MODULAR FORMS

0.1 Introduction

This chapter, which was written by William Stein (`was@math.harvard.edu`), with the help of feedback from Kevin Buzzard and Mark Watkins, describes how to compute with modular forms in MAGMA.

0.1.1 Modular Forms

This theoretically-oriented section serves as a guide to the rest of the chapter. We recall the definition of modular forms, then briefly discuss q-expansions, Hecke operators, eigenforms, congruences, and modular symbols.

Fix positive integers N and k, let $\mathbf{H}$ denote the complex upper half plane. Denote by $M_k(\Gamma_1(N))$ the space of modular forms on $\Gamma_1(N)$ of weight k. This is the complex vector space of holomorphic functions $f : \mathbf{H} \to \mathbf{C}$ such that

$$f\left(\frac{az+b}{cz+d}\right) = (cz+d)^k f(z) \qquad \text{for all} \qquad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_1(N),$$

and $f(z)$ vanishes at each element of $¶^1(\mathbf{Q}) = \mathbf{Q} \cup \{\infty\}$ (see, e.g., [DI95] for a more precise definition.)

A *Dirichlet character* is a homomorphism $\varepsilon : (\mathbf{Z}/N\mathbf{Z})^* \to \mathbf{C}^*$ of abelian groups. Dirichlet characters are of interest because they decompose $M_k(\Gamma_1(N))$ into more manageable chunks. If V is any complex vector space equipped with an action $\rho : (\mathbf{Z}/N\mathbf{Z})^* \to \mathrm{Aut}(V)$ and ε is a Dirichlet character, we set

$$V(\varepsilon) = \{x \in V : \rho(a)x = \varepsilon(a)x \quad \text{all} \quad a \in (\mathbf{Z}/N\mathbf{Z})^*\}.$$

The space $M_k(\Gamma_1(N))$ is equipped with an action of $(\mathbf{Z}/N\mathbf{Z})^*$ by the diamond-bracket operators $\langle d \rangle$, which are defined as follows. Given $\overline{d} \in (\mathbf{Z}/N\mathbf{Z})^*$, choose a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(N)$ such that $d \bmod N = \overline{d}$. Then

$$\langle d \rangle f(z) = (cz+d)^{-k} f\left(\frac{az+b}{cz+d}\right).$$

We call $M_k(\Gamma_1(N))(\varepsilon)$ the space of modular forms of weight k, level N, and character ε. This is the complex vector space of holomorphic functions $f : \mathbf{H} \to \mathbf{C}$ such that

$$f\left(\frac{az+b}{cz+d}\right) = \varepsilon(a)(cz+d)^k f(z) \qquad \text{for all} \qquad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(N),$$

and which vanish at the cusps. We let $M_k([\varepsilon])$ denote the direct sum of the spaces $M_k(\Gamma_1(N))(\varepsilon)$ *as* ε *varies over the* $\mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q})$*-conjugates* of ε. It is unnecessary to specify the level because it is built into ε.

To summarize, for any integer k and positive integer N, there is a finite-dimensional $\mathbf{C}$-vector space $M_k(\Gamma_1(N))$. Moreover,

$$M_k(\Gamma_1(N)) = \bigoplus_{\text{all } \varepsilon} M_k(\Gamma_1(N))(\varepsilon) = \bigoplus_{\mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q})\text{-class reps. } \varepsilon} M_k([\varepsilon]).$$

In Section 0.2, we describe how to create the spaces $M_k(\Gamma_1(N))$ and $M_k([\varepsilon])$ in Magma, for any $k \geq 1$, $N \geq 1$, and character ε.

Let f be a modular form, and observe that since $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \in \Gamma_1(N)$, we have $f(z) = f(z+1)$. If we set $q = \exp(2\pi i z)$, there is a *q-expansion* representation for f:

$$f = a_0 + a_1 q + a_2 q^2 + a_3 q^3 + a_4 q^4 + \cdots.$$

The a_n are called the *Fourier coefficients* of f. Magma contains an algorithm for computing a basis of q-expansions for any space of modular forms of weight $k \geq 2$ (see Section 0.3).

Fix a positive integer N and let M be a sum of spaces $M_k(N, \varepsilon)$. Let $q{-}\exp : M \to \mathbf{C}[[q]]$ denote the map that associates to a modular form f its q-expansion. One can prove that there is a basis $f_1, \ldots, f_d$ of M that maps to a basis for the free $\mathbf{Z}$-module $q{-}\exp(M) \cap \mathbf{Z}[[q]]$. See Section 0.4 for how to compute such a basis in Magma. Let $M_{\mathbf{Z}}$ be the $\mathbf{Z}$-module spanned by $f_1, \ldots, f_d$. For any ring R, we define the *space of modular forms over* R to be $M_R = M_{\mathbf{Z}} \otimes_{\mathbf{Z}} R$. Thus M_R is a free R-module of rank d with basis the images of $f_1, \ldots, f_d$ in $M_{\mathbf{Z}} \otimes_{\mathbf{Z}} R$. The computation of M_R is discussed in Section 0.2.2.

Any space M of modular forms is equipped with an action of a commutative ring $\mathbf{T} = \mathbf{Z}[\ldots T_n \ldots]$ of Hecke operators. The computation of Hecke operators T_n and their characteristic polynomials is described in Section 0.9.

An *eigenform* is a simultaneous eigenvector for every element of the Hecke algebra $\mathbf{T}$. A *newform* is an eigenform that doesn't come from a space of lower level and is normalized so that the coefficient of q is 1. Section 0.11 describes how to find newforms. Computation of the mod p reductions and p-adic and complex embeddings of a newform is described in Section 0.12.

Computation of congruences is discussed in Section 0.13.

Modular symbols are closely related to modular forms. See Section 0.16 for the connection between the two.

0.1.2 Status and Future Directions

This is the first version of the modular forms package. I have decided not to delay its release because it contains much useful functionality. However, certain basic algorithms have not yet been fully implemented in MAGMA, though I have a good idea of how to implement them. This includes some operator computations, special values of L-functions, weight one modular forms, and q-expansions at cusps other than ∞. Much closely allied functionality is already available in the modular symbols package. Verbose printing is also currently very unverbose.

0.1.3 Categories

In MAGMA, spaces of modular forms belong to the category `ModFrm`, and the elements of spaces of modular forms belong to `ModFrmElt`.

0.1.4 Verbose Output

To set the verbosity level use the command `SetVerbose("ModularForms",n)`, where `n` is 0 (silent), 1 (verbose), or 2 (very verbose). The default verbose level is 0.

Example H0E1

In this example, we illustrate categories and verbosity for modular forms.

```
> M := ModularForms(11,2); M;
Space of modular forms on Gamma_0(11) of weight 2 and dimension 2 over
Integer Ring.
> Type(M);
ModFrm
> B := Basis(M); B;
[
    1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + 24*q^6 + 24*q^7 + O(q^8),
    q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
]
> Type(B[1]);
ModFrmElt
```

Using `SetVerbose`, we get some information about what is happening during computations.

```
> SetVerbose("ModularForms",2);
> M := ModularForms(30,4);
> EisensteinSubspace(M);
ModularForms:  Computing eisenstein subspace.
ModularForms:  Computing dimension.
Space of modular forms on Gamma_0(30) of weight 4 and dimension 8 over
Integer Ring.
> SetVerbose("ModularForms",0);   // turn off verbose mode
```

0.1.5 An Illustrative Overview

In this section, we give a longer example that serves as an overview of the modular forms package. It illustrates computations of modular forms of level 1, and illustrates the exceptional case of Serre's conjecture with a level 13 example.

Example H0E2

First, we compute the two-dimensional space of modular forms of weight 12 and level 1 over **Z**.

```
> M := ModularForms(Gamma0(1),12); M;
Space of modular forms on Gamma_0(1) of weight 12 and dimension 2 over
Integer Ring.
```

The default output precision is 8:

```
> Basis(M);
[
    1 + 196560*q^2 + 16773120*q^3 + 398034000*q^4 + 4629381120*q^5 +
    34417656000*q^6 + 187489935360*q^7 + O(q^8),
    q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 - 16744*q^7
    + O(q^8)
]
> [PowerSeries(f,10) :  f in Basis(M)];
[
    1 + 196560*q^2 + 16773120*q^3 + 398034000*q^4 + 4629381120*q^5 +
        34417656000*q^6 + 187489935360*q^7 + 814879774800*q^8 +
        2975551488000*q^9 + O(q^10),
    q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 - 16744*q^7
        + 84480*q^8 - 113643*q^9 + O(q^10)
]
> f := Basis(M)[1];
> Coefficient(f,2);
196560
```

The `Newforms` command returns a list of the Galois-orbits of newforms.

```
> NumberOfNewformClasses(M);
2
> f := Newform(M,1); f;
q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 - 16744*q^7 +
O(q^8)
```

We can call the "q" in the q-expansion anything we want and also compute forms to higher precision.

```
> Mf<w> := Parent(f); Mf;
Space of modular forms on Gamma_0(1) of weight 12 and dimension 1 over
Rational Field.
> f + O(w^12);
w - 24*w^2 + 252*w^3 - 1472*w^4 + 4830*w^5 - 6048*w^6 - 16744*w^7 +
    84480*w^8 - 113643*w^9 - 115920*w^10 + 534612*w^11 + O(w^12)
```

Typing `f + O(w^12)` is equivalent to typing `PowerSeries(f,12)`. The second newform orbit contains an Eisenstein series.

```
> E := Newform(M,2); PowerSeries(E,2);
691/65520 + q + O(q^2)
```

Next we compute the two conjugate newforms in $S_2(\Gamma_1(13))$:

```
> M := ModularForms(Gamma1(13),2);
> S := CuspidalSubspace(M); S;
Space of modular forms on Gamma_1(13) of weight 2 and dimension 2 over
Integer Ring.
> NumberOfNewformClasses(S);
1
> f := Newform(S,1); f;
q + (-a - 1)*q^2 + (2*a - 2)*q^3 + a*q^4 + (-2*a + 1)*q^5 + (-2*a +
4)*q^6 + O(q^8)
```

Here a is a root of the polynomial $x^2 - x + 1$.

```
> Parent(f);
Space of modular forms on Gamma_1(13) of weight 2 and dimension 2 over
Number Field with defining polynomial x^2 - x + 1 over the Rational
Field.
```

The Galois-conjugacy class of f has two newforms in it.

```
> Degree(f);
2
> g := Newform(S,1,2); g;
q + (-b - 1)*q^2 + (2*b - 2)*q^3 + b*q^4 + (-2*b + 1)*q^5 + (-2*b +
4)*q^6 + O(q^8)
> BaseRing(Parent(g));
Number Field with defining polynomial x^2 - x + 1 over the Rational
Field
```

The parents of f and g are isomorphic (but distinct) abstract extensions of $\mathbf{Q}$ both isomorphic to $\mathbf{Q}(\sqrt{-3})$.

```
> Parent(f) eq Parent(g);
false
```

We can also list all of the newforms at once, gathered into Galois orbits, using the `Newforms` command:

```
> N := Newforms(M);
> #N;
8
> N[3];
[*
1/13*(-7*zeta_6 - 11) + q + (2*zeta_6 + 1)*q^2 + (-3*zeta_6 + 1)*q^3 +
(6*zeta_6 - 3)*q^4 - 4*q^5 + (-7*zeta_6 + 7)*q^6 + (-7*zeta_6 + 8)*q^7
+ O(q^8),
```

```
1/13*(7*zeta_6 - 18) + q + (-2*zeta_6 + 3)*q^2 + (3*zeta_6 - 2)*q^3 +
(-6*zeta_6 + 3)*q^4 - 4*q^5 + 7*zeta_6*q^6 + (7*zeta_6 + 1)*q^7 +
O(q^8)
*]
```

The “Nebentypus” or character of f has order 6.

```
> e := DirichletCharacter(f); Parent(e);
Group of Dirichlet characters of modulus 13 over Cyclotomic Field of
order 6 and degree 2
> Order(e);
6
```

This shows that there are no cuspidal newforms with Dirichlet character of order 2. As we see below, the mod-3 reduction of f has character $\overline{\varepsilon}$ that takes values in $\mathbf{F}_3$ and has order 2. The minimal lift of $\overline{\varepsilon}$ to characteristic 0 has order 2, while there are no newforms of level 13 with character of order 2. (This example illustrates the “exceptional case of Serre’s conjecture”.)

```
> f3 := Reductions(f,3); f3;
[* [*
q + 2*q^3 + 2*q^4 + O(q^8)
*] *]
> M3<q> := Parent(f3[1][1]);
> f3[1][1]+O(q^15);
q + 2*q^3 + 2*q^4 + q^9 + q^12 + 2*q^13 + O(q^15)
```

The modular forms package can also be used to quickly compute dimensions of spaces of modular forms.

```
> M := ModularForms(Gamma0(1),2048);
> Dimension(M);
171
> M := ModularForms(Gamma1(389),2);
> Dimension(M);
6499
> Dimension(CuspidalSubspace(M));
6112
```

Don’t try to compute a basis of q-expansions for M!

```
> M := ModularForms(Gamma0(123456789),6);
> Dimension(CuspidalSubspace(M));
68624152
> Dimension(EisensteinSubspace(M));
16
```

0.2 Creation Functions

0.2.1 Ambient Spaces

The following are used for creating spaces of modular forms. Each of the following can be replaced by `CuspForms` to obtain the subspace of cusp forms instead.

`ModularForms(N)`

The space $M_2(\Gamma_0(N), \mathbf{Z})$ of modular forms on $\Gamma_0(N)$ of weight 2. See the documentation for `ModularForms(N,k)` below, with $k = 2$.

`ModularForms(N, k)`

The space $M_k(\Gamma_0(N), \mathbf{Z})$ of weight k modular forms on $\Gamma_0(N)$ over $\mathbf{Z}$.

`ModularForms(chars, k)`

The space $M_k(\text{chars})$, which is the direct sum of the spaces $M_k(\Gamma_1(N))(e)$ as e runs over all characters Galois-conjugate to some character in `chars`.

`ModularForms(G)`

This is the same as `ModularForms(G,2)` (see below).

`ModularForms(G, k)`

The space $M_k(G, \mathbf{Z})$, where G is a congruence subgroup. The groups $\Gamma_0(N)$ and $\Gamma_1(N)$ are currently supported, and can be created using the commands `Gamma0(N)` and `Gamma1(N)`, respectively.

Example H0E3

In this example, we illustrate each of the above constructors in turn. First we create $M_2(\Gamma_0(65))$.

```
> M := ModularForms(65); M;
Space of modular forms on Gamma_0(65) of weight 2 and dimension 8 over
Integer Ring.
> Dimension(M);
8
> Basis(CuspidalSubspace(M));
[
    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)
]
```

Next we create $M_4(\Gamma_0(8))$.

```
> M := ModularForms(8,4); M;
Space of modular forms on Gamma_0(8) of weight 4 and dimension 5 over
Integer Ring.
```

```
> Dimension(M);
5
> Basis(CuspidalSubspace(M));
[
    q - 4*q^3 - 2*q^5 + 24*q^7 + O(q^8)
]
```

Now we create the space $M_3(N, \varepsilon)$, where ε is a character of level 20, conductor 5 and order 4.

```
> G := DirichletGroup(20,CyclotomicField(EulerPhi(20)));
> chars := Elements(G); #chars;
8
> [Conductor(eps) :  eps in chars];
[ 1, 4, 5, 20, 5, 20, 5, 20 ]
> eps := chars[3];
> IsEven(eps);
false
> M := ModularForms([eps],3); M;
Space of modular forms on Gamma_1(20) with character all conjugates of
[$.2], weight 3, and dimension 12 over Integer Ring.
> Dimension(EisensteinSubspace(M));
6
> Dimension(CuspidalSubspace(M));
6
```

Next we create the direct sum of the spaces $M_k(20, \varepsilon)$ as ε varies over the four mod 20 characters of order at most 2, for $k = 2$ and 3.

```
> G := DirichletGroup(20, RationalField());   // (Z/20Z)^* --> Q^*
> chars := Elements(G); #chars;
4
> M := ModularForms(chars,2); M;
Space of modular forms on Gamma_1(20) with characters all
conjugates of [1, .1,.2, .1*.2], weight 2, and dimension 12
over Integer Ring.
> M := ModularForms(chars,3); M;
Space of modular forms on Gamma_1(20) with characters all
conjugates of [1, .1,.2, .1*.2], weight 3, and dimension 16
over Integer Ring.
```

Now we create the spaces $M_k(\Gamma_1(20))$ for $k = 2, 3$.

```
> ModularForms(Gamma1(20));
Space of modular forms on Gamma_1(20) of weight 2 and dimension 22
over Integer Ring.
> ModularForms(Gamma1(20),3);
Space of modular forms on Gamma_1(20) of weight 3 and dimension 34
over Integer Ring.
```

We can also create the subspace of cuspforms directly:

```
> CuspForms(Gamma1(20));
```

```
Space of modular forms on Gamma_1(20) of weight 2 and dimension 3
over Integer Ring.
> CuspForms(Gamma1(20),3);
Space of modular forms on Gamma_1(20) of weight 3 and dimension 14
over Integer Ring.
```

0.2.2 Base Extension

If M is a space of modular formS created using one of the constructors in Section 0.2.1, then the base ring of M is $\mathbf{Z}$. Thus we can base extend M to any ring R. The examples below illustrate some simple applications of `BaseExtend`.

`BaseExtend(M, R)`

The base extension of the space M of modular forms to the ring R and the induced map from M to `BaseExtend(M,R)`. The only requirement on R is that there is a natural coercion map from the base ring of M to R. For example, when `BaseRing(M)` is the integers, any ring R is allowed.

`BaseExtend(M, phi)`

The base extension of the space M of modular forms to the ring R using the map ϕ : `BaseRing(M)` $\rightarrow R$, and the induced map from M to `BaseExtend(M,R)`

Example H0E4

We first illustrate an Eisenstein series in $M_{12}(1)$ that is congruent to 1 modulo 3.

```
> M<q> := EisensteinSubspace(ModularForms(1,12));
> E12 := M.1; E12 + O(q^4);
691 + 65520*q + 134250480*q^2 + 11606736960*q^3 + O(q^4)
> M3<q3> := BaseExtend(M,GF(3));
> Dimension(M3);
1
> M3.1+O(q3^20);
1 + O(q3^20)
```

This congruence can be proved by noting that the coefficient of q^n in the q-expansion of $E_{12}/65520$, for any $n \geq 1$, is an eigenvalue of a Hecke operator, hence an integer, and that 65520 is divisible by 3. Because E_{12} is defined over $\mathbf{Z}$ the command `"E12Q/65520"` would result in an error, so we first base extend to $\mathbf{Q}$.

```
> MQ, phi := BaseExtend(M,RationalField());
> E12Q := phi(E12);
> E12Q/65520;
691/65520 + q + 2049*q^2 + 177148*q^3 + 4196353*q^4 + 48828126*q^5 +
362976252*q^6 + 1977326744*q^7 + O(q^8)
```

It is possible to base extend to almost any silly commutative ring.

```
> M := ModularForms(11,2);
```

```
> R := PolynomialRing(GF(17),3);
> MR<q> := BaseExtend(M,R); MR;
Space of modular forms on Gamma_0(11) of weight 2 and dimension 2 over
Polynomial ring of rank 3 over GF(17)
Lexicographical Order
Variables:  $.1, $.2, $.3.
> f := MR.1;  f + O(q^5);
1 + 12*q^2 + 12*q^3 + 12*q^4 + O(q^5)
> f*(R.1+3*R.2) + O(q^4);
$.1 + 3*$.2 + (12*$.1 + 2*$.2)*q^2 + (12*$.1 + 2*$.2)*q^3 + O(q^4)
```

0.2.3 Elements

`M . i`

The ith basis vector of M.

`M ! f`

The coercion of f into M. Here f can be a modular form, a power series with absolute precision, or something that can be coerced into `RSpace(M)`.

`ModularForm(E)`

The modular form associated to the elliptic curve E over **Q**. (See Section 0.15.)

Example H0E5

```
> M := ModularForms(Gamma0(11),2);
> M.1;
1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + 24*q^6 + 24*q^7 + O(q^8)
> M.2;
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
> R<q> := PowerSeriesRing(Integers());
> f := M!(1 + q + 10*q^2 + O(q^3));
> f;
1 + q + 10*q^2 + 11*q^3 + 14*q^4 + 13*q^5 + 26*q^6 + 22*q^7 + O(q^8)
> Eltseq(f);
[ 1, 1 ]
```

`Eltseq` gives f as a linear combination of `M.1` and `M.2`. Next we coerce f into $M_2(\Gamma_0(22))$.

```
> M22 := ModularForms(Gamma0(22),2);
> g := M22!f; g;
1 + q + 10*q^2 + 11*q^3 + 14*q^4 + 13*q^5 + 26*q^6 + 22*q^7 + O(q^8)
> Eltseq(g);
[ 1, 1, 10, 11, 14 ]
```

The elliptic curve E below defines an element of $M_2(\Gamma_0(11))$.

```
> E := EllipticCurve([ 0, -1, 1, -10, -20 ]);
```

```
> Conductor(E);
11
> f := ModularForm(E);
> f;
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
> f + M.1;
1 + q + 10*q^2 + 11*q^3 + 14*q^4 + 13*q^5 + 26*q^6 + 22*q^7 + O(q^8)
```

Note, however, that the ambient space of the parent of f is not equal to M, despite the fact that both are isomorphic to $M_2(\Gamma_0(11))$. This is because they were created independently. The above operations are defined because there is a canonical way to coerce f into M using its q-expansion.

```
> f in M;
false
> AmbientSpace(Parent(f)) eq M;
false
```

0.3 Bases

`Basis(M)`

The canonical basis of M. Any space of modular forms that can be created in MAGMA is of the form $M_{\mathbf{Z}} \otimes_{\mathbf{Z}} R$ for some ring R and some space $M_{\mathbf{Z}}$ of modular forms defined over $\mathbf{Z}$. The basis of M is the image in M of `Basis(M_Z)`, and `Basis(M_Z)` is in Hermite normal form.

Example H0E6

```
> M := ModularForms(Gamma1(16),3); M;
Space of modular forms on Gamma_1(16) of weight 3 and dimension 23
over Integer Ring.
> Dimension(CuspidalSubspace(M));
9
> SetPrecision(M,19);
> Basis(NewSubspace(CuspidalSubspace(M)))[1];
q - 76*q^8 + 39*q^9 + 132*q^10 - 44*q^11 + 84*q^12 - 144*q^13 -
232*q^14 + 120*q^15 + 160*q^16 + 158*q^17 - 76*q^18 + O(q^19)
```

We can print the whole basis to less precision:

```
> SetPrecision(M,10);
> Basis(NewSubspace(CuspidalSubspace(M)));
[
q - 76*q^8 + 39*q^9 + O(q^10),
q^2 + q^7 - 58*q^8 + 30*q^9 + O(q^10),
q^3 + 2*q^7 - 42*q^8 + 18*q^9 + O(q^10),
q^4 + q^7 - 26*q^8 + 13*q^9 + O(q^10),
q^5 + 2*q^7 - 18*q^8 + 5*q^9 + O(q^10),
```

```
q^6 + 2*q^7 - 12*q^8 + 3*q^9 + O(q^10),
3*q^7 - 8*q^8 + O(q^10)
]
```

Note the coefficient 3 of q^7, which emphasizes that this is not the reduced echelon form over a field, but the image of a reduced form over the integers:

```
> MQ := BaseExtend(M,RationalField());
> Basis(NewSubspace(CuspidalSubspace(MQ)))[7];
3*q^7 - 8*q^8 + O(q^10)
```

0.4 q-Expansions

The following intrinsics give the q-expansion of a modular form (about the cusp ∞).

`qExpansion(f)`

`qExpansion(f, prec)`

`PowerSeries(f)`

`PowerSeries(f, prec)`

The q-expansion (at the cusp ∞) of the modular form f to absolute precision `prec`. This is an element of the power series ring over the base ring of the parent of f.

`Coefficient(f, n)`

The nth coefficient of the q-expansion of f.

`Precision(M)`

The default printing precision for elements of the space M of modular forms. This is the precision that is used when printing elements of M if the user does not specifically make a choice. The hard-coded default value is 8.

`SetPrecision(M, prec)`

Set the default printing precision for elements of the space M of modular forms.

`PrecisionBound(M : `*parameters*`)`

`Exact` BoolElt *Default : false*

An integer b such that $f + O(q^b)$ determines any modular form f in M. If the optional parameter `Exact` is set to true, then this intrinsic returns the *smallest* integer b such that $f + O(q^b)$ determines any modular form f in M. **WARNING:** In previous versions of MAGMA the default was `Exact := true`.

Example H0E7

In this example, we compute the q-expansion of a modular form $f \in M_3(\Gamma_1(11))$ in several ways.

```
> M := ModularForms(Gamma1(11),3); M;
Space of modular forms on Gamma_1(11) of weight 3 and dimension 15
over Integer Ring.
> f := M.1;
> f;
1 + O(q^8)
> qExpansion(f);
1 + O(q^8)
> Coefficient(f,16);  // f is a modular form, so has infinite precision
-5457936
> qExpansion(f,17);
1 + 763774*q^15 - 5457936*q^16 + O(q^17)
> PowerSeries(f,20);   // same as qExpansion(f,20)
1 + 763774*q^15 - 5457936*q^16 + 14709156*q^17 - 12391258*q^18 -
21614340*q^19 + O(q^20)
```

The "big-oh" notation is supported via addition of a modular form and a power series.

```
> M<q> := Parent(f);
> Parent(q);
Power series ring in q over Integer Ring
> f + O(q^17);
1 + 763774*q^15 - 5457936*q^16 + O(q^17)
> 5*q - O(q^17) + f;
1 + 5*q + 763774*q^15 - 5457936*q^16 + O(q^17)
> 5*q + f;
1 + 5*q + O(q^8)
```

Default printing precision can be set using the command `SetPrecision`.

```
> SetPrecision(M,16);
> f;
1 + 763774*q^15 + O(q^16)
```

Example H0E8

The PrecisionBound intrinsic is related to Weierstrass points on modular curves. Let N be a positive integer such that $S = S_2(\Gamma_0(N))$ has dimension at least 2. Then the point ∞ is a Weierstrass point on $X_0(N)$ if and only if `PrecisionBound(S : Exact := true)-1 ne Dimension(S)`.

```
> function InftyIsWP(N)
>    S := CuspidalSubspace(ModularForms(Gamma0(N),2));
>    assert Dimension(S) ge 2;
>    return (PrecisionBound(S : Exact := true)-1) ne Dimension(S);
> end function;
> [<N,InftyIsWP(N)> :  N in [97..100]];
[ <97, false>, <98, true>, <99, false>, <100, true> ]
```

It is an open problem to give a simple characterization of the integers N such that ∞ is a Weierstrass point on $X_0(N)$, though Atkin and others have made significant progress on this problem (see, e.g., 1967 Annals paper [Atk67]). I verified that if $N < 3223$ is square free, then ∞ is not a Weierstrass point on $X_0(N)$, which suggests a nice conjecture.

0.5 Arithmetic

`f + g`

The sum of the modular forms f and g.

`f + g`

The sum of the modular form f and the power series g. The q-expansion of f must be coercible into the parent of g. The sum $g+f$ is also defined, as are the differences $f-g$ and $g-f$.

`f - g`

The difference of the modular forms f and g.

`a * f`

The product of the scalar a and the modular form f.

`f / a`

The product of the scalar $1/a$ and the modular form f.

`f ^ n`

The power f^n of the modular form f, where $n \geq 1$ is an integer.

`f * g`

The product of the modular forms f and g. The only condition is that the base fields of f and g be the same. The weight of `f*g` is the sum of the weights of f and g.

Example H0E9

```
> M2 := ModularForms(Gamma0(11),2);
> f := M2.1;
> g := M2.2;
> f;
1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + 24*q^6 + 24*q^7 + O(q^8)
> g;
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
> f+g;
1 + q + 10*q^2 + 11*q^3 + 14*q^4 + 13*q^5 + 26*q^6 + 22*q^7 + O(q^8)
> 2*f;
2 + 24*q^2 + 24*q^3 + 24*q^4 + 24*q^5 + 48*q^6 + 48*q^7 + O(q^8)
> MQ,phi := BaseExtend(M2,RationalField());
```

```
> phi(2*f)/2;
1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + 24*q^6 + 24*q^7 + O(q^8)
> f^2;
1 + 24*q^2 + 24*q^3 + 168*q^4 + 312*q^5 + 480*q^6 + 624*q^7 + O(q^8)
> Parent($1);
Space of modular forms on Gamma_0(11) of weight 4 and dimension 4 over
Integer Ring.
> M3 := ModularForms([DirichletGroup(11).1],3); M3;
Space of modular forms on Gamma_1(11) with character all conjugates of
[$.1], weight 3, and dimension 3 over Integer Ring.
> M3.1*f;
1 + 12*q^2 + 2*q^3 + 6*q^4 - 126*q^5 - 168*q^6 - 384*q^7 + O(q^8)
> Parent($1);
Space of modular forms on Gamma_1(11) of weight 5 and dimension 25
over Integer Ring.
```

0.6 Predicates

`IsAmbientSpace(M)`

True if and only if M is an ambient space. Ambient spaces are those space constructed in Section 0.2.1.

`IsCuspidal(M)`

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

`IsEisenstein(M)`

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

`IsEisensteinSeries(f)`

True if f is an Eisenstein newform or was computed using the `EisensteinSeries` intrinsic. (See Section 0.10.)

`IsGamma0(M)`

True if M is a space of modular forms for $\Gamma_0(N)$.

`IsGamma1(M)`

True if M is a space of modular forms for $\Gamma_1(N)$. (For efficiency purposes, if you create a space using the `ModularForms(chars,k)` constructor, and `chars` consists of all mod N Dirichlet characters, then `IsGamma1` will still be false.)

`IsNew(M)`

True if M is contained in the new subspace of the ambient space.

`IsNewform(f)`

True if f was created using `Newforms`. (Sometimes true in other cases in which f is obviously a newform. In number theory, "newform" means "normalized eigenform that lies in the new subspace".)

`IsRingOfAllModularForms(M)`

True if and only if M is the ring of all modular forms over a given ring.

Example H0E10

We illustrate each of the above predicates with some simple computations in $M_3(\Gamma_1(11))$.

```
> M := ModularForms(Gamma1(11),3);
> f := Newform(M,1);
> IsAmbientSpace(M);
true
> IsAmbientSpace(CuspidalSubspace(M));
false
> IsCuspidal(M);
false
> IsCuspidal(CuspidalSubspace(M));
true
> IsEisenstein(CuspidalSubspace(M));
false
> IsEisenstein(EisensteinSubspace(M));
true
> IsGamma1(M);
true
> IsNew(M);
true
> IsNewform(M.1);
false
> IsNewform(f);
true
> IsRingOfAllModularForms(M);
false
> Level(f);
11
> Level(M);
11
> Weight(f);
3
> Weight(M);
3
> Weight(M.1);
3
```

0.7 Properties

`Degree(f)`

The number of Galois-conjugates of the modular form f over the prime subfield of (the fraction field of) the base ring of f.

`Dimension(M)`

The dimension of the space M of modular forms.

`DirichletCharacters(M)`

A sequence whose elements give one representative from each Galois-conjugacy class of Dirichlet characters associated to M.

`Eltseq(f)`

The sequence $[a_1, \ldots, a_n]$ such that $f = a_1 g_1 + \cdots + a_n g_n$, where $g_1, \ldots, g_n$ is the basis of the parent of f.

`Level(f)`

The level of f.

`Level(M)`

The level of M.

`Weight(f)`

The weight of the modular form f, if it is defined.

`Weight(M)`

The weight of the space M of modular forms.

Example H0E11

We illustrate each of the above properties with some simple computations in $M_3(\Gamma_1(11))$.

```
> M := ModularForms(Gamma1(11),3);
> Degree(M.1);
1
> f := Newform(M,1);
> Degree(f);
4
> Dimension(M);
15
> DirichletCharacters(M);
[
1,
$.1,
$.1^2,
$.1^5
```

```
]
> Level(f);
11
> Level(M);
11
> Weight(f);
3
> Weight(M);
3
> Weight(M.1);
3
```

0.8 Subspaces

The following functions compute the cuspidal, Eisenstein, and new subspaces.

`CuspidalSubspace(M)`

The subspace of forms f in M such that the constant term of the Fourier expansion of f at every cusp is 0.

`EisensteinSubspace(M)`

The Eisenstein subspace of M.

`NewSubspace(M)`

The new subspace of M.

`ZeroSubspace(M)`

The trivial subspace of M.

Example H0E12

We compute a basis of q-expansions for each of the above subspace of $M_2(\Gamma_0(33))$.

```
> M := ModularForms(Gamma0(33),2); M;
Space of modular forms on Gamma_0(33) of weight 2 and dimension 6 over
Integer Ring.
> Basis(M);
[
    1 + O(q^8),
    q - q^5 + 2*q^7 + O(q^8),
    q^2 + 2*q^7 + O(q^8),
    q^3 + O(q^8),
    q^4 + q^5 + O(q^8),
    q^6 + O(q^8)
]
> Basis(CuspidalSubspace(M));
```

```
[
    q - q^5 - 2*q^6 + 2*q^7 + O(q^8),
    q^2 - q^4 - q^5 - q^6 + 2*q^7 + O(q^8),
    q^3 - 2*q^6 + O(q^8)
]
> Basis(EisensteinSubspace(M));
[
    1 + O(q^8),
    q + 3*q^2 + 7*q^4 + 6*q^5 + 8*q^7 + O(q^8),
    q^3 + 3*q^6 + O(q^8)
]
> Basis(NewSubspace(M));
[
    q + q^2 - q^3 - q^4 - 2*q^5 - q^6 + 4*q^7 + O(q^8)
]
> Basis(NewSubspace(EisensteinSubspace(M)));
[]
> Basis(NewSubspace(CuspidalSubspace(M)));
[
    q + q^2 - q^3 - q^4 - 2*q^5 - q^6 + 4*q^7 + O(q^8)
]
> ZeroSubspace(M);
Space of modular forms on Gamma_0(33) of weight 2 and dimension 0 over
Integer Ring.
```

0.9 Operators

Each space M of modular forms comes equipped with a commuting family $T_1, T_2, T_3, \ldots$ of linear operators acting on it called the *Hecke operators*. Unfortunately, at present, the computation of Hecke and other operators on spaces of modular forms with nontrivial character has not yet been implemented, though computation of characteristic polynomials of Hecke operators is supported.

`HeckeOperator(M, n)`

The matrix representing the nth Hecke operator T_n with respect to `Basis(M)`. (Currently M must be a space of modular forms with trivial character and integral weight ≥ 2.)

`HeckePolynomial(M, n : ` *parameters*`)`

`Proof` BoolElt *Default : true*

The characteristic polynomial of the nth Hecke operator T_n. In some situations this is more efficient than `CharacteristicPolynomial(HeckeOperator(M,n))` or any of its variants. Note that M can be arbitrary.

`AtkinLehnerOperator(M, q)`

The matrix representing the qth Atkin-Lehner involution W_q on M with respect to `Basis(M)`. (Currently M must be a cuspidal space of modular forms with trivial character and integral weight ≥ 2.)

Example H0E13

First we compute a characteristic polynomial on $S_2(\Gamma_1(13))$ over both $\mathbf{Z}$ and the finite field $\mathbf{F}_2$.

```
> R<x> := PolynomialRing(Integers());
> S := CuspForms(Gamma1(13),2);
> HeckePolynomial(S,2);
x^2 + 3*x + 3
> S2 := BaseExtend(S,GF(2));
> R<y> := PolynomialRing(GF(2));
> Factorization(HeckePolynomial(S2,2));
[
<y^2 + y + 1, 1>
]
```

Next we compute a Hecke operator on $M_4(\Gamma_0(14))$.

```
> M := ModularForms(Gamma0(14),4);
> T := HeckeOperator(M,2);
> T;
[  1   0   0   0   0   0   0 240]
[  0   0   0   0  18  12  50 100]
[  0   1   0   0  -2  18  12 -11]
[  0   0   0   0   1  22  25  46]
[  0   0   1   0  -1 -16 -20 -82]
[  0   0   0   0  -1  -6  -9 -38]
[  0   0   0   1   3   9  15  39]
[  0   0   0   0   0   0   0   8]
> Parent(T);
Full Matrix Algebra of degree 8 over Integer Ring
> Factorization(CharacteristicPolynomial(T));
[
<x - 8, 2>,
<x - 2, 1>,
<x - 1, 2>,
<x + 2, 1>,
<x^2 + x + 8, 1>
]
> f := M.1;
> f*T;
1 + 240*q^7 + O(q^8)
> M.1 + 240*M.8;
1 + 240*q^7 + O(q^8)
```

This example demonstrates the Atkin-Lehner involution W_3 on $S_2(\Gamma_0(33))$.

```
> M := ModularForms(33,2);
```

```
> S := CuspidalSubspace(M);
> W3 := AtkinLehnerOperator(S,3);
> W3;
[   1    0    0]
[ 1/3  1/3 -4/3]
[ 1/3 -2/3 -1/3]
> Factorization(CharacteristicPolynomial(W3));
[
<x - 1, 2>,
<x + 1, 1>
]
> f := S.2;
> f*W3;
1/3*q + 1/3*q^2 - 4/3*q^3 - 1/3*q^4 - 2/3*q^5 + 5/3*q^6
     + 4/3*q^7 + O(q^8)
```

The Atkin-Lehner and Hecke operators need not commute:

```
> T3 := HeckeOperator(S,3);
> T3;
[ 0 -2 -1]
[ 0 -1  1]
[ 1 -2 -1]
> T3*W3 - W3*T3 eq 0;
false
```

0.10 Eisenstein Series

The intrinsics below require that the base ring of M has characteristic 0. To compute mod p eigenforms, use the `Reduction` intrinsic (see Section 0.12).

`EisensteinSeries(M)`

List of the Eisenstein series associated to the modular forms space M. By "associated to" we mean that the Eisenstein series lies in $M\otimes\mathbf{C}$.

`IsEisensteinSeries(f)`

True if f was created using `EisensteinSeries`.

`EisensteinData(f)`

The data $<\chi,\psi,t,\chi',\psi'>$ that defines the Eisenstein series f. Here χ is a primitive character of conductor S, ψ is primitive of conductor M, and MSt divides N, where N is the level of f. (The additional characters χ' and ψ' are equal to χ and ψ respectively, except they take values in the big field $\mathbf{Q}(\zeta_{\varphi(N)})^*$ instead of $\mathbf{Q}(\zeta_n)^*$, where n is the order of χ or ψ.) The Eisenstein series associated to (χ, ψ, t) has q-expansion

$$c_0 + \sum_{m \geq 1} \left(\sum_{n|m} \psi(n) n^{k-1} \chi(m/n) \right) q^{mt},$$

where $c_0 = 0$ if $S > 1$ and $c_0 = L(1-k, \psi)/2$ if $S = 1$.

Example H0E14

We illustrate the above intrinsics by computing the Eisenstein series in $M_3(\Gamma_1(12))$.

```
> M := ModularForms(Gamma1(12),3); M;
Space of modular forms on Gamma_1(12) of weight 3 and dimension 13
over Integer Ring.
> E := EisensteinSubspace(M); E;
Space of modular forms on Gamma_1(12) of weight 3 and dimension 10
over Integer Ring.
> s := EisensteinSeries(E); s;
[*
-1/9 + q - 3*q^2 + q^3 + 13*q^4 - 24*q^5 - 3*q^6 + 50*q^7 + O(q^8),
-1/9 + q^2 - 3*q^4 + q^6 + O(q^8),
-1/9 + q^4 + O(q^8),
-1/4 + q + q^2 - 8*q^3 + q^4 + 26*q^5 - 8*q^6 - 48*q^7 + O(q^8),
-1/4 + q^3 + q^6 + O(q^8),
q + 3*q^2 + 9*q^3 + 13*q^4 + 24*q^5 + 27*q^6 + 50*q^7 + O(q^8),
q^2 + 3*q^4 + 9*q^6 + O(q^8),
q^4 + O(q^8),
q + 4*q^2 + 8*q^3 + 16*q^4 + 26*q^5 + 32*q^6 + 48*q^7 + O(q^8),
q^3 + 4*q^6 + O(q^8)
*]
> a := EisensteinData(s[1]); a;
<1, $.1, 1, 1, $.2>
> Parent(a[2]);
Group of Dirichlet characters of modulus 3 over Rational Field
> Order(a[2]);
2
> Parent(a[5]);
Group of Dirichlet characters of modulus 12 over Cyclotomic Field of
order 4 and degree 2
> Parent(s[1]);
Space of modular forms on Gamma_1(12) of weight 3 and dimension 10
over Rational Field.
> IsEisensteinSeries(s[1]);
true
```

0.11 Newforms

In this section we describe how to compute both cuspidal and Eisenstein newforms.

The intrinsics below require that the base ring of M has characteristic 0. To compute mod p eigenforms, use the `Reduction` intrinsic (see Section 0.12).

`NumberOfNewformClasses(M : parameters)`

`Proof` BoolElt *Default : true*

The number of Galois conjugacy-classes of newforms associate to the modular forms space M, which must have base ring **Z** or **Q**. By "associated to" we mean that the newform lies in $M \otimes \mathbf{C}$.

`Newform(M, i, j : parameters)`

`Proof` BoolElt *Default : true*

The jth Galois-conjugate newform in the ith Galois-orbit of newforms, which must have base ring **Z** or **Q**.

`Newform(M, i : parameters)`

`Proof` BoolElt *Default : true*

The first Galois-conjugate newform in the ith orbit, which must have base ring **Z** or **Q**.

`Newforms(M : parameters)`

`Proof` BoolElt *Default : true*

Sort list of the newforms associated to M divided up into Galois orbits.

`Newforms(I, M)`

Use this intrinsic to find the newforms associated to M with prespecified eigenvalues. Here I is a sequence $[\langle p_1, f_1(x)\rangle, ..., \langle p_n, f_n(x)\rangle]$ of pairs. Each pair consists of a prime number that does not divide the level of M and a polynomial. This intrinsic returns the set of newforms $\sum a_n q^n$ in M such that $f_n(a_{p_n}) = 0$. (This intrinsic only works when M is cuspidal and defined over **Q** or **Z**.)

Example H0E15

We compute the newforms in $M_5(\Gamma_1(8))$.

```
> M := ModularForms(Gamma1(8),5); M;
Space of modular forms on Gamma_1(8) of weight 5 and dimension 11 over
Integer Ring.
> NumberOfNewformClasses(M);
4
> Newforms(M);
[* [*
q + 4*q^2 - 14*q^3 + 16*q^4 - 56*q^6 + O(q^8)
*], [*
```

```
q + 1/24*(a - 30)*q^2 + 6*q^3 + 1/12*(-a - 162)*q^4 + 1/3*(-a + 6)*q^5
+ 1/4*(a - 30)*q^6 + 1/3*(2*a - 12)*q^7 + O(q^8),
q + 1/24*(b - 30)*q^2 + 6*q^3 + 1/12*(-b - 162)*q^4 + 1/3*(-b + 6)*q^5
+ 1/4*(b - 30)*q^6 + 1/3*(2*b - 12)*q^7 + O(q^8)
*], [*
57/2 + q + q^2 + 82*q^3 + q^4 - 624*q^5 + 82*q^6 - 2400*q^7 + O(q^8)
*], [*
q + 16*q^2 + 82*q^3 + 256*q^4 + 624*q^5 + 1312*q^6 + 2400*q^7 + O(q^8)
*] *]
> Newform(M,1);
q + 4*q^2 - 14*q^3 + 16*q^4 - 56*q^6 + O(q^8)
> Newform(M,2);
q + 1/24*(a - 30)*q^2 + 6*q^3 + 1/12*(-a - 162)*q^4 + 1/3*(-a + 6)*q^5
+ 1/4*(a - 30)*q^6 + 1/3*(2*a - 12)*q^7 + O(q^8)
> Parent(Newform(M,2));
Space of modular forms on Gamma_1(8) of weight 5 and dimension 2 over
Number Field with defining polynomial x^2 - 12*x + 8676 over the
Rational Field.
> Newform(M,2,2);
q + 1/24*(b - 30)*q^2 + 6*q^3 + 1/12*(-b - 162)*q^4 + 1/3*(-b + 6)*q^5
+ 1/4*(b - 30)*q^6 + 1/3*(2*b - 12)*q^7 + O(q^8)
> IsEisensteinSeries(Newform(M,1));
false
> IsEisensteinSeries(Newform(M,2));
false
> IsEisensteinSeries(Newform(M,3));
true
> IsEisensteinSeries(Newform(M,4));
true
```

The following example demonstrates picking out a newform in $S_2(\Gamma_0(65))$ with prespecified eigenvalues.

```
> S := CuspForms(65,2);
> R<x> := PolynomialRing(IntegerRing());
> I := [<3,x+2>];
> Newforms(I,S);
[* [*
q - q^2 - 2*q^3 - q^4 - q^5 + 2*q^6 - 4*q^7 + O(q^8)
*] *]
> Factorization(HeckePolynomial(S,2));
[
<x + 1, 1>,
<x^2 - 3, 1>,
<x^2 + 2*x - 1, 1>
]
> I := [<2,x^2-3>];
> Newforms(I,S);
[* [*
```

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

0.11.1 Labels

It is possible to obtain the galois-conjugacy class of a newform by giving a descriptive label as an argument to `Newforms`. The format of the label is as follows:

`[G0N or G1N][Level]k[Weight][Isogeny Class]`.

Some example labels are `"G0N11k2A"`, `"G0N1k12A"`, `"G1N17k2B"`, and `"G1N9k3B"`. If the string `"G0N"` or `"G1N"` is omitted, then the default is `"G0N"`. Thus the following are also valid: `"11k2A"`, `"1k12A"`, `"37k4A"`. If `k[Weight]` is omitted, then the default is weight 2, so the following are valid and all refer to weight 2 modular forms on some $\Gamma_0(N)$: `"11A"`, `"37A"`, `"65B"`. In order, possibilities for the isogeny class are as follows:

`A, B, C, ..., Y, Z, AA, BB, CC, ..., ZZ, AAA, BBB, CCC, ....`

This is essentially the notation used in [Cre97] for isogeny classes, though sometimes for levels ≤ 450 the ordering differs from that in [Cre97].

Suppose s is a valid label, and let M be the space of modular forms that contains `ModularForm(s)`. Then `ModularForm(s)` is by definition `Newforms(M)[i]` where the isogeny class in the label s is the ith isogeny class. For example `C` corresponds to the 3rd isogeny class and `BB` corresponds to the 28th.

`Newforms(label)`

The Galois-conjugacy class(es) of newforms described by the label. See the handbook for a description of the notation used for the label.

Example H0E16

We give many examples of constructing newforms using labels.

```
> Newforms("11A");
[*
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
*]
>  Newforms("G0N11k2A");
[*
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
*]
>  Newforms("G0N1k12A");
[*
q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 - 16744*q^7 +
O(q^8)
*]
>  Newforms("G1N17k2B");
[*
q + (-a^3 + a^2 - 1)*q^2 + (a^3 - a^2 - a - 1)*q^3 + (2*a^3 - a^2 +
```

```
2*a)*q^4 + (-a^3 - a^2)*q^5 + (-a^3 + a^2 - a + 1)*q^6 + (-a^3 + a^2 +
a - 1)*q^7 + O(q^8),
q + (-b^3 + b^2 - 1)*q^2 + (b^3 - b^2 - b - 1)*q^3 + (2*b^3 - b^2 +
2*b)*q^4 + (-b^3 - b^2)*q^5 + (-b^3 + b^2 - b + 1)*q^6 + (-b^3 + b^2 +
b - 1)*q^7 + O(q^8),
q + (-c^3 + c^2 - 1)*q^2 + (c^3 - c^2 - c - 1)*q^3 + (2*c^3 - c^2 +
2*c)*q^4 + (-c^3 - c^2)*q^5 + (-c^3 + c^2 - c + 1)*q^6 + (-c^3 + c^2 +
c - 1)*q^7 + O(q^8),
q + (-d^3 + d^2 - 1)*q^2 + (d^3 - d^2 - d - 1)*q^3 + (2*d^3 - d^2 +
2*d)*q^4 + (-d^3 - d^2)*q^5 + (-d^3 + d^2 - d + 1)*q^6 + (-d^3 + d^2 +
d - 1)*q^7 + O(q^8)
*]
> Newforms("G1N9k3B");
[*
1/3*(-5*zeta_6 - 2) + q + (4*zeta_6 + 1)*q^2 + q^3 + (20*zeta_6 -
15)*q^4 + (-25*zeta_6 + 26)*q^5 + (4*zeta_6 + 1)*q^6 + (-49*zeta_6 +
1)*q^7 + O(q^8),
1/3*(5*zeta_6 - 7) + q + (-4*zeta_6 + 5)*q^2 + q^3 + (-20*zeta_6 +
5)*q^4 + (25*zeta_6 + 1)*q^5 + (-4*zeta_6 + 5)*q^6 + (49*zeta_6 -
48)*q^7 + O(q^8)
*]
> Newforms("11k2A");
[*
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
*]
> Newforms("11A");
[*
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 + O(q^8)
*]
> Newforms("1k12A");
[*
q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5 - 6048*q^6 - 16744*q^7 +
O(q^8)
*]
> Newforms("37k4A");
[*
q + a*q^2 + 1/8*(-a^3 - 9*a^2 - 26*a - 22)*q^3 + (a^2 - 8)*q^4 +
1/8*(13*a^3 + 85*a^2 + 50*a - 186)*q^5 + 1/8*(-3*a^3 - 27*a^2 - 38*a +
6)*q^6 + 1/4*(-19*a^3 - 119*a^2 - 30*a + 170)*q^7 + O(q^8),
q + b*q^2 + 1/8*(-b^3 - 9*b^2 - 26*b - 22)*q^3 + (b^2 - 8)*q^4 +
1/8*(13*b^3 + 85*b^2 + 50*b - 186)*q^5 + 1/8*(-3*b^3 - 27*b^2 - 38*b +
6)*q^6 + 1/4*(-19*b^3 - 119*b^2 - 30*b + 170)*q^7 + O(q^8),
q + c*q^2 + 1/8*(-c^3 - 9*c^2 - 26*c - 22)*q^3 + (c^2 - 8)*q^4 +
1/8*(13*c^3 + 85*c^2 + 50*c - 186)*q^5 + 1/8*(-3*c^3 - 27*c^2 - 38*c +
6)*q^6 + 1/4*(-19*c^3 - 119*c^2 - 30*c + 170)*q^7 + O(q^8),
q + d*q^2 + 1/8*(-d^3 - 9*d^2 - 26*d - 22)*q^3 + (d^2 - 8)*q^4 +
1/8*(13*d^3 + 85*d^2 + 50*d - 186)*q^5 + 1/8*(-3*d^3 - 27*d^2 - 38*d +
6)*q^6 + 1/4*(-19*d^3 - 119*d^2 - 30*d + 170)*q^7 + O(q^8)
```

```
*]
> Newforms("37k2");
[* [*
q - 2*q^2 - 3*q^3 + 2*q^4 - 2*q^5 + 6*q^6 - q^7 + O(q^8)
*], [*
q + q^3 - 2*q^4 - q^7 + O(q^8)
*], [*
3/2 + q + 3*q^2 + 4*q^3 + 7*q^4 + 6*q^5 + 12*q^6 + 8*q^7 + O(q^8)
*] *]
```

0.12 Reductions and Embeddings

`Reductions(f, p)`

The mod p reductions of the modular forms f, where $p \in \mathbf{Z}$ is a prime number and f is a modular form over a number field (or the rationals or integers). Because of denominators, the list of reductions might be empty. (In some cases when f is defined over a field of large degree, the algorithm that I've implemented is definitely not close to optimal. I know a much better algorithm, but haven't implemented it yet.)

`pAdicEmbeddings(f, p)`

The p-adic embeddings of f.

`ComplexEmbeddings(f)`

The complex embeddings of f.

Example H0E17

We computed various reductions and embeddings of a degree 3 newform in $S_2(\Gamma_0(47))$.

```
> M := ModularForms(Gamma0(47),2);
> f := Newform(M,1);
> Degree(f);
4
> f;
q + a*q^2 + (a^3 - a^2 - 6*a + 4)*q^3 + (a^2 - 2)*q^4 + (-4*a^3 +
2*a^2 + 20*a - 10)*q^5 + (-a^2 - a + 1)*q^6 + (3*a^3 - a^2 - 16*a +
7)*q^7 + O(q^8)
> Parent(f);
Space of modular forms on Gamma_0(47) of weight 2 and dimension 4 over
Number Field with defining polynomial x^4 - x^3 - 5*x^2 + 5*x - 1 over
the Rational Field.
> Reductions(f,3);
[* [*
q + 2*q^2 + 2*q^3 + 2*q^4 + q^6 + q^7 + O(q^8)
```

```
*], [*
q + $.1^4*q^2 + $.1^25*q^3 + $.1^15*q^4 + $.1^20*q^5 + $.1^3*q^6 +
$.1^3*q^7 + O(q^8),
q + $.1^10*q^2 + $.1^17*q^3 + $.1^5*q^4 + $.1^24*q^5 + $.1*q^6 +
$.1*q^7 + O(q^8),
q + $.1^12*q^2 + $.1^23*q^3 + $.1^19*q^4 + $.1^8*q^5 + $.1^9*q^6 +
$.1^9*q^7 + O(q^8)
*] *]
```

The reductions are genuine modular forms, so we can compute them to higher precision later.

```
> f3 := Reductions(f,3)[1][1];
> Type(f3);
ModFrmElt
> Parent(f3);
Space of modular forms on Gamma_0(47) of weight 2 and dimension 4 over
Finite field of size 3.
> PowerSeries(f3,15);
q + 2*q^2 + 2*q^3 + 2*q^4 + q^6 + q^7 + q^9 + q^12 + 2*q^14 + O(q^15)
> f3^2;
q^2 + q^3 + 2*q^4 + q^7 + O(q^8)
> pAdicEmbeddings(f,3)[1][1];
q + (1383893738 + O(3^20))*q^2 + (288495368 + O(3^20))*q^3 +
(1448516780 + O(3^20))*q^4 + (291254407*3 + O(3^20))*q^5 + (654373882
+ O(3^20))*q^6 - (443261663 + O(3^20))*q^7 + O(q^8)
```

The p-adic precision can be increased by recreating p-adic field with higher precision.

```
> _ := pAdicField(3 :  Precision := 30);
> pAdicEmbeddings(f,3)[1][1];
q + (27072777983102 + O(3^30))*q^2 - (7262683411915 + O(3^30))*q^3 +
(102359491392536 + O(3^30))*q^4 - (26022742991723*3 + O(3^30))*q^5 +
(76458862719010 + O(3^30))*q^6 + (31185356420881 + O(3^30))*q^7 +
O(q^8)
> ComplexEmbeddings(f)[1][1];
q + 0.283841290783911427282405879557117103888*q^2 +
2.239254299847758500287247170743453726159900811628*q^3 -
1.919434121646123037854324315861907783948535827709*q^4 -
4.253514119234433634569963569478467752300447373828*q^5 +
0.635592830862211610571918436304790680059056881712*q^6 +
2.446577237818852279717904639620779818361588671448*q^7 + O(q^8)
```

0.13 Congruences

`CongruenceGroup(M1, M2, prec)`

A group C that measures all possible congruences (to precision `prec`) between some modular form in M_1 and some modular form in M_2. The group C is defined as follows. Let W_1 be the finite-rank $\mathbf{Z}$-module $q\text{-exp}(M_1) \cap \mathbf{Z}[[q]]$ and let W_2 be $q\text{-exp}(M_2) \cap \mathbf{Z}[[q]]$. Let V be the saturation of $W_1 + W_2$ in $\mathbf{Z}[[q]]$. Then $C = V/(W_1 + W_2)$.

Example H0E18

We verify that the newform corresponding to the first elliptic curve of rank 2 is congruent modulo 5 to some Galois-conjugate newform corresponding to the winding quotient of $J_0(389)$ (there is also a congruence modulo 2 to some conjugate form).

```
> M := ModularForms(Gamma0(389),2);
> f := Newform(M,1);
> Degree(f);
1
> g := Newform(M,5);
> Degree(g);
20
> CongruenceGroup(Parent(f),Parent(g),30);
Abelian Group isomorphic to Z/20
Defined on 1 generator
Relations:
    20*$.1 = 0
```

The congruence can be seen directly by computing the reductions of f and g modulo 5:

```
> fmod5 := Reductions(f,5);
> gmod5 := Reductions(g,5);  // takes a few seconds.
> #gmod5;
7
> #fmod5;
1
> [gbar :  gbar in gmod5 | #gbar eq 1];
[ [*
q + 4*q^2 + q^3 + 4*q^4 + q^5 + 4*q^6 + O(q^8)
*], [*
q + 3*q^2 + 3*q^3 + 2*q^4 + 2*q^5 + 4*q^6 + O(q^8)
*] ]
> fmod5[1][1];
q + 3*q^2 + 3*q^3 + 2*q^4 + 2*q^5 + 4*q^6 + O(q^8)
```

0.14 Algebraic Relations

`Relations(M, d, prec)`

The relations of degree d satisfied by the q-expansions of in the space M of modular forms. The q-expansions are computed to precision `prec`. If prec is too small, this intrinsic might return relations that are not really satisfied by the modular forms. To be sure of your result, `prec` must be at least as large as `PrecisionBound(M2)`, where M_2 has the same level as M and weight d times the weight of M.

Example H0E19

We compute an equation that defines the canonical embedding of $X_0(34)$.

```
> S := CuspidalSubspace(ModularForms(Gamma0(34)));
> Relations(S,4,20);
[
    a^3*c - a^2*b^2 - 3*a^2*c^2 + 2*a*b^3 + 3*a*b^2*c - 3*a*b*c^2 +
        4*a*c^3 - b^4 + 4*b^3*c - 6*b^2*c^2 + 4*b*c^3 - 2*c^4
]
[
    ( 0  0  1 -1  0 -3  2  3 -3  4 -1  4 -6  4 -2)
]
> // a, b, and c correspond to the cusp forms S.1, S.2 and S.3:
> S.1;
q - 2*q^4 - 2*q^5 + 4*q^7 + O(q^8)
> S.2;
q^2 - q^4 + O(q^8)
> S.3;
q^3 - 2*q^4 - q^5 + q^6 + 4*q^7 + O(q^8)
```

Next we compute the canonical embedding of $X_0(75)$.

```
> S := CuspidalSubspace(ModularForms(Gamma0(75)));
> R := Relations(S,2,20); R;
[
    a*c - b^2 - d^2 - 4*e^2,
    a*d - b*c + b*e + d*e - 3*e^2,
    a*e - b*d - c*e
]
> // NOTE: It is still much faster to compute in the power
> // series ring than the ring of modular forms!
> a, b, c, d, e := Explode([PowerSeries(f,20) :  f in Basis(S)]);
> a*c - b^2 - d^2 - 4*e^2;
O(q^21)
```

If you would like to understand the connection between the above computations and models for modular curves, see Steven Galbraith's Oxford Ph.D. thesis.

0.15 Elliptic Curves

Little has been implemented so far.

`ModularForm(E)`

The modular form associated to the elliptic curve E over **Q**.

`EllipticCurve(f)`

An elliptic curve E with associated modular form f, when f is a weight 2 newform on $\Gamma_0(N)$ with rational Fourier coefficients. When $N < 10000$ the Cremona database is used. (When $N > 10000$ I coded an algorithm for computing E, but it is currently not reasonable to expect my implementation to work at such high level.)

Example H0E20

```
> M := ModularForms(Gamma0(389),2);
> f := Newform(M,1);
> Degree(f);
1
> E := EllipticCurve(f);
> E;
Elliptic Curve defined by y^2 + y = x^3 + x^2 - 2*x over Rational
Field
> Conductor(E);
389
> time s := PowerSeries(f,200);  // faster because it knows the elliptic curve
Time:  0.509
```

0.16 Modular Symbols

`ModularSymbols(M)`

The sequence of characteristic 0 spaces of modular symbols with given sign associated to M, when this makes sense.

`ModularSymbols(M, sign)`

The sequence of characteristic 0 spaces of modular symbols with given sign associated to M, when this makes sense.

Example H0E21

```
> M := ModularForms(Gamma0(389),2);
> ModularSymbols(M,+1);
[
    Full Modular symbols space of level 389, weight 2, and dimension
    33
```

```
]
> ModularSymbols(M,-1);
[
Full Modular symbols space of level 389, weight 2, and dimension
32
]
> M := ModularForms(Gamma1(13),2);
> ModularSymbols(M);
[
    Full Modular symbols space of level 13, weight 2, and dimension 1,
    Full Modular symbols space of level 13, weight 2, character $.1,
    and dimension 0,
    Full Modular symbols space of level 13, weight 2, character $.1,
    and dimension 4,
    Full Modular symbols space of level 13, weight 2, character $.1,
    and dimension 0,
    Full Modular symbols space of level 13, weight 2, character $.1^2,
    and dimension 2,
    Full Modular symbols space of level 13, weight 2, character $.1,
    and dimension 2
]
> Basis($1[3]);
[
{-1/8, 0},
{-1/4, 0},
{-1/6, 0},
{oo, 0}
]
>
```

0.17 Bibliography

[**Atk67**] A. O. L. Atkin. Weierstrass points at cusps of $\Gamma_0(N)$. *Ann. of Math. (2)*, 85:42–45, 1967.

[**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.