $N$ of $M$. If $m$ and $n$ are coprime, then $T_{mn} = T_m T_n$. If $p$ is a prime, $r \geq 2$ is an integer, $\varepsilon$ is the Dirichlet character associated to $M$, and $k$ is the weight of $M$, then

$$T_{p^r} = T_p T_{p^{r-1}} - \varepsilon(p) p^{k-1} T_{p^{r-2}}.$$

**Example H0E14**

In MAGMA, Hecke operators are represented as $n \times n$-matrices, acting from the right, with respect to the basis `Basis(M)`. For example

```
> M := ModularSymbols(12);
> T2 := HeckeOperator(M,2);
> M.1;
{oo, 0}
> T2;
[ 2  0 -1  0  0]
[ 2  0 -1  0  0]
[ 0  0  1 -2 -2]
[ 0 -1  1 -1 -2]
[ 0  1 -1  1  2]
> M.1*T2;
2*{oo, 0} + -1*{-1/10, 0}
```

`HeckeOperator(M, n)`

Compute a matrix representing the $n$th Hecke operator $T_n$ with respect to `Basis(M)`.

`HeckePolynomial(M, n)`

Compute the characteristic polynomial of the Hecke operator $T_n$. When $n$ is prime, the Deligne bound on the sizes of Hecke eigenvalues is used, so `HeckePolynomial` is frequently much faster than `CharacteristicPolynomial(HeckeOperator(M,n))`.

`IntegralHeckeOperator(M, n)`

A matrix representing the $n$th Hecke operator with respect to `Basis(Lattice(M))`.

`DualHeckeOperator(M, n)`

Compute a matrix representing the Hecke operator $T_n$ on the dual vector space representation of $M$. This function is much more efficient than `HeckeOperator(M,n)` when the dimension of $M$ is small relative to the dimension of the `AmbientSpace(M)`. Note that `DualHeckeOperator(M,n)` is not guaranteed to equal the transpose of `HeckeOperator(M,n)` because `DualHeckeOperator(M,n)` is computed with respect to `Basis(DualVectorSpace(M))`.

`AtkinLehner(M, q)`

A matrix representing the $q$th Atkin-Lehner involution $W_q$ on $M$, when it is defined. The involution $W_q$ is defined when $M$ has trivial character and even weight. When possible, the Atkin-Lehner map is normalized so that it is an involution; such normalization may not be possible when $k > 2$ and the characteristic of the base field of $M$ divides $q$.

To each divisor $q$ of $N$ such that $\gcd(q, N/q) = 1$ there is an *Atkin-Lehner involution* $W_q$ on $M$, which is defined as follows. Using the Euclidean algorithm, choose integers $x, y, z, w$ such that $qxw - (N/q)yz = 1$; let $g = \begin{pmatrix} dx & y \\ Nz & qw \end{pmatrix}$ and define

$$W_q(x) = g(x)/q^{\frac{k-2}{2}}.$$

For example, when $q = N$ we have $g = \begin{pmatrix} 0 & -1 \\ N & 0 \end{pmatrix}$.

`DualAtkinLehner(M, q)`

The action of the Atkin-Lehner involution on the dual representation of $M$, when it is defined.

`StarInvolution(M)`

The conjugation involution $*$ on $M$ that sends the modular symbol $X^iY^j\{u, v\}$ to $(-1)^j X^iY^j\{-u, -v\}$.

`DualStarInvolution(M)`

The conjugation involution $*$ on the dual representation of $M$ (see the documentation for `StarInvolution`.)

`ThetaOperator(M1, M2)`

Multiplication by $X^pY - XY^p$, which is a possible analogue of the $\theta$-operator. (On mod $p$ modular forms, the $\theta$-operator is the map given by $f \mapsto q\frac{df}{dq}$.) Both $M_1$ and $M_2$ must be spaces of modular symbols over a field of positive characteristic $p$; they must have the same level and character, and the weight of $M_2$ must equal the weight of $M_1$ plus $p + 1$.

**Example H0E15**

```
> M := ModularSymbols(11,4,+1); M;
```

```
Full modular symbols space for Gamma_0(11) of weight 4 and dimension 4
over Rational Field
> HeckeOperator(M,2);
[   9    0  2/5 -2/5]
[   0    5  9/5 11/5]
[   0    5  7/5 13/5]
[   0    0 22/5 23/5]
```

The entries of $T_2$ are not guaranteed to be integers because `Basis(M)` is just a basis of a $\mathbf{Q}$-vector space. The entries will be integers if we compute $T_2$ with respect to an integral basis.

```
> IntegralHeckeOperator(M,2);
[ 0  2  0  0]
[ 1  2  0  0]
[-5  6  9  0]
[ 2  0  0  9]
```

The matrix for the Hecke operator on the dual of $M$ is the transpose of $T_2$. However, the chosen basis for the cuspidal subspace of the dual of $M$ need not satisfy any compatibility with `CuspidalSubspace(M)`.

```
> DualHeckeOperator(M,2);
[   9    0    0    0]
[   0    5    5    0]
[ 2/5  9/5  7/5 22/5]
[-2/5 11/5 13/5 23/5]
> S := CuspidalSubspace(M);
> HeckeOperator(S,2);
[    5 -13/5]
[    5    -3]
> DualHeckeOperator(S,2);
[-3/4  1/8]
[-1/2 11/4]
> // NOT the transpose!
```

We can also compute the Atkin-Lehner and the $*$-involution. The $*$-involution is the identity because we are working in the $+1$-quotient, which is the largest quotient of `ModularSymbols(11,4)` where $*$ acts as $+1$.

```
> AtkinLehner(S,11);
[1 0]
[0 1]
> StarInvolution(S);
[1 0]
[0 1]
```

On the $-1$ quotient the Atkin-Lehner involution is the same, but $*$ acts as $-1$:

```
> M := ModularSymbols(11,4,-1); M;
Full modular symbols space for Gamma_0(11) of weight 4 and dimension 2
over Rational Field
```

```
> S := CuspidalSubspace(M);
> AtkinLehner(S,11);
[1 0]
[0 1]
> StarInvolution(S);
[-1  0]
[ 0 -1]
```

---

**Example H0E16**

We compute an example of our analogue of the $\theta$-operator on modular symbols.

```
> N := 11; p := 3;
> k1 := 2; k2 := k1 + (p+1);
> M1 := ModularSymbols(11,k1,GF(p));
> M2 := ModularSymbols(11,k2,GF(p));
> theta  := ThetaOperator(M1,M2); theta;
Mapping from:  ModSym:  M1 to ModSym:  M2 given by a rule [no inverse]
```

Now that we have computed `theta`, we can apply it to one of the modular symbols corresponding to the newform in $S_2(\Gamma_0(11))$.

```
> D := Decomposition(M1,2);
> f := qEigenform(D[2],10); f;
q + q^2 + 2*q^3 + 2*q^4 + q^5 + 2*q^6 + q^7 + q^9 + O(q^10)
> x := D[2].1;
> y := theta(x); y;
(X^4 + X*Y^3)*{-1/7, 0} + (X^4 + X^3*Y + X*Y^3 + Y^4)*{-1/7, 0} + (X^4
+ 2*X^3*Y + 2*X*Y^3 + Y^4)*{-1/5, 0} + Y^4*{oo, 0}
```

Finally, we verify for $n < 10$ that the $n$th Hecke eigenvalue of $y = \theta(x)$ equals $n \cdot a_n(f)$, where $f$ is as above.

```
> [y*HeckeOperator(M2,n) - n*Coefficient(f,n)*y :  n in [1..9]];
[
0,
0,
0,
0,
0,
0,
0,
0,
0
]
```

---

## 0.10 The Hecke Algebra

`HeckeBound(M)`

A positive integer $n$ such that the Hecke operators $T_1, \ldots, T_n$ generate the Hecke algebra as a $\mathbf{Z}$-module. When the character is trivial, the default bound is $(k/12) \cdot [\mathrm{SL}_2(\mathbf{Z}) : \Gamma_0(N)]$. That this suffices follows from [Stu87], as is explained in [AS]. When the character of $M$ is nontrivial, the default bound is twice the above bound; however, *it is not known that this bound is large enough in all cases in which the character is nontrivial*, so one may wish to increase the bound using `SetHeckeBound`.

`SetHeckeBound(M, n)`

Many computations require a bound $n$ such that $T_1, \ldots, T_n$ generate the Hecke algebra as a $\mathbf{Z}$-module. This command allows you to set the bound that is used internally. Setting it too low can result in functions quickly producing incorrect results.

`HeckeAlgebra(M : Bound)`

The Hecke algebra associated to $M$. This is an algebra `TQ` over $\mathbf{Q}$, such that `Generators(TQ)` is a set that generates the ring $\mathbf{Z}[T_1, T_2, T_3, \ldots]$, as a $\mathbf{Z}$-module. If the optional integer parameter `Bound` is set, then `HeckeAlgebra` only computes the algebra generated by those $T_n$, with $n \leq$ `Bound`.

`DiscriminantOfHeckeAlgebra(M : Bound)`

The discriminant of the Hecke algebra associated to $M$. If the optional parameter `Bound` is set, then the discriminant of the algebra generated by only those $T_n$, with $n \leq$ `Bound`, is computed instead.

`HeckeEigenvalueRing(M : parameters)`

`Bound` RngIntElt *Default :* $-1$

The order generated by the Fourier coefficients of one of the $q$-expansions of a newform corresponding to $M$, along with a map from the ring containing the coefficients of `qExpansion(A)` to the order. If the optional parameter `Bound` is set, then the order generated only by those $a_n$, with $n \leq$ `Bound`, is computed.

`HeckeEigenvalueField(M)`

The number field generated by the Fourier coefficients of one of the $q$-expansions of a newform corresponding to $M$, along with a map from the ring containing the coefficients of `qExpansion(M)` to the number field. We require that $M$ be defined over $\mathbf{Q}$.

**Example H0E17**

In this example, we compute the discriminant of the Hecke algebra of prime level 389.

```
> M := ModularSymbols(389,2,+1);
> C := CuspidalSubspace(M);
```

```
> DiscriminantOfHeckeAlgebra(C);
629670054720061882880174736321392595498204931550235108311040000000
> Factorization($1);
[ <2, 53>, <3, 4>, <5, 6>, <31, 2>, <37, 1>, <389, 1>, <3881, 1>,
<215517113148241, 1>, <477439237737571441, 1> ]
```

The prime 389 is the only prime $p < 10000$ such that $p$ divides the discriminant of the Hecke algebra associated to $S_2(\Gamma_0(p))$. It is an open problem to decide whether or not there are any other such primes. Are there infinitely many?

---

## 0.11 The Intersection Pairing

MAGMA can compute the intersection pairing

$$H_1(X_0(N), \mathbf{Q}) \times H_1(X_0(N), \mathbf{Q}) \to \mathbf{Q}$$

on the homology of the modular curve $X_0(N)$. The algorithm that we implemented is essentially the one given in [Mer93]. (Warning: There is a typo in Proposition 4 of [Mer93]; $W_i$ should be replaced by $W_i^{\varepsilon_i}$.)

`IntersectionPairing(x, y)`

The intersection pairing of the homology classes corresponding to the weight-2 cuspidal modular symbols $x$ and $y$. The symbols $x$ and $y$ must have the same parent, which must have trivial character and not be a $+1$ or $-1$ quotient.

**Example H0E18**

In this example, we illustrate several basic properties of the intersection pairing on $H_1(X_0(37), \mathbf{Z})$. First, let `H37` be the space of modular symbols that corresponds to $H_1(X_0(37), \mathbf{Z})$, and compute a basis for `H37`.

```
> M37 := ModularSymbols(37,2);
> H37 := CuspidalSubspace(M37);
> Z := IntegralBasis(H37); Z;
[
    {-1/29, 0},
    {-1/22, 0},
    {-1/12, 0},
    {-1/18, 0}
]
```

Now we compute some intersection numbers.

```
> IntersectionPairing(Z[1],Z[2]);
-1
> IntersectionPairing(Z[3],Z[4]);
0
```

The intersection pairing is perfect and skew-symmetric, so the matrix that defines it is skew-symmetric and has determinant $\pm 1$ (in fact, it has determinant $+1$).

```
> A := MatrixAlgebra(RationalField(),4);
> I := A![IntersectionPairing(x,y) :  x in Z, y in Z]; I;
[ 0  1  0  1]
[-1  0  1  1]
[ 0 -1  0  0]
[-1 -1  0  0]
> I + Transpose(I) eq 0;
true
> Determinant(I);
1
```

The Hecke operators are compatible with the intersection pairing in the sense that $(T_n x, y) = (x, T_n y)$.

```
> T2 := HeckeOperator(M37,2);
> IntersectionPairing(Z[1]*T2,Z[2]);
1
> IntersectionPairing(Z[1],Z[2]*T2);
1
```

It is note the case $(T_n x, T_n y) = (x, y)$ for all $n$, $x$, and $y$.

```
> IntersectionPairing(Z[1]*T2,Z[2]*T2);
-2
```

The existence of the intersection pairing implies that $H_1(X_0(N), \mathbf{Z})$ is isomorphic, as a module over the Hecke algebra, to its linear dual $\mathrm{Hom}(H_1(X_0(N), \mathbf{Z}), \mathbf{Z})$.

---

## 0.12 $q$-Expansions

The following functions should only be called on modular symbols spaces that are cuspidal. For $q$-expansions of Eisenstein series, use the modular forms functions instead (see the example below).

`qEigenform(M, prec)`

`qEigenform(M)`

`PowerSeries(M, prec)`

`PowerSeries(M)`

The $q$-expansion of one of the Galois-conjugate newforms associated to the irreducible cuspidal space $M$ of modular symbols, computed to absolute precision `prec` (which defaults to the highest precision computed in previous calls to this intrinsic, or 8 if none have been computed). The coefficients of the $q$-expansion lie in a quotient of a polynomial extension of the base field of $M$. In most cases, it is necessary for $M$ to have been defined using `NewformDecomposition`.

`qExpansionBasis(M, prec : parameters)`

`Al` MonStgElt *Default : "Newform"*

The reduced row-echelon basis of $q$-expansions for the space of modular forms associated to $M$, where $K$ is the base field of $M$. The absolute precision of the $q$-expansions is `prec`.

The optional parameter `Al` can take the values `"Newform"` and `"Universal"`. The default is `"Newform"`, which computes a basis of $q$-expansions by finding a decomposition of $M$ into subspaces corresponding to newforms, computing their $q$-expansions, and then taking all of their images under the degeneracy maps. If `Al := "Universal"` then the algorithm of Section 4.3 of [Mer94] is used. This latter algorithm does not require computing a newform decomposition of $M$, but requires computing the action of many more Hecke operators. Consequently, in practice, our implementation of Merel's algorithm is usually less efficient than our implementation of the newform algorithm.

`qIntegralBasis(M, prec : parameters: Al)`

`Al` MonStgElt *Default : "Newform"*

The reduced integral basis of $q$-expansions for the space of modular forms associated to $M$, computed to absolute precision `prec`. The base field of $M$ must be `RationalField()`.

`SystemOfEigenvalues(M, prec)`

The sequence of Hecke eigenvalues $[a_2, a_3, a_5, a_7, \ldots, a_p]$ attached to $M$, where $p$ is the largest prime less than or equal to `prec`. Let $K$ be the base field of $M$. Then the $a_\ell$ either lie in $K$ or a quotient of $K[x]$. We assume that $M$ corresponds to a single Galois-conjugacy class of newforms.

**Example H0E19**

First we compute the a $q$-basis and a representative newform for the two-dimensional space $S_2(\Gamma_0(23))$. We work in the $+1$ quotient of modular symbols since, for the purpose of computing $q$-expansions, nothing is lost and many algorithms are more efficient.

```
> M := CuspidalSubspace(ModularSymbols(23,2, +1));
> qExpansionBasis(M);
[
q - q^3 - q^4 - 2*q^6 + 2*q^7 + O(q^8),
q^2 - 2*q^3 - q^4 + 2*q^5 + q^6 + 2*q^7 + O(q^8)
]
> f := qEigenform(M,6); f;
q + a*q^2 + (-2*a - 1)*q^3 + (-a - 1)*q^4 + 2*a*q^5 + O(q^6)
> Parent(f);
Power series ring in q over Univariate Quotient Polynomial Algebra
in a over Rational Field with modulus a^2 + a - 1
> PowerSeries(M);
q + a*q^2 + (-2*a - 1)*q^3 + (-a - 1)*q^4 + 2*a*q^5 + (a - 2)*q^6 +
```

```
(2*a + 2)*q^7 + O(q^8)
> SystemOfEigenvalues(M, 7);
[
a,
-2*a - 1,
2*a,
2*a + 2
]
```

Next we compare an integral and rational basis of $q$-expansions for $S_2(\Gamma_0(65))$, computed using modular symbols.

```
> S := CuspidalSubspace(ModularSymbols(65,2,+1));
> qExpansionBasis(S);
[
q + 1/3*q^6 + 1/3*q^7 + O(q^8),
q^2 - 1/3*q^6 + 2/3*q^7 + O(q^8),
q^3 - 4/3*q^6 + 2/3*q^7 + O(q^8),
q^4 - 1/3*q^6 + 5/3*q^7 + O(q^8),
q^5 + 5/3*q^6 + 2/3*q^7 + O(q^8)
]
> qIntegralBasis(S);
[
q + q^5 + 2*q^6 + q^7 + O(q^8),
q^2 + 2*q^5 + 3*q^6 + 2*q^7 + O(q^8),
q^3 + 2*q^5 + 2*q^6 + 2*q^7 + O(q^8),
q^4 + 2*q^5 + 3*q^6 + 3*q^7 + O(q^8),
3*q^5 + 5*q^6 + 2*q^7 + O(q^8)
]
```

If you're interested in $q$-expansions of Eisenstein series, see the chapter on modular forms. For example:

```
> E := EisensteinSubspace(ModularForms(65,2));
> Basis(E);
[
1 + O(q^8),
q + 3*q^2 + 4*q^3 + 7*q^4 + 12*q^6 + 8*q^7 + O(q^8),
q^5 + O(q^8)
]
```

## 0.13 Special Values of $L$-functions

Let $M$ be an irreducible space of cuspidal modular symbols defined over $\mathbf{Q}$, irreducible in the sense that $M$ corresponds to a single Galois-conjugacy class of cuspidal newforms. Such an $M$ can be computed using `NewformDecomposition`. Let $f^{(1)}, \ldots, f^{(d)}$ be the $\mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q})$-conjugate newforms that correspond to $M$, and write $f^{(d)} = \sum_{n=1}^{\infty} a_n^{(d)} q^n$. By a theorem of Hecke, the Dirichlet series

$$L(f^{(i)}, s) = \sum_{n=1}^{\infty} \frac{a_n^{(i)}}{n^s}$$

extends (uniquely) to a holomorphic function on the whole complex plane. Of particular interest is the special value

$$L(M, j) = L(f^{(1)}, j) \cdots L(f^{(d)}, j),$$

for any $j \in \{1, 2, \ldots, k-1\}$.

In this section we describe how to approximate the complex numbers $L(M, j)$ in Magma. If you are interested in computing individual special values $L(f^{(i)}, j)$, then you should use the modular forms package instead of the modular symbols package for this.

The variable `prec` below refers to the number of terms of the $q$-expansion of each $f^{(i)}$ that are used in the computation, and not to the number of decimals of the answer that are correct. Thus, for example, to get a heuristic idea of the quality of an answer, you can increase `prec`, make another call to `LSeries`, and observe the difference between the two answers. If the difference is "small", then the approximation is probably "good".

`LSeries(M, j, prec)`

The special value $L(M, j)$, where $j$ is an integer that lies in the critical strip, so $1 \leq j \leq k-1$ with $k$ the weight of $M$. Here $M$ is a space of modular symbols with sign 0, and `prec` is a positive integer which specifies the numbers of terms of $q$-expansions to use in the computation.

`LSeriesLeadingCoefficient(M, j, prec)`

The leading coefficient of Taylor expansion about the critical integer $j$ and order of vanishing of $L(M, s)$ at $s = 1$. Thus if the series expansion of $L(M, s)$ about $s = 1$ is

$$L(M, s) = a_r(s-1)^r + a_{r+1}(s-1)^{r+1} + a_{r+2}(s-1)^{r+2} + \cdots,$$

then the leading coefficient of $L(M, s)$ is $a_r$ and the order of vanishing is $r$.

`RealVolume(M, prec)`

The volume of $A_M(\mathbf{R})$, which is defined as follows. Let $S \subset \mathbf{C}[[q]]$ be the space of cusp forms associated to $M$. Choose a basis $f_1, \ldots, f_d$ for the free $\mathbf{Z}$-module $S \cap \mathbf{Z}[[q]]$; one can prove that $f_1, \ldots, f_d$ is also a basis for $S$. There is a period map $\Phi$ from integral cuspidal modular symbols $H$ to $\mathbf{C}^d$ that sends a modular

symbol $x \in H$ to the $d$-tuple of integrals $(\langle f_1, x\rangle, \ldots, \langle f_d, x\rangle) \in \mathbf{C}^d$. The cokernel of $\Phi$ is isomorphic to $A_M(\mathbf{C})$. Moreover, the standard measure on the Euclidean space $\mathbf{C}^d$ induces a measure on $A_M(\mathbf{R})$. It is with respect to this measure that we compute the volume. For more details, see Section 3.12.16 of [Ste00].

`MinusVolume(M, prec)`

The volume of the subgroup of $A_M(\mathbf{C})$ on which complex conjugation acts as $-1$.

`LRatio(M, j : parameters)`

`Bound` RngIntElt *Default :* $-1$

The rational number

$$\frac{L(A, j) \cdot (j-1)!}{(2\pi)^{j-1} \cdot \Omega,}$$

where $j$ is a "critical integer", so $1 \leq j \leq k-1$, and $\Omega$ is `RealVolume(M)` when $j$ is odd and `MinusVolume(M)` when $j$ is even. If the optional parameter Bound is set, then `LRatio` is only a divisibility upper bound on the above rational number. If `Sign(M)` is not 0, then `LRatio(M,j)` is only correct up to a power of 2.

`LRatioOddPart(M, j)`

The odd part of the rational number `LRatio(M,j)`. Hopefully, computing `LRatioOddPart(M,j)` takes less time than finding the odd part of `LRatio(M,j)`.

**Example H0E20**

```
> M := ModularSymbols(11,2);
> C := CuspidalSubspace(M);
> LSeries(C,1,100);
0.25384186085591068433775892 33
> A := ModularSymbols("65B"); A;  // <--> dimension two abelian variety
Modular symbols space of level 65, weight 2, and dimension 4
> LSeries(A,1,100);
0.91225158869818984109351402 11 +  0.E-29*i
```

## 0.13.1 Winding Elements

Let $\mathbf{M_k}(\mathbf{N})$ be a space of modular symbols over $\mathbf{Q}$. For $i = 1, \ldots, k$, the *ith winding element*

$$\mathbf{e}_i = X^{i-1}Y^{k-2-(i-1)}\{0, \infty\} \in \mathbf{M_k}(\mathbf{N})$$

is of importance for the computation of special values. For any modular form $f \in S_k(N)$ and homogeneous polynomial $P(X, Y)$ of degree $k-2$, let

$$\langle f, P(X, Y)\{0, \infty\}\rangle = -2\pi i \cdot \int_0^{i\infty} f(z)P(z, 1)\mathrm{dz}.$$

Fix a newform $f \in S_k(N)$ corresponding to a space $M$ of modular symbols, and let $j$ be a integer in $\{0, 1, \ldots, k-1\}$. The winding element is significant because

$$L(f, j) = \frac{(2\pi)^{j-1}}{i^{j+1}(j-1)!} \cdot \langle f, X^{j-1}Y^{k-2-(j-1)}\{0, \infty\}\rangle.$$

Moreover, the submodule that is generated by the winding element is used in the formula for a canonical rational part of the number $L(M, j)$ (see `LRatio`, above).

`WindingElement(M)`

The winding element $Y^{k-2}\{0, \infty\}$.

`WindingElement(M, i)`

The winding element $X^{i-1}Y^{k-2-(i-1)}\{0, \infty\}$.

`TwistedWindingElement(M, i, eps)`

The element $\sum_{a \in (\mathbf{Z}/m\mathbf{Z})^*} \varepsilon(a) X^{i-1}Y^{k-2-(i-1)}\{0, \frac{a}{m}\}$.

`WindingLattice(M, j : parameters)`

`Bound` RngIntElt *Default :* $-1$

The image under `RationalMapping(M)` of the lattice generated by the images of the $j$th winding element under all Hecke operators $T_n$. If $M$ is the ambient space, then the image under `RationalMapping(M)` is not taken.

`WindingSubmodule(M, j : parameters)`

`Bound` RngIntElt *Default :* $-1$

The image under `RationalMapping(M)` of the vector space generated by all images of `WindingElement(M,j)` under all Hecke operators $T_n$. If $M$ is the ambient space, then the image under the rational period mapping is not taken.

`TwistedWindingSubmodule(M, j, eps)`

The Hecke submodule of the vector space $\Phi(M)$ generated by the image of the $j$th $\varepsilon$-twisted modular winding element, where $\Phi$ is `RationalMapping(M)`. This module is useful, for example, because in characteristic 0, if $M$ is new of weight 2, has sign $+1$ or $-1$, and corresponds to a collection $\{f_i\}$ of Galois-conjugate newforms, then the dimension of the twisted winding submodule equals the cardinality of the subset of $f_i$ such that $L(f_i, eps, 1) \neq 0$.

## 0.14 The Associated Complex Torus

Let $M$ be a space of cuspidal modular symbols, which is the kernel of an ideal in the Hecke algebra. When $M$ has weight 2 there is an abelian variety $A_M$ attached to $M$; more generally, there is a complex torus $A_M(\mathbf{C})$ attached to $M$. The associated complex torus $A_M(\mathbf{C})$ is constructed as follows. Let $S$ be the space of modular forms corresponding to $M$. The integration pairing gives rise to a natural map $M \to \mathrm{Hom}(S, \mathbf{C})$, and the cokernel of this map is $A_M(\mathbf{C})$.

`SubgroupOfTorus(M, x)`

The cyclic subgroup of the complex torus attached to $M$ that is generated by the image under the period map of the modular symbol $x$.

`SubgroupOfTorus(M, s)`

An abelian group that is isomorphic to the finite group generated by the sequence of images $\pi(s[i])$ in the complex torus attached to $M$, where $\pi$ is `PeriodMapping(M)`.

**Example H0E21**

The cuspidal subgroup of $J_0(N)$ is the subgroup generated by the degree 0 divisors on $X_0(N)$ of the form $(\alpha) - (\beta)$, where $\alpha$ and $\beta$ are cusps. The following examples illustrate how to use the above functions to compute the cuspidal subgroup, as an abstract abelian group.
The modular symbols approach has the advantage that it is essentially no more complicated for $N$ highly composite than for $N$ prime. However, it is only applicable when the corresponding space of modular symbols can be computed in a reasonable amount of time, which at present means that $N$ should have less than 5 decimal digits. There are other methods which may be much more efficient in special cases. For example, when $p$ is prime Andrew Ogg showed that the cuspidal subgroup of $J_0(p)$ is cyclic of order equal to the numerator of $(p-1)/12$. More generally, he gave a simple formula for the order of the cuspidal subgroup when $N = pq$ is the product of two primes. See also papers of Ligozat.

```
> M := ModularSymbols(20); M;
Full Modular symbols space of level 20, weight 2, and dimension 7
> e := M ! <1, [Cusps()|0,Infinity()] >;   // the path from 0 to infinity
> e;
-1*{oo, 0}
> J0of20 := CuspidalSubspace(M);
> A := SubgroupOfTorus(J0of20, e); A;
Abelian Group isomorphic to Z/6
Defined on 1 generator
Relations:
    6*A.1 = 0
> // Next, the subgroup generated by all cusps
> A := SubgroupOfTorus(J0of20, IntegralBasis(M));  A;
Abelian Group isomorphic to Z/6
Defined on 1 generator
Relations:
    6*A.1 = 0
```

```
> // Let's do another example.
> M := ModularSymbols(100);
> J0of100 := CuspidalSubspace(M);
> A := SubgroupOfTorus(J0of100, IntegralBasis(M));  A;
Abelian Group isomorphic to Z/6 + Z/30 + Z/30 + Z/30 + Z/30
Defined on 5 generators
Relations:
    6*A.1 = 0
    30*A.2 = 0
    30*A.3 = 0
    30*A.4 = 0
    30*A.5 = 0
> M := ModularSymbols(77);
> J0of77 := CuspidalSubspace(M);
> A := SubgroupOfTorus(J0of77, IntegralBasis(M)); A;
Abelian Group isomorphic to Z/10 + Z/60
Defined on 2 generators
Relations:
    10*A.1 = 0
    60*A.2 = 0
> M := ModularSymbols(97);
> A := SubgroupOfTorus(CuspidalSubspace(M), IntegralBasis(M)); A;
Abelian Group isomorphic to Z/8
Defined on 1 generator
Relations:
    8*A.1 = 0
> Numerator((97-1)/12);
8
```

**Example H0E22**

The following code creates a file that contains a table which lists, for each integer $N$ in some range, the abstract group structure of the subgroup of $J_0(N) = \mathrm{Jac}(X_0(N))$ generated by the cusps $(\alpha) - (\infty)$, with $\alpha \in \mathbf{Q} \cup \{\infty\}$.

```
> function CuspidalSubgroup(N)
>    M := ModularSymbols(N);
>    J := CuspidalSubspace(M);
>    G := SubgroupOfTorus(J,IntegralBasis(M));
>    DisownChildren(M);
>    return G;
> end function;
> // Test the function
> CuspidalSubgroup(65);
Abelian Group isomorphic to Z/2 + Z/84
Defined on 2 generators
Relations:
    2*$.1 = 0
```

```
    84*$.2 = 0
> procedure CuspidalTable(start, stop)
>    fname := Sprintf("cuspidal_subgroup_%o-%o.m", start, stop);
>    file  := Open(fname,"w");
>       for N in [start..stop] do
>       G := Invariants(CuspidalSubgroup(N));
>       fprintf file, "C[%o] := \t%o;\n\n", N, G;
>       printf "C[%o] := \t%o;\n\n", N, G;
>       Flush(file);
>    end for;
> end procedure;
```

---

`ModularKernel(M)`

The kernel of the modular isogeny. Let $T$ be the complex torus attached to $M$. Then the modular isogeny is the natural map from the dual of T into T induced by autoduality of `CuspidalSubspace(AmbientSpace(M))`.

`CongruenceGroup(M : parameters)`

`Bound` RngIntElt *Default :* $-1$

The congruence group of the space of cusp forms corresponding to the space of cuspidal modular symbols $M$. Let $S = S_k(\Gamma_0(N), \mathbf{Z})$, let $V$ be the sub $\mathbf{Z}$-module corresponding to $M$, and $W$ be its orthogonal complement. Then the congruence group is $S/(V + W)$. This group encodes information about congruences between forms in $V$ and forms in the complement of $V$.

The optional parameter `Bound` is a positive integer $b$ such that the $q$-expansions of cusp forms are computed to absolute precision $b$. If the bound is too small, then `CongruenceGroup` will give only an upper bound on the correct answer. The default is `HeckeBound(M) + 1`, which gives a provably correct answer.

`IntersectionGroup(M1, M2)`

An abelian group $G$ that encodes information about the intersection of the complex tori corresponding to $M_1$ and $M_2$. We require that $M_1$ and $M_2$ lie in a common ambient space. When the `IntersectionGroup(M1,M2)` is finite, it is isomorphic to $A_{M_1}(\mathbf{C}) \cap A_{M_2}(\mathbf{C})$.

`IntersectionGroup(S)`

An abelian group $G$ that encodes information about the intersection of the collection of complex tori corresponding to the sequence $S$ of spaces of modular symbols.

**Example H0E23**

In this example, we investigate a 2-dimensional abelian variety $B$, which is a quotient of $J_0(43)$. The purpose of this example is to show how numerical computation with modular symbols suggests interesting arithmetic questions about familiar abelian varieties. In the following example, we find that the conjecture of Birch and Swinnerton-Dyer (plus the Manin $c = 1$ conjecture) implies that

the first nontrivial Shafarevich-Tate group of an (optimal) modular abelian variety has order TWO. Thus the surprising existence of an abelian varieties with non-square order could have been (but was not) hinted at long ago by somebody playing around with a modular symbols package (in fact, it was discovered by B. Poonen and M. Stoll [PS99] while they were designing and implementing algorithms for computing with Jacobians of genus-two curves).

```
> M43 := ModularSymbols(43,2);     // Level 43, weight 2.
> H1  := CuspidalSubspace(M43);    // H_1(X_0(43),Q)
> D   := NewformDecomposition(H1); // factors corresponding to newforms
> A,B := Explode(D);
> A;     // The homology of the elliptic curve "43A"
Modular symbols space of level 43, weight 2, and dimension 2
> B;     // The homology of the 2-dimensional abelian variety "43B"
Modular symbols space of level 43, weight 2, and dimension 4
> LRatio(B,1);        // L(B,1)/Omega_B
2/7
```

The Birch and Swinnerton-Dyer conjecture predicts that the Shafarevich-Tate group of $B$ has order as given by the formula for `ShaAn` in the code below. To compute this value, it remains to compute $\#B(\mathbf{Q})$ and the Tamagawa number $c_{43}$.

```
> T := TorsionBound(B,11);  T; // #B(Q) divides this number
7
> // Compute the subgroup of B(Q) generated by (0)-(oo).
> C := SubgroupOfTorus(B,WindingElement(M43));  C;
Abelian Group isomorphic to Z/7
Defined on 1 generator
Relations:
    7*C.1 = 0
> TamagawaNumber(B,43);
7
> ShaAn := LRatio(B,1)*TorsionBound(B,11)^2/TamagawaNumber(B,43);
```

`ShaAn` is the Birch and Swinnerton-Dyer conjectural order of the Shafarevich-Tate group of $B$, under the assumption that the Manin constant of $B$ is 1.

```
> ShaAn;
2
```

One of the Galois conjugate newforms associated to $B$ is given below.

```
> qEigenform(B,12);
q + a*q^2 - a*q^3 + (-a + 2)*q^5 - 2*q^6 + (a - 2)*q^7 - 2*a*q^8 - q^9
    + (2*a - 2)*q^10 + (2*a - 1)*q^11 + O(q^12)
> BaseRing(Parent(qEigenform(B,12)));
Univariate Quotient Polynomial Algebra in a over Rational Field
with modulus a^2 - 2
> qIntegralBasis(B,12);
[
    q + 2*q^5 - 2*q^6 - 2*q^7 - q^9 - 2*q^10 - q^11 + O(q^12),
    q^2 - q^3 - q^5 + q^7 - 2*q^8 + 2*q^10 + 2*q^11 + O(q^12)
```

```
]
```

By integrating homology against the differentials corresponding to the two modular forms above, we obtain a lattice that defines the complex torus $A_B(\mathbf{C})$:

```
> Periods(B,97);
[
    (-0.22594995830676421187395192 -
        1.76664467629959953227333140*i
        0.52502811591322194337294916 +
        0.80660185770293072302831423 71*i),
    (0.59815631622412229864757672 20 -
        1.92008563861211949327648563 2*i
        0.82410627422619603486491720 82 -
        0.15344096225717705687483549 95*i),
    (-0.82410627453088651052152864 45 -
        0.15344096231251996100315249 20*i
        -0.29907815831297409149196804 34 -
        0.96004281996010777990314973 67*i),
    (-0.59815631622412229864757672 20 -
        1.92008563861211949327648563 2*i
        -0.82410627422619603486491720 83 -
        0.15344096225717705687483549 95*i)
]
```

Finally, it is tempting to ask whether or not the (conjectural) two-torsion element of the Shafarevich-Tate group of $B$ suggested above is "visible" in the sense that it is "explained by a jump in the rank of the Mordell-Weil group of $A$" (see [CM00]). The following computation suggests, but does not prove, that this is the case.

```
> G := MordellWeilGroup(EllipticCurve(A)); G;
Abelian Group isomorphic to Z
Defined on 1 generator (free)
> IntersectionGroup(A,B);
Abelian Group isomorphic to Z/2 + Z/2
Defined on 2 generators
Relations:
    2*$.1 = 0
    2*$.2 = 0
```

---

### 0.14.1 The Period Map

Let $M$ be a space of modular symbols the corresponds to a Galois-conjugacy class of newforms. The *period map* attached to $M$ is a linear map

$$\text{AmbientSpace(M)} \to \mathbf{C}^d,$$

where $d$ is the dimension of the space of modular forms associated to $M$. The cokernel of the period map is a complex torus $A_M(\mathbf{C})$. The terminology "period mapping" comes

from the fact that there are (often?) mereomorphic functions on $\mathbf{C}^d$ whose periods are the image of the integral cuspidal modular symbols under the period mapping.

In the functions below, $M$ must not be a $+1$ or $-1$ quotient and must be cuspidal.

`PeriodMapping(M, prec)`

The period mapping attached to $M$, computed using `prec` terms of the $q$-expansions of modular forms associated to $M$.

`Periods(M, prec)`

The complex period lattice associated to $M$, computed using `prec` terms of the $q$-expansions of modular forms associated to $M$.

`ClassicalPeriod(M, j, prec)`

The value

$$r_j(f) = \int_0^{i\infty} f(z) z^j dz.$$

### 0.14.2 Projection Mappings

Let $M$ be a space of modular symbols over a field $K$. For many purposes it is useful to have a surjective map

$$\pi : \text{AmbientSpace(M)} \to V,$$

where $V$ is a vector space over $K$ and $\ker(\pi)$ is the same as the kernel of the period mapping.

`RationalMapping(M)`

A surjective linear map from the ambient space of $M$ to a vector space, such that the kernel of this map is the same as the kernel of the period mapping.

`IntegralMapping(M)`

A surjective linear map from the ambient space of $M$ to a vector space, such that the kernel of this map is the same as the kernel of the period mapping. This map is chosen in such a way that the image of

`IntegralBasis(CuspidalSubspace(AmbientSpace(M)))`

is the standard $\mathbf{Z}$-lattice. (Note that $M$ must be defined over $\mathbf{Q}$.)

**Example H0E24**

```
> M := ModularSymbols(33); M;
Full Modular symbols space of level 33, weight 2, and dimension 9
> S := CuspidalSubspace(M);
> N := NewSubspace(S);
> phi := RationalMapping(N);
> [phi(x) :  x in IntegralBasis(S)];
[
```

```
    ( -2 4/3),
    ( -4 2/3),
    ( -2 2/3),
    (-2  0),
    (  -2 -2/3),
    (-4  0)
]
```

Notice that the image of the basis `IntegralBasis(S)` for $H_1(X_0(33), \mathbf{Z})$ is not $\mathbf{Z} \times \mathbf{Z}$. However, `IntegralMapping(N)` is normalized so that the image is $\mathbf{Z} \times \mathbf{Z}$:

```
> int := IntegralMapping(N);
> [int(S.i) :  i in [1..Dimension(S)]];
[
    ( 2 -1),
    ( 1 -2),
    ( 1 -1),
    ( 0 -2),
    ( 0 -1),
    (-1 -1)
]
```

Consider a quotient $A_f$ of $J_0(N)$ attached to a newform $f \in S_2(\Gamma_0(N))$. Using `IntegralMapping` and the Abel-Jacobi theorem, we can see the image in $A_f(\mathbf{Q})$ of the point $(0) - (\infty) \in J_0(N)(\mathbf{Q})$. In the level 97 example below, this image has order 8, which is the numerator of $(97-1)/12$.

```
> Af := ModularSymbols("97B"); Af;
Modular symbols space of level 97, weight 2, and dimension 8
> int := IntegralMapping(Af);
> // Let x be the modular symbol {0,oo}
> x := AmbientSpace(Af)!<1,[Cusps()|0,Infinity()]>;
> int(x);
(-5/8  1/4 -1/4    0    0  1/4  3/8  1/4)
> Numerator((97-1)/12);
8
```

---

## 0.15 Modular Abelian Varieties

Let $M$ be a space of weight 2 cuspidal modular symbols with trivial character that corresponds to a Galois-conjugacy class of newforms, and let $A_M(\mathbf{C})$ be the cokernel of the period map. G. Shimura proved that $A_M(\mathbf{C})$ is the set of complex points of an abelian variety $A_M$ defined over $\mathbf{Q}$. Let $N$ be the level of $M$ and let $J_0(N)$ be the Jacobian of the modular curve $X_0(N)$. Shimura constructed $A_M$ as a quotient of $J_0(N)$ by an abelian subvariety. More precisely, if $I$ is the annihilator of $M$ in the Hecke algebra, then $A_M = J_0(N)/IJ_0(N)$.

When $A_M$ has dimension 1 it is an elliptic curve, and the theory of computing with $A_M$ is well developed, though many interesting problems remain. In the contrary case, when $A_M$ has dimension greater than 1, the theory of computation with $A_M$ is still in its

infancy. Fortunately, it is possible to compute a number of interesting quantities about $A_M$ using algorithms that rely on our extensive knowledge of $J_0(N)$.

MAGMA contains functions for computing the modular degree, congruence modulus, upper and lower bounds on the order of the torsion subgroup, and the order of the component group of the closed fiber of the Néron model of $A_M$ at primes that exactly divide the level of $M$.

### 0.15.1 Modular Degree and Torsion

`ModularDegree(M)`

The modular degree of $M$, which is defined as follows. Let $M$ be a space of modular symbols of weight 2 and trivial character. The modular degree of $M$ is the square root of `#ModularKernel(M)`. When $M$ corresponds to an elliptic curve $E = A_M$, then the modular degree of $M$ is the degree of induced map $X_0(N) \to E$.

`CongruenceModulus(M : parameters)`

`Bound` RNGINTELT *Default :* $-1$

The congruence number $r$ of $M$. This is the index in $S_k(\Gamma_0(N), \mathbf{Z})$ of the sum $L+W$ of the lattice $W$ of cusp forms $L$ corresponding to $M$ and the lattice of cusp forms corresponding to the complement of $L$ in $S$.

`TorsionBound(M, maxp)`

The following upper bound on the order of the torsion subgroup of the abelian variety $A$ attached to $M$:

$$\gcd\{\#A(\mathbf{F}_p) : 3 \leq p \leq \text{maxp},\ \ p \not| N\},$$

where $N$ is the level of $M$. This bound is an isogeny invariant, so it is also a bound on the order of the torsion subgroup of the dual abelian variety $A^\vee$ of $A$.

To compute a lower bound, use `#SubgroupOfTorus(M,WindingElement(M))`.

**Example H0E25**

We compute the first example of an optimal elliptic curve over $\mathbf{Q}$ such that the congruence modulus does not equal the modular degree. (See [FM99] for further discussion of this problem. We warn the reader that the divisibility $r \mid \deg(\varphi) \mid rN^i$ cited there is incorrect, as our `54B` example shows.)

```
> E := ModularSymbols("54B");
> ModularDegree(E);
2
> CongruenceModulus(E);
6
```

We next verify directly that the congruence modulus is divisible by 3.

```
> A := ModularSymbols("27A"); A;   // 27=54/2.
Modular symbols space of level 27, weight 2, and dimension 2
```

```
> A54 := ModularSymbols(A,54); A54;  // all images of A at level 54.
Modular symbols space of level 54, weight 2, and dimension 4
> qE := qIntegralBasis(E,17);
> qA54 := qIntegralBasis(A54,17);
> &+qA54 - &+qE;
-3*q^4 + 3*q^5 - 3*q^8 + 3*q^10 - 3*q^11 + 9*q^13 + 3*q^16 + O(q^17)
> IntersectionGroup(E,A54);   // however, the intersection is trivial.
Abelian Group of order 1
```

Ken Ribet proved that if $E$ is an optimal elliptic curve quotient of $J_0(N)$, with $N$ prime, and if $f_E$ is the corresponding newform, then the congruence modulus of $f_E$ equals the modular degree of $E$. The author is aware of no counterexamples to the following more general statement: "If $E$ is an optimal elliptic curve of square-free conductor, then the congruence modulus of the newform $f_E$ attached to $E$ equals the modular degree of $E$." An analogous statement for abelian varieties is false, even at prime level. The first counterexample is `ModularSymbols("431F")`, which corresponds to an abelian variety of dimension 24. In this case, the modular degree is $2^{11} \cdot 6947$, whereas the congruence modulus is $2^{10} \cdot 6947$.
The following code makes a table of congruence moduli and modular degrees for the elliptic curves of conductor near 54. Notice the counterexample at level 54.

```
> for N in [53..55] do
>    C := CuspidalSubspace(ModularSymbols(N,2));
>    newforms := NewSubspace(C);
>    D := EllipticFactors(newforms,19);
>    for E in D do
>       printf "%o:\t%o,\t%o\n", N, ModularDegree(E), CongruenceModulus(E);
>    end for;
> end for;
53:          2,      2
54:          2,      6
54:          6,      6
55:          2,      2
```

`ModularKernel` makes sense even for spaces of modular symbols of weight greater than 2. As in the case of weight 2, this number gives information about congruences between modular forms. The following example illustrates how `ModularKernel` suggest a congruence between a form of level 10 and weight 4 with a form of level 5.

```
> M := ModularSymbols(10,4);
> S := CuspidalSubspace(M);
> D := NewformDecomposition(S); D;
[
Modular symbols space of level 10, weight 4, and dimension 2,
Modular symbols space of level 10, weight 4, and dimension 4
]
> #ModularKernel(D[1]);
10
> f := qEigenform(D[1],8);
> g := qEigenform(D[2],8);
> g2 := Evaluate(g,Parent(g).1^2);
```

```
> f-(g+6*g2);   // a congruence modulo 10!
-10*q^3 + 20*q^4 + 10*q^5 - 20*q^6 - 10*q^7 + O(q^8)
```

### 0.15.2 Tamagawa Numbers and Orders of Component Groups

We provide several functions for computing the orders of component groups of optimal quotients of $J_0(N)$ at primes $p$ that exactly divide $N$. Our algorithm involves Grothendieck's monodromy pairing on the character group of the toric part of the closed fiber at $p$ of the Néron model of $J_0(N)$; the theory behind this algorithm is described in [Ste01] (or [Ste00]); see [KS00] for a computationally-oriented introduction to the algorithm. When $N$ is prime, we use the Mestre and Oesterlé method to construct the character group of the torus, as described in [Mes86]. In general, the ideal theory of quaternion algebras is used.

**Note:** In the appendix to [Maz77], Mazur and Rapaport give an explicit formula for the order of the component group of $J_0(N)$ at primes $p \geq 5$ that exactly divide $N$. Their formula is not currently used by the `ComponentGroupOrder` function.

The `RealTamagawaNumber` function computes the order of the "component group at infinity".

`ComponentGroupOrder(M, p)`

The order of the component group at $p$. This is the order of the group of $\overline{\mathbf{F}}_p$-points of the component group of the reduction modulo $p$ of the Néron model of the abelian variety attached to $M$. At present, it is necessary that $p$ exactly divides the level. If `Sign(M)` is not equal to 0, then only the odd part of the order is returned.

`TamagawaNumber(M, p)`

The order of the group of $\mathbf{F}_p$-rational points of the component group of $M$. We require $M$ to be associated to a single Galois-conjugacy class of newforms.

`RealTamagawaNumber(M)`

The number of connected components of $A_M(\mathbf{R})$.

`MinusTamagawaNumber(M)`

The number of connected components of the subgroup $A_M(\mathbf{C})^-$ of $A_M(\mathbf{C})$ on which complex conjugation acts as $-1$

**Example H0E26**

We compute the orders of the component groups of some abelian varieties.

```
> X11 := ModularSymbols("11A");     // corresponds to X_0(11).
> ComponentGroupOrder(X11,11);
5
> TamagawaNumber(X11,11);
5
> RealTamagawaNumber(X11);
```

```
1
> MinusTamagawaNumber(X11);
1
> J37 := ModularSymbols("37"); J37;
Modular symbols space of level 37, weight 2, and dimension 4
> ComponentGroupOrder(J37,37);
3
> A, B := Explode(NewformDecomposition(J37));
> ComponentGroupOrder(A,37);
3
> ComponentGroupOrder(B,37);
1
```

We can also compute component groups of optimal quotients whose dimension is greater than 1. The abelian varieties `B` and `C` below correspond to the Jacobians labeled $65B$ and $65A$ in [FLS+02], respectively.

```
> J65 := ModularSymbols("65");
> A,B,C := Explode(SortDecomposition(NewformDecomposition(J65)));
> B;
Modular symbols space of level 65, weight 2, and dimension 4
> C;
Modular symbols space of level 65, weight 2, and dimension 4
> ComponentGroupOrder(B,5);      // not the Tamagawa number
3
> ComponentGroupOrder(B,13);
3
> ComponentGroupOrder(C,5);
7
> ComponentGroupOrder(C,13);
1
> HeckeEigenvalueField(C);
Number Field with defining polynomial x^2 + 2*x - 1 over the
Rational Field
Mapping from:  Univariate Quotient Polynomial Algebra in a over
Rational Field
with modulus a^2 + 2*a - 1 to Number Field with defining
polynomial x^2 + 2*x - 1 over the Rational Field given by a rule
[no inverse]
> ComponentGroupOrder(J65,5);
42
```

When the Atkin-Lehner involution $W_p$ acts as $+1$ on a modular abelian variety $A$, the order of the component group can be larger than the Tamagawa number $c_p = [A(\mathbf{Q}_p) : A_0(\mathbf{Q}_p)]$ that appears in the conjecture of Birch and Swinnerton-Dyer.

```
> AtkinLehner(B,5);
[1 0 0 0]
[0 1 0 0]
[0 0 1 0]
```

```
[0 0 0 1]
> ComponentGroupOrder(B,5);
3
> TamagawaNumber(B,5);
1
```

The real and minus Tamagawa numbers are defined for spaces of modular symbols of any weight over the rationals.

```
> Del := ModularSymbols("1k12A");
> Del;
Modular symbols space of level 1, weight 12, and dimension 2
```

Next we see that the period lattice associated to $\Delta$ is rectangular.

```
> RealTamagawaNumber(Del);
2
> MinusTamagawaNumber(Del);
2
> Periods(Del,40);
[
    (-0.00048533816492995160492413044429*i),
    (0.0011407374495830793360445453337)
]
```

---

## 0.16 Elliptic Curves

Let $E$ be an elliptic curve. By the modularity theorem, which was recently proved by Breuil, Conrad, Diamond, Taylor, and Wiles there is a two-dimensional space $M$ of modular symbols attached to $E$. Let $N$ be the conductor of $E$; then $M$ is obtained from `ModularSymbols(N,2)` by intersecting the kernels of $T_p - a_p(E)$ for sufficiently many $p$.

**Warning:** The computation of $M$ can already be very resource intensive for elliptic curves for which `Conductor(E)` is on the order of 5000. For example, the seemingly harmless expression `ModularSymbols(EllipticCurve([0,6]))` would bring my computer to its knees.

`ModularSymbols(E)`

`ModularSymbols(E, sign)`

The space $M$ of modular symbols associated to the elliptic curve E.

**Example H0E27**

We use the elliptic curve functions to numerically compute the Birch and Swinnerton-Dyer conjectural order of the Shafarevich-Tate group of the elliptic curve **389A**, which is the curve of rank 2 with smallest conductor. The Birch and Swinnerton-Dyer conjecture asserts that

$$\frac{L^{(r)}(E,1)}{r!} = \frac{\prod c_p \cdot \mathrm{Sha} \cdot \mathrm{Reg}}{|E(\mathbf{Q})_{\mathrm{tor}}|^2},$$

where $r$ is the order of vanishing of $L(E, s)$ at $s = 1$.

```
> E := EllipticCurve(CremonaDatabase(),"389A");
> M := ModularSymbols(E);
> M;
Modular symbols space of level 389, weight 2, and dimension 2
> LRatio(M,1);
0
```

Next we compute the analytic rank and the leading coefficient of the $L$-series at $s = 1$. (If your computer is very slow, use a number smaller than 300 below.)

```
> L1, r := LSeriesLeadingCoefficient(M,1,300);
> L1;
0.75931650029224679065762600 31
> r;        // The analytic rank is 2.
2
```

Finally we check that the rank conjecture is true in this case, and compute the conjectural order of the Shafarevich-Tate group.

```
> Rank(E);  // The algebraic rank is 2.
2
> Omega := RealVolume(M,300); Omega;
4.980435433609741580582713757
> Reg := Regulator(E); Reg;
0.1524601779431437875
> #TorsionSubgroup(E);
1
> TamagawaNumber(E,389);
1
> TamagawaNumber(M,389);        // entirely different algorithm
1
> Sha := L1/(Omega*Reg); Sha;
0.9999979295234896211
```

---

## 0.17 Dimension Formulas

`DimensionCuspFormsGamma0(N, k)`

The dimension of the space $S_k(\Gamma_0(N))$ of weight $k$ cusp forms for $\Gamma_0(N)$.

`DimensionNewCuspFormsGamma0(N, k)`

The dimension of the new subspace of the space $S_k(\Gamma_0(N))$ of weight $k$ cusp forms for $\Gamma_0(N)$.

`DimensionCuspFormsGamma1(N, k)`

The dimension of the space $S_k(\Gamma_1(N))$ of weight $k$ cusp forms for $\Gamma_1(N)$.

`DimensionNewCuspFormsGamma1(N, k)`

The dimension of the new subspace of the space $S_k(\Gamma_1(N))$ of weight $k$ cusp forms for $\Gamma_1(N)$.

`DimensionCuspForms(eps, k)`

The dimension of the space $S_k(\Gamma_1(N))(\varepsilon)$ of cusp forms of weight $k$ and Dirichlet character `eps`. The level $N$ is the modulus of `eps`. The dimension is computed using the formula of Cohen and Oesterlè (see [CO77]).

**Example H0E28**

```
> DimensionCuspFormsGamma0(11,2);
1
> DimensionCuspFormsGamma0(1,12);
1
> DimensionCuspFormsGamma0(5077,2);
422
> DimensionCuspFormsGamma1(5077,2);
1071460
> G := DirichletGroup(5*7);
> eps := G.1*G.2;
> IsOdd(eps);
true
> DimensionCuspForms(eps,2);
0
> DimensionCuspForms(eps,3);
6
```

The dimension of the space of cuspidal modular symbols is twice the dimension of the space of cusp forms.

```
> Dimension(CuspidalSubspace(ModularSymbols(eps,3)));
12
```

## 0.18 Bibliography

[**AS**] Amod Agashe and William A. Stein. *Appendix to Joan-C. Lario and René Schoof:* Some computations with Hecke rings and deformation rings. *submitted*.

[**CM00**] John E. Cremona and Barry Mazur. Visualizing elements in the Shafarevich-Tate group. *Experiment. Math.*, 9(1):13–28, 2000.

[**CO77**] Henri Cohen and J. Oesterlé. Dimensions des espaces de formes modulaires. pages 69–78. Lecture Notes in Math., Vol. 627, 1977.

[**Cre92**] John E. Cremona. Modular symbols for $\Gamma_1(N)$ and elliptic curves with everywhere good reduction. *Math. Proc. Cambridge Philos. Soc.*, 111(2):199–218, 1992.

[**Cre97**] John E. Cremona. *Algorithms for modular elliptic curves*. Cambridge University Press, Cambridge, second edition, 1997.

[**DI95**] Fred Diamond and Ju Im. Modular forms and modular curves. In *Seminar on Fermat's Last Theorem*, pages 39–133. Providence, RI, 1995.

[**FLS$^+$02**] 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. *to appear in Math. of Comp.*, 2002.

[**FM99**] Gerhard Frey and Michael Muller. Arithmetic of modular curves and applications. In *Algorithmic algebra and number theory (Heidelberg, 1997)*, pages 11–48. Springer, Berlin, 1999.

[**KS00**] David R. Kohel and William A. Stein. Component Groups of Quotients of $J_0(N)$. In *Proceedings of the 4th International Symposium (ANTS-IV), Leiden, Netherlands, July 2–7, 2000*, Berlin, 2000. Springer.

[**Maz77**] Barry Mazur. Modular curves and the Eisenstein ideal. *Inst. Hautes Études Sci. Publ. Math.*, (47):33–186 (1978), 1977.

[**Mer93**] Loic Merel. Intersections sur des courbes modulaires. *Manuscripta Math.*, 80(3):283–289, 1993.

[**Mer94**] Loic Merel. Universal Fourier expansions of modular forms. In *On Artin's conjecture for odd 2-dimensional representations*, pages 59–94. Springer, 1994.

[**Mes86**] Jean-Francois Mestre. La méthode des graphes. Exemples et applications. *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata)*, pages 217–242, 1986.

[**PS99**] Bjorn Poonen and Michael Stoll. The Cassels-Tate pairing on polarized abelian varieties. *Ann. of Math. (2)*, 150(3):1109–1149, 1999.

[**Ste00**] William A. Stein. Explicit approaches to modular abelian varieties. *Ph.D. thesis, University of California, Berkeley*, 2000.

[**Ste01**] William A. Stein. Component Groups of Purely Toric Quotients of Semistable Jacobians. *submitted*, 2001.

[**Ste02**] William A. Stein. An introduction to computing modular forms using modular symbols. *will appear in an MSRI Proceedings*, 2002.

[**Stu87**] Jacob Sturm. On the congruence of modular forms. In *Number theory (New York, 1984–1985)*, pages 275–280. Springer, Berlin, 1987.