[Tutorial] Modular Forms

Henri Cohen

June 22, 2017

Construction of Modular Forms

There are not so many ways to construct modular forms explicitly.

- Eisenstein Series.
- Theta functions associated to a lattice and a spherical polynomial.
- Modular forms attached to Hecke characters on imaginary and real quadratic fields.
- Modular forms associated to elliptic curves by Wiles's modularity theorem
- Trace forms using the Eichler–Selberg trace formula.
- Many types of operators enabling to construct new modular forms from existing ones.
- Certain explicit q-series or products: eta quotients, Nahm's conjecture, etc...

Implementation

Three types of objects:

- Modular form spaces, initialized by the command `mfinit` with a flag corresponding to one of the five spaces mentioned above.
- Modular forms themselves: if `F` is such a form, `mfcoefs(F,n)` gives the vector of coefficients $[a(0), a(1), ..., a(n)]$, and `mfparams(F)` gives $[N, k, \chi]$, level, weight, and character.
- Dirichlet characters: represented either by a discriminant D for the Kronecker–Legendre symbol (D/n) ($D = 1$ trivial character), by an `intmod` `Mod(a,N)` with $\gcd(a, N) = 1$ (Conrey numbering), or by a general `Pari/GP` group $[G, \chi]$.

Implementation

Three types of objects:

- Modular form spaces, initialized by the command `mfinit` with a flag corresponding to one of the five spaces mentioned above.
- Modular forms themselves: if `F` is such a form, `mfcoefs(F,n)` gives the vector of coefficients $[a(0), a(1), ..., a(n)]$, and `mfparams(F)` gives $[N, k, \chi]$, level, weight, and character.
- Dirichlet characters: represented either by a discriminant D for the Kronecker–Legendre symbol (D/n) ($D = 1$ trivial character), by an intmod `Mod(a,N)` with $\gcd(a, N) = 1$ (Conrey numbering), or by a general `Pari/GP` group $[G, \chi]$.

Implementation

Three types of objects:

- Modular form spaces, initialized by the command `mfinit` with a flag corresponding to one of the five spaces mentioned above.
- Modular forms themselves: if `F` is such a form, `mfcoefs(F,n)` gives the vector of coefficients $[a(0), a(1), \ldots, a(n)]$, and `mfparams(F)` gives $[N, k, \chi]$, level, weight, and character.
- Dirichlet characters: represented either by a discriminant D for the Kronecker–Legendre symbol (D/n) ($D = 1$ trivial character), by an `intmod` `Mod(a,N)` with $\gcd(a, N) = 1$ (Conrey numbering), or by a general `Pari/GP` group $[G, \chi]$.

Practice: Modular Form Leaves I

```
? D = mfDelta(); V = mfcoefs(D, 8)
% = [0, 1, -24, 252, -1472, 4830, -6048, -16744, 84480]
? Ser(V,q)
% = q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5
      - 6048*q^6 -16744*q^7 + 84480*q^8 + O(q^9)
```

```
? E4 = mfEk(4); E6 = mfEk(6); apply(x->mfcoefs(x,4),[E4,E6
% = [[1, 240, 2160, 6720, 17520],
     [1, -504, -16632, -122976, -532728]]
? E43 = mfpow(E4, 3); E62 = mfpow(E6, 2);
DP = mflinear([E43, E62], [1, -1]/1728); mfcoefs(DP, 6)
% = [0, 1, -24, 252, -1472, 4830, -6048]
? mfisequal(D, DP)
% = 1
```

Practice: Modular Form Leaves I

```
? D = mfDelta(); V = mfcoefs(D, 8)
% = [0, 1, -24, 252, -1472, 4830, -6048, -16744, 84480]
? Ser(V,q)
% = q - 24*q^2 + 252*q^3 - 1472*q^4 + 4830*q^5
      - 6048*q^6 -16744*q^7 + 84480*q^8 + O(q^9)
```

```
? E4 = mfEk(4); E6 = mfEk(6); apply(x->mfcoefs(x,4),[E4,E6
% = [[1, 240, 2160, 6720, 17520],
     [1, -504, -16632, -122976, -532728]]
? E43 = mfpow(E4, 3); E62 = mfpow(E6, 2);
DP = mflinear([E43, E62], [1, -1]/1728); mfcoefs(DP, 6)
% = [0, 1, -24, 252, -1472, 4830, -6048]
? mfisequal(D, DP)
% = 1
```

Practice: Modular Form Leaves II

```
? F = mfetaquo([1,2;11,2]); mfcoefs(F,10)
% = [0, 1, -2, -1, 2, 1, 2, -2, 0, -2, -2]
? G = mffromell(ellinit("11a1"))[2]; mfisequal(F, G)
% = 1
```

Here `mfetaquo` represents an eta quotient, here $\eta(\tau)^2\eta(11\tau)^2$. The corresponding modular form is equal to the modular form associated to the elliptic curve "11a1" of conductor 11.

The `mffromell` command returns a triple `[mf, F, coe]`, where `mf` is a modular form space, `F` is the form, and `coe` the coefficients of `F` on the basis of `mf`

Practice: Modular Form Spaces I

```
? mf = mfinit([1,12]); L = mfbasis(mf); #L
% = 2
? mfdim(mf)
% = 2
? mfcoefs(L[1],6)
% = [691/65520, 1, 2049, 177148, 4196353, 48828126]
? mfcoefs(L[2],6)
% = [0, 1, -24, 252, -1472, 4830, -6048]
```

The default is to ask for the full space $M_k(\Gamma_0(N),\chi)$ (`flag = 4`).

Note: for now, the Eisenstein series are given before the cusp forms, and they are normalized with $a(1)=1$, not $a(0)=1$ (which is impossible in general).

Practice: Modular Form Spaces II

The cuspidal space is with `flag = 1`:

```
? mf = mfinit([1,12], 1); L = mfbasis(mf); #L
% = 1
? mfcoefs(L[1],6)
% = [0, 1, -24, 252, -1472, 4830, -6048]
```

The newspace is with `flag = 0`:

```
? mf = mfinit([35,2], 0); L = mfbasis(mf);
? for (i = 1, #L, print(mfcoefs(L[i], 10)))
[0, 3, -1, 0, 3, 1, -8, -1, -9, 1, -1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9]
[0, 0, -8, 10, 4, -2, 4, 2, -4, -12, -8]
```

These are (essentially) random modular cusp forms. Usually, one wants eigenforms: this is obtained by the command `mfsplit`, which applies only to the newspace, even if the input is larger:

Practice: Modular Form Spaces II

The cuspidal space is with `flag = 1`:

```
? mf = mfinit([1,12], 1); L = mfbasis(mf); #L
% = 1
? mfcoefs(L[1],6)
% = [0, 1, -24, 252, -1472, 4830, -6048]
```

The newspace is with `flag = 0`:

```
? mf = mfinit([35,2], 0); L = mfbasis(mf);
? for (i = 1, #L, print(mfcoefs(L[i], 10)))
[0, 3, -1, 0, 3, 1, -8, -1, -9, 1, -1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9]
[0, 0, -8, 10, 4, -2, 4, 2, -4, -12, -8]
```

These are (essentially) random modular cusp forms. Usually, one wants eigenforms: this is obtained by the command `mfsplit`, which applies only to the newspace, even if the input is larger:

Practice: Modular Form Spaces III

```
? mf = mfsplit(mf); mffields(mf)
% = [y, y^2 - y - 4]
? L = mfeigenbasis(mf); #L
% = 2
? mfcoefs(L[1],10)
% = [0, 1, 0, 1, -2, -1, 0, 1, 0, -2, 0]
? mfcoefs(L[2],3)
% = [Mod(0, y^2 - y - 4), Mod(1, y^2 - y - 4),
    Mod(-y, y^2 - y - 4), Mod(y - 1, y^2 - y - 4)]
? lift(mfcoefs(L[2],9))
% = [0, 1, -y, y - 1, y + 2, 1, -4, -1, -y - 4, -y + 2]
```

Practice: Modular Form Spaces IV

Very often, need numerical values of coefficients: need to embed in $\mathbb{C}$, so a given eigenform can give several forms.

```
? F=mfembed(L[2]); for(i=1,2,print(mfcoefs(F[i],5)))
[0, 1, 1.5615528128088302749107049279870385126,
      -2.5615528128088302749107049279870385126,
       0.43844718719116972508929507201296148743, 1]
[0, 1, -2.5615528128088302749107049279870385126,
        1.5615528128088302749107049279870385126,
        4.5615528128088302749107049279870385126, 1]
```

```
? [mf,F,co] = mffromell(ellinit("35a1")); mfcoefs(F, 10)
% = [0, 1, 0, 1, -2, -1, 0, 1, 0, -2, 0]
? mfisequal(F, L[1])
% = 1
```

Practice: Modular Form Spaces IV

Very often, need numerical values of coefficients: need to embed in $\mathbb{C}$, so a given eigenform can give several forms.

```
? F=mfembed(L[2]); for(i=1,2,print(mfcoefs(F[i],5)))
[0, 1, 1.5615528128088302749107049279870385126,
      -2.5615528128088302749107049279870385126,
       0.43844718719116972508929507201296148743, 1]
[0, 1, -2.5615528128088302749107049279870385126,
        1.5615528128088302749107049279870385126,
        4.5615528128088302749107049279870385126, 1]
```

```
? [mf,F,co] = mffromell(ellinit("35a1")); mfcoefs(F, 10)
% = [0, 1, 0, 1, -2, -1, 0, 1, 0, -2, 0]
? mfisequal(F, L[1])
% = 1
```

Practice: Modular Form Spaces V

```
? apply(x->mfdim([96, 2], x), [0..4])
% = [2, 9, 7, 15, 24]
? mf = mfinit([96,2]); L = mfbasis(mf);
? for (i = 12, 15, print(mfcoefs(L[i], 14)))
[23/24, 1, 3, 4, 7, 6, 12, 8, 15, 13, 18, 12, 28]
[31/24, 1, 3, 4, 7, 6, 12, 8, 15, 13, 18, 12, 28]
[47/24, 1, 3, 4, 7, 6, 12, 8, 15, 13, 18, 12, 28]
[95/24, 1, 3, 4, 7, 6, 12, 8, 15, 13, 18, 12, 28]
```

Practice: Modular Form Spaces VI

```
? F = mflinear([L[14],L[12]],[1,-1]); mfcoefs(F, 50)
% = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 0, 0]
? G = mfhecke(F, 24); mfcoefs(G, 11)
% = [1, 24, 24, 96, 24, 144, 96, 192, 24, 312, 144, 288]
? mftobasis(mf, G)
% = [0, 0, 0, 0, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0]~
? 24*mfcoefs(L[5], 11)
% = [1, 24, 24, 96, 24, 144, 96, 192, 24, 312, 144, 288]
```

Practice: Modular Form Spaces VII

```
? mf=mfsplit([96,2]); mffields(mf)
% = [y, y]
? L = mfeigenbasis(mf); for(i=1,2,print(mfcoefs(L[i], 15))
    [0, 1, 0, 1, 0, 2, 0, -4, 0, 1, 0, 4, 0, -2, 0, 2]
    [0, 1, 0, -1, 0, 2, 0, 4, 0, 1, 0, -4, 0, -2, 0, -2]
? Fa = mffromell(ellinit("96a1"))[2]; mfcoefs(Fa, 15)
% = [0, 1, 0, 1, 0, 2, 0, -4, 0, 1, 0, 4, 0, -2, 0, 2]
? Fb = mffromell(ellinit("96b1"))[2]; mfcoefs(Fb, 15)
% = [0, 1, 0, -1, 0, 2, 0, 4, 0, 1, 0, -4, 0, -2, 0, -2]
? mfisequal(mftwist(Fa, -4), Fb)
% = 1
```

Practice: Spaces with Characters I

```
? mf = mfsplit([35,2,5]); mffields(mf)
% = [y^2 + 1]
? F = mfeigenbasis(mf)[1]; lift(mfcoefs(F, 10))
% = [0, 1, 2*y, -y, -2, -y - 2, 2, -y, 0, 2, -4*y + 2]
```

Here, the last 5 in the command `mfsplit([35,2,5])` represents the quadratic Legendre character $(5/n)$.

Because `mffields` gives y^2+1, in the last output y is equal to one of the two roots of $y^2+1=0$.

Practice: Spaces with Characters II

```
? mf = mfinit([23,1,-23], 0); mfdim(mf)
% = 1
? F = mfbasis(mf)[1]; mfcoefs(F, 15)
% = [0, 1, -1, -1, 0, 0, 1, 0, 1, 0, 0, 0, 0, -1, 0, 0]
? mfgaloistype(mf,F)
% = 6
```

`mfgaloistype` asks for the image in $\mathrm{PSL}_2(\mathbb{C})$ of the projective representation associated to the form, here D_6. This command is specific to weight 1 forms.

Practice: Spaces with Characters III

Since the form is of type D_6, hence dihedral, it can be obtained by theta functions. In the present case, the command `mffromqf` does what is required:

```
? F1 = mffromqf([2,1;1,12])[2]; V1 = mfcoefs(F1, 15)
% = [1, 2, 0, 0, 2, 0, 4, 0, 4, 2, 0, 0, 4, 0, 0, 0]
? F2 = mffromqf([4,1;1,6])[2]; V2 = mfcoefs(F2, 15)
% = [1, 0, 2, 2, 2, 0, 2, 0, 2, 2, 0, 0, 4, 2, 0, 0]
? (V1 - V2)/2
% = [0, 1, -1, -1, 0, 0, 1, 0, 1, 0, 0, 0, 0, -1, 0, 0]
? mfisequal(F, mflinear([F1, F2], [1, -1]/2))
% = 1
```

Practice: Modular Forms of Weight One I

Note that when we wrote `mf=mfinit([23,1,-23], 0)`, we knew that the character to choose is $(-23/n)$. What if we don't ? This is especially important for weight 1 modular forms.

```
? G = znstar(23, 1);
? L = [[G,chi]|chi<-znchargalois(G),zncharisodd(G,chi)];#L
% = 2
? apply(x->mfdim([23,1,x], 0), L)
% = [0, 1]
? apply(x->charorder(x[1],x[2]), L)
% = [22, 2]
```

The above shows the most general way to define a Dirichlet character: first define the group G using `znstar(N,1)` (flag 1 necessary), then specify `chi` on generators, e.g., using `znchargalois` or otherwise.

Practice: Modular Forms of Weight One II

```
? mfa = mfinit([23,1,0], 0); #mfa
% = 1
? mf = mfa[1]; mfdim(mf)
% = 1
? mfparams(mf)
% = [23, 1, -23]
```

This illustrates wildcards: the 0 (which is of course not limited to weight 1) means that the result is a vector of `mf` of all nonzero spaces with given level and weight, but varying character (here, `mfparams` says that the only one is $(-23/n)$).

Practice: Modular Forms of Weight One III

```
wt1exp(lim1,lim2)=
{
  my(mfall,mf,chi,chiz,ord,M,res,V);
  for(N=lim1,lim2,
    mfall=mfinit([N,1,0], 0); /* Use wildcard, more effici
    for(i=1,#mfall,
      mf=mfsplit(mfall[i]);
      chi=mfparams(mf)[3]; /* nice format: D or Mod(a,N) */
      chiz=znchar(chi); /* necessary to use charorder */
      ord=charorder(chiz[1],chiz[2]);
      M=mfeigenbasis(mf);
      for(k=1,#M,
        res=mfgaloistype(mf,M[k]);
        if(res<0,print([N,chi,k,ord,-res]))
      )
    )
  );
```

Practice: Modular Forms of Weight One IV

Copy the preceding program from the GP file available with the tutorial on the website: it explores “exotic” weight 1 forms between given levels, i.e., those whose projective image is not dihedral, so cannot easily be constructed explicitly (image A_4 code -12, S_4 code -24, A_5 code -60, opposite of their cardinality).

For instance, try `w1exp(1,230)`, or `w1exp(633,633)`. The latter outputs

```
[633, Mod(71, 633), 2, 10, 60]
```

Practice: Modular Forms of Weight One IV

Copy the preceding program from the GP file available with the tutorial on the website: it explores "exotic" weight 1 forms between given levels, i.e., those whose projective image is not dihedral, so cannot easily be constructed explicitly (image A_4 code -12, S_4 code -24, A_5 code -60, opposite of their cardinality).

For instance, try `w1exp(1,230)`, or `w1exp(633,633)`. The latter outputs

```
[633, Mod(71, 633), 2, 10, 60]
```

Practice: Miscellaneous Commands I

```
? mf=mfsplit([96,6]); mffields(mf)
% = [y, y, y, y, y, y, y^2 - 31, y^2 - 31]
? mfatkineigenvalues(mf,3)
% = [[-1], [-1], [-1], [1], [1], [1], [-1, -1], [1, 1]]
? mf=mfsplit([96,3,-3]); mffields(mf)
% = [y^4 + 8*y^2 + 9, y^4 + 4*y^2 + 1]
? mfatkineigenvalues(mf,3)
  ***   at top-level: mfatkineigenvalues(mf,3)
  ***                 ^-------------------------
  *** mfatkineigenvalues: sorry, pseudo eigenvalues
      for W_Q is not yet implemented.
? mfatkineigenvalues(mf,32)
% = [[I, -I, -I, I], [-I, I, I, -I]]
```

It is normal to have an error in the command `mfatkineigenvalues(mf,3)`.

Practice: Miscellaneous Commands II

```
? mf = mfinit([96,2], 1); L = mfbasis(mf);
? apply(x->mfconductor(mf,x), L)
% = [24, 48, 96, 32, 96, 48, 96, 96, 96]
```

```
? mf = mfsplit([35,2]); L=mfbasis(mf);
? for (i=1,#L,print(mfcoefs(L[i],14)))
[0, 3, -1, 0, 3, 1, -8, -1, -9, 1, -1, -2, 4, 10, 1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9, 8, -20, 6, -9]
[0, 0, -8, 10, 4, -2, 4, 2, -4, -12, -8, -12, 12, -6, 8]
? for (i=1,#L,print(mfcuspexpansion(mf,L[i],1/5,14)));
[0, 1, -1, -2, 7, 3, -8, -3, -9, 5, -1, 4, 8, 0, 1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9, 8, -20, 6, -9]
[0, -2, -8, 8, 8, 0, 4, 0, -4, -8, -8, -6, 16, -16, 8]
```

Practice: Miscellaneous Commands II

```
? mf = mfinit([96,2], 1); L = mfbasis(mf);
? apply(x->mfconductor(mf,x), L)
% = [24, 48, 96, 32, 96, 48, 96, 96, 96]
```

```
? mf = mfsplit([35,2]); L=mfbasis(mf);
? for (i=1,#L,print(mfcoefs(L[i],14)))
[0, 3, -1, 0, 3, 1, -8, -1, -9, 1, -1, -2, 4, 10, 1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9, 8, -20, 6, -9]
[0, 0, -8, 10, 4, -2, 4, 2, -4, -12, -8, -12, 12, -6, 8]
? for (i=1,#L,print(mfcuspexpansion(mf,L[i],1/5,14)));
[0, 1, -1, -2, 7, 3, -8, -3, -9, 5, -1, 4, 8, 0, 1]
[0, -1, 9, -8, -11, -1, 4, 1, 13, 7, 9, 8, -20, 6, -9]
[0, -2, -8, 8, 8, 0, 4, 0, -4, -8, -8, -6, 16, -16, 8]
```

Practice: Miscellaneous Commands III

```
? C = mfcusps(108)
% = [0, 1/2, 1/3, 2/3, 1/4, 1/6, 5/6, 1/9, 2/9, 1/12,
       5/12, 1/18, 5/18, 1/27, 1/36, 5/36, 1/54, 1/108]
? apply(x->mfcuspwidth(108,x), C)
% = [108, 27, 12, 12, 27, 3, 3, 4, 4, 3,
        3, 1, 1, 4, 1, 1, 1, 1]
? NK = [108,3,-4]; apply(x->mfcuspisregular(NK,x), C)
% = [1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1]
? [c | c<-C, !mfcuspisregular(NK,c)]
% = [1/2, 1/6, 5/6, 1/18, 5/18, 1/54]
```

Practice: Miscellaneous Commands IV

```
? E4 = mfEk(4); G = mfderivE2(E4); mfcoefs(G, 6)
% = [-1/3, 168, 5544, 40992, 177576, 525168, 1352736]
? mfcoefs(mfEk(6), 6)/(-3)
% = [-1/3, 168, 5544, 40992, 177576, 525168, 1352736]
? F = mfderivE2(E4, 3); (-9)*mfcoefs(F, 5)
% = [1, -264, -135432, -5196576, -69341448, -515625264]
? mfisequal(mfEk(10), mflinear([F],[-9]))
% = 1
```

```
? E4 = mfEk(4); mfeval(E4,I)
% = 1.4557628922687093224624220035988692874
? 3*gamma(1/4)^8/(2*Pi)^6
% = 1.4557628922687093224624220035988692874
```

Practice: Miscellaneous Commands IV

```
? E4 = mfEk(4); G = mfderivE2(E4); mfcoefs(G, 6)
% = [-1/3, 168, 5544, 40992, 177576, 525168, 1352736]
? mfcoefs(mfEk(6), 6)/(-3)
% = [-1/3, 168, 5544, 40992, 177576, 525168, 1352736]
? F = mfderivE2(E4, 3); (-9)*mfcoefs(F, 5)
% = [1, -264, -135432, -5196576, -69341448, -515625264]
? mfisequal(mfEk(10), mflinear([F],[-9]))
% = 1
```

```
? E4 = mfEk(4); mfeval(E4,I)
% = 1.4557628922687093224624220035988692874
? 3*gamma(1/4)^8/(2*Pi)^6
% = 1.4557628922687093224624220035988692874
```

Practice: Miscellaneous Commands V

```
? mf = mfinit([96,4], 0); M = mfmathecke(mf, 7)

% =
[0     0   0    372    696   0]

[0     0  36      0      0 -96]

[0  27/5   0 -276/5 -276/5   0]

[1     0 -12      0      0  62]

[0     0   1      0      0 -16]

[0  -3/5   0   14/5  -16/5   0]
```

Practice: Miscellaneous Commands VI

```
? P = charpoly(M)
% = x^6 - 1456*x^4 + 209664*x^2 - 2985984
? print(factor(P))
[x - 36, 1; x - 12, 1; x - 4, 1; x + 4, 1;
                       x + 12, 1; x + 36, 1]
```

Note that this shows that all the eigenvalues of $T(7)$ are integral, so the splitting will be entirely rational and the eigenforms with integral coefficients. Let's check:

Practice: Miscellaneous Commands VII

```
? mf = mfsplit(mf); mffields(mf)
% = [y, y, y, y, y, y]
? L = mfeigenbasis(mf); for(i=1,6,print(mfcoefs(L[i],15)))
[0, 1, 0, 3, 0, 10, 0, 4, 0, 9, 0, -20, 0, 70, 0, 30]
[0, 1, 0, 3, 0, 2, 0, 12, 0, 9, 0, 60, 0, -42, 0, 6]
[0, 1, 0, 3, 0, -14, 0, -36, 0, 9, 0, -36, 0, 54, 0, -42]
[0, 1, 0, -3, 0, 10, 0, -4, 0, 9, 0, 20, 0, 70, 0, -30]
[0, 1, 0, -3, 0, 2, 0, -12, 0, 9, 0, -60, 0, -42, 0, -6]
[0, 1, 0, -3, 0, -14, 0, 36, 0, 9, 0, 36, 0, 54, 0, 42]
```

Note again the twisting phenomenon: there are three eigenforms, and three twists by the character $(-4/n)$.

Practice: Miscellaneous Commands VIII

```
? M = mfmatatkin(mf,3)
? [matdet(M), mfatkineigenvalues(mf,3)]

% =
[   0 -3    0    0  -24   0]

[-1/3  0 -4/3    0    0 -12]

[   0  0    0 -9/5 -6/5   0]

[   0  0 -2/3    0    0  -1]

[   0  0  1/6    0    0 3/2]

[   0  0    0  1/5  4/5   0]
% = [-1, [[-1], [-1], [-1], [1], [1], [1]]]
```

Practice: Combination with L-Functions I

```
? LE = mftolfun(mfEk(4), 1); lfun(LE, 2)/Pi^2
% = -3.3333333333333333333333333333333333333
? lfun(LE, 0)
% = -1.0000000000000000000000000000000000000
? D = mfDelta(); L = mftolfun(D, 3);
? lfunlambda(L, 3)/lfunlambda(L, 5)
% = 1.5555555555555555555555555555555555556
? r = lfunlambda(L, 1)/lfunlambda(L, 3)
% = 2.3444283646888567293777134587554269175
? bestappr(r)
% = 1620/691
```

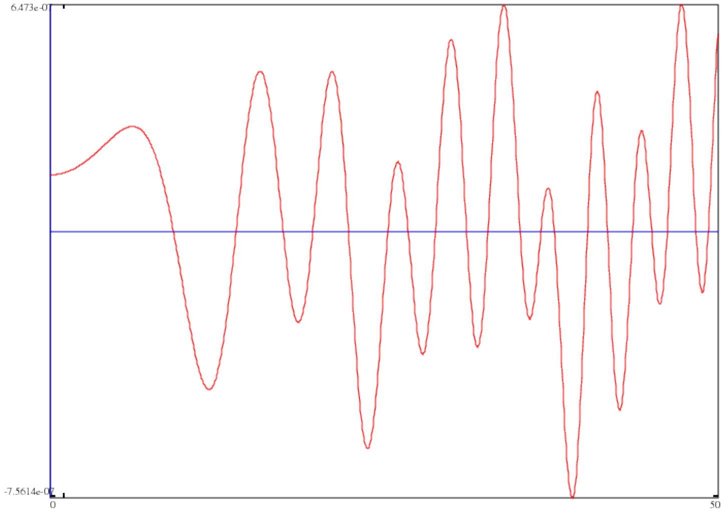
```
? LIN = lfuninit(L, [6, 6, 50]);
? ploth(t = 0, 50, lfunhardy(LIN, t))
```

Practice: Combination with L-Functions I

```
? LE = mftolfun(mfEk(4), 1); lfun(LE, 2)/Pi^2
% = -3.333333333333333333333333333333333333
? lfun(LE, 0)
% = -1.000000000000000000000000000000000000
? D = mfDelta(); L = mftolfun(D, 3);
? lfunlambda(L, 3)/lfunlambda(L, 5)
% = 1.555555555555555555555555555555555556
? r = lfunlambda(L, 1)/lfunlambda(L, 3)
% = 2.3444283646888567293777134587554269175
? bestappr(r)
% = 1620/691


? LIN = lfuninit(L, [6, 6, 50]);
? ploth(t = 0, 50, lfunhardy(LIN, t))
```

Practice: Combination with L-Functions II



Practice: Combination with L-Functions III

```
? PP = mfperiodpol(mfDelta(),-1); PP /= polcoeff(PP,1);
? bestappr(PP)
% = x^9 - 25/4*x^7 + 21/2*x^5 - 25/4*x^3 + x
? PM = mfperiodpol(mfDelta(),1); PM /= polcoeff(PM,0);
? bestappr(PP)
% = -x^10 + 691/36*x^8 - 691/12*x^6
          + 691/12*x^4 - 691/36*x^2 + 1
? mfperiodpolbasis(12)
% = [x^8 - 3*x^6 + 3*x^4 - x^2,
     4*x^9 - 25*x^7 + 42*x^5 - 25*x^3 + 4*x, x^10 - 1]
```

Practice: Miscellaneous Commands IX

```
? E4 = mfEk(4); F = mfbracket(E4, E4, 2); mfcoefs(F, 6)/48
% = [0, 1, -24, 252, -1472, 4830, -6048]
? D = mfDelta(); mftaylor(D, 9)*1728
% = [1, 0, -1/12, 0, 1/96, 0, 1/288, 0, -11/2304, 0]
? D3 = mftwist(D, -3); mfcoefs(D3, 9)
% = [0, 1, 24, 0, -1472, -4830, 0, -16744, -84480, 0]
? P = mfparams(D3)
% = [9, 12, 1]
? mf = mfinit(P, 1); mftobasis(mf, D3)
% = [0, 0, 0, 0, 0,
      5546/4131, -1232/12393, -47/16524, 11/24786]~
```

Practice: Miscellaneous Commands X

```
? F = mffromell(ellinit("49a1"))[2]; mfisCM(F)
% = -7
? mfisequal(F, mftwist(F, -7))
% = 1
? mf = mfsplit([23,1,-23], 1);
? F = mfeigenbasis(mf)[1]; mfisCM(F)
% = -23
? mfisequal(F, mftwist(F, -23))
% = 0
```

Practice: Searching: `mfsearch` I

We want to search for normalized eigenforms with integral (equivalently, rational) Fourier coefficients, given a few $a(p)$ for p prime, possibly modulo something.

```
? L = mfsearch([30,4], [[2,2],[3,-1]]); #L
% = 1
? [N, F] = L[1]; mfparams(F)
% = [26, 4, 1]
? mfcoefs(F, 10)
% = [0, 1, 2, -1, 4, 17, -2, -35, 8, -26, 34]
```

Practice: Searching: mfsearch II

```
? L=mfsearch([30,4], [[2,Mod(2,5)], [3,Mod(-1,5)]]); #L
% = 2
? apply(x->x[1], L)
% = [26, 26]
? F1 = L[1][2]; mfcoefs(F1, 10)
% = [0, 1, 2, -1, 4, 17, -2, -35, 8, -26, 34]
? F2 = L[2][2]; mfcoefs(F2, 10)
% = [0, 1, 2, 4, 4, -18, 8, 20, 8, -11, -36]
? F = mflinear([F1, F2], [-1, 1]); mfcoefs(F, 14)/5
% = [0, 0, 0, 1, 0, -7, 2, 11, 0, 3, -14, -10, 4, 0, 22]
? mfsturm([26,4])
% = 15
```

Thank you for your attention !