Modular_Forms_Sage

February 5, 2022

1 Modular Forms

Some introductory material on Sage computations for modular forms and modular symbols _Surya Teja Gavva

To get started: Type > sage -n jupyter on terminal. (On a computer with sage installed)

Upload the file Modular_Forms_Sage.ipynb to jupyter notebook and run it

2

2.1 Congruence Subgroups

[8]:
```
G = SL(2,ZZ)
```

[9]:
```
G
```

[9]:
```
Special Linear Group of degree 2 over Integer Ring
```

Generators

[10]:
```
G.gens()
```

[10]:
```
(
[ 0  1]  [1 1]
[-1  0], [0 1]
)
```

[11]:
```
G= Gamma0(10)
```

[12]:
```
G
```

[12]:
```
Congruence Subgroup Gamma0(10)
```

[18]:
```
G.generators()
```

[18]:
```
[
[1 1]  [ 3 -1]  [ 19  -7]  [11 -5]  [ 7 -5]
[0 1], [10 -3], [ 30 -11], [20 -9], [10 -7]
]
```

[41]:
```
Gamma(2).generators(algorithm="todd-coxeter")
```

[41]:
```
[
[1 2]  [-1  0]  [ 1  0]  [-1  0]  [-1  2]  [-1  0]  [1 0]
[0 1], [ 0 -1], [-2  1], [ 0 -1], [-2  3], [ 2 -1], [2 1]
]
```

[51]:
```
[Gamma0(n).index() for n in [1..20]]
```

[51]:
```
[1, 3, 4, 6, 6, 12, 8, 12, 12, 18, 12, 24, 14, 24, 24, 24, 18, 36, 20, 36]
```

[52]:
```
Gamma1(3).image_mod_n()
```

[52]:
```
Matrix group over Ring of integers modulo 3 with 1 generators (
[1 1]
[0 1]
)
```

[42]:
```
Gamma0(20).is_even()
```

[42]:
```
True
```

[43]:
```
Gamma(3).is_normal()
```

[43]:
```
True
```

[44]:
```
Gamma1(3).is_normal()
```

[44]:
```
False
```

Coset Representative in SL_2(Z)

[16]:
```
list(G.coset_reps())
```

[16]:
```
[
[1 0]  [ 0 -1]  [1 0]  [ 0 -1]  [ 0 -1]  [ 0 -1]  [ 0 -1]  [ 0 -1]
[0 1], [ 1  0], [1 1], [ 1  2], [ 1  3], [ 1  4], [ 1  5], [ 1  6],

[ 0 -1]  [ 0 -1]  [ 0 -1]  [1 0]  [1 1]  [1 2]  [1 3]  [1 4]  [1 0]
[ 1  7], [ 1  8], [ 1  9], [2 1], [2 3], [2 5], [2 7], [2 9], [5 1],

[-2 -1]
[ 5  2]
```

]

Dimension of New Cusp Forms

[17]:
```
Gamma0(110).dimension_new_cusp_forms()
```

[17]: 5

[37]:
```
Gamma1(31).dimension_cusp_forms(2)
26
```

[37]: 26

[38]:
```
Gamma1(31).dimension_modular_forms(2)
55
```

[38]: 55

[39]:
```
Gamma(13).dimension_modular_forms(1)
```

```
       ␣
↪---------------------------------------------------------------------------

       NotImplementedError                       Traceback (most recent call␣
↪last)

       <ipython-input-39-ee4b81dc02bb> in <module>
   ----> 1 Gamma(Integer(13)).dimension_modular_forms(Integer(1))


       /var/tmp/sage-9.4-current/local/lib/python3.9/site-packages/sage/modular/
↪arithgroup/arithgroup_generic.py in dimension_modular_forms(self, k)
      1151             NotImplementedError: Computation of dimensions of weight␣
↪1 cusp forms spaces not implemented in general
      1152         """
   -> 1153         return self.dimension_cusp_forms(k) + self.dimension_eis(k)
      1154
      1155     def dimension_cusp_forms(self, k=2):


       /var/tmp/sage-9.4-current/local/lib/python3.9/site-packages/sage/modular/
↪arithgroup/arithgroup_generic.py in dimension_cusp_forms(self, k)
      1213                 return ZZ(0)
      1214             else:
   -> 1215                 raise NotImplementedError("Computation of␣
↪dimensions of weight 1 cusp forms spaces not implemented in general")
```

```
   1216
   1217     def dimension_eis(self, k=2):


     NotImplementedError: Computation of dimensions of weight 1 cusp forms␣
↪spaces not implemented in general
```

[24]:
```
G.is_subgroup(Gamma0(10))
```

[24]: True

[22]:
```
G.is_subgroup(Gamma1(20))
```

[22]: False

Farey Symbol

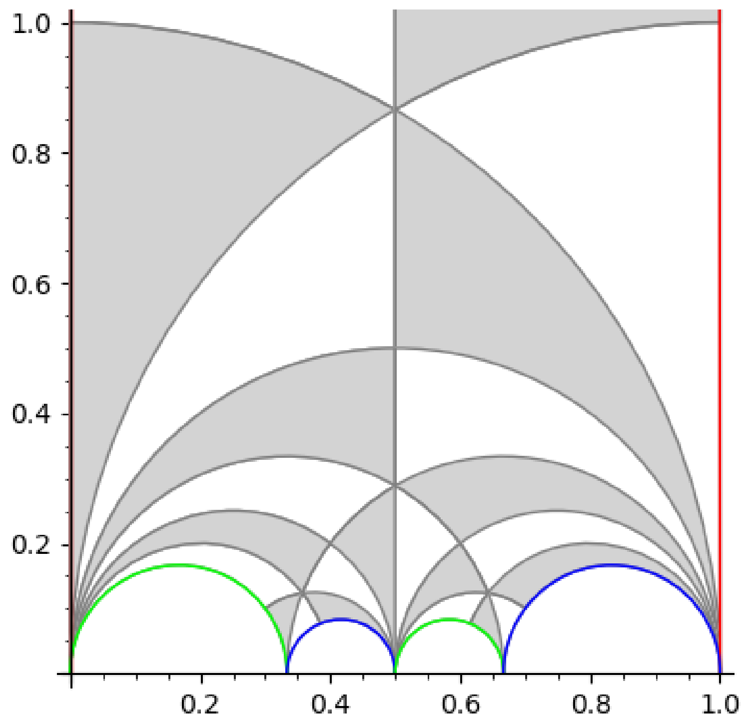
[57]:
```
f=Gamma1(4).farey_symbol()
```

[58]:
```
f.generators()
```

[58]:
```
[
[1 1]  [-3  1]
[0 1], [-4  1]
]
```

[60]:
```
FareySymbol(Gamma0(11)).fundamental_domain()
```

[60]:



Number of Cusps

[26]:
```
[Gamma0(n).ncusps() for n in [1..20]]
```

[26]: [1, 2, 2, 3, 2, 4, 2, 4, 4, 4, 2, 6, 2, 4, 4, 6, 2, 8, 2, 6]

[28]:
```
[Gamma0(n).ncusps() for n in prime_range(2,100)]
```

[28]: [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

[36]:
```
Gamma0(42).cusps()
```

[36]: [0, 1/21, 1/14, 1/7, 1/6, 1/3, 1/2, Infinity]

[49]:
```
G = Gamma1(5).as_permutation_group()
```

[50]:
```
G.cusp_widths()
```

[50]: [1, 1, 5, 5]

[33]:
```
Gamma0(7).are_equivalent(Cusp(1/3), Cusp(1/7))
```

[33]: False

[34]:
```
[Gamma0(100).cusp_width(c) for c in Gamma0(100).cusps()]
```

[34]: [100, 1, 4, 1, 1, 1, 4, 25, 1, 1, 4, 1, 25, 4, 1, 4, 1, 1]

[53]:
```
Gamma1(4).is_regular_cusp(Cusps(1/2))
```

[53]: False

[54]:
```
Gamma0(24).reduce_cusp(Cusps(-1/4))
```

[54]: 1/4

[56]:
```
Gamma0(1).reduce_cusp(Cusps(-1/4))
```

[56]: Infinity

Elliptic Points

[30]:
```
Gamma0(7).nu3()
```

[30]: 2

[31]:
```
Gamma0(1105).nu2()
```

[31]: 8

2.2 Creating Modular Forms

[1]:
```
M= ModularForms(Gamma0(4), 4)
```

[2]:
```
M
```

[2]: Modular Forms space of dimension 3 for Congruence Subgroup Gamma0(4) of weight 4 over Rational Field

[79]:
```
M.base_ring()
```

[79]: Rational Field

[66]:
```
M1= ModularForms(Gamma1(4), 10, prec=12)
```

[67]:
```
M1
```

[67]: Modular Forms space of dimension 6 for Congruence Subgroup Gamma1(4) of weight 10 over Rational Field

[68]:
```
M1.basis()
```

[68]:
```
[
q + 246*q^5 - 3136*q^7 + 15885*q^9 - 45696*q^11 + O(q^12),
q^2 + 16*q^4 - 156*q^6 + 256*q^8 + 870*q^10 + O(q^12),
q^3 - 4*q^5 - 14*q^7 + 72*q^9 + 67*q^11 + O(q^12),
1 - 264*q^4 - 135432*q^8 + O(q^12),
q + 19684*q^3 + 1953126*q^5 + 40353608*q^7 + 387440173*q^9 + 2357947692*q^11 +
O(q^12),
q^2 + 512*q^4 + 19684*q^6 + 262144*q^8 + 1953126*q^10 + O(q^12)
]
```

[96]:
```
M1.gens()
```

[96]:
```
[
q + 246*q^5 - 3136*q^7 + 15885*q^9 - 45696*q^11 + O(q^12),
q^2 + 16*q^4 - 156*q^6 + 256*q^8 + 870*q^10 + O(q^12),
q^3 - 4*q^5 - 14*q^7 + 72*q^9 + 67*q^11 + O(q^12),
1 - 264*q^4 - 135432*q^8 + O(q^12),
q + 19684*q^3 + 1953126*q^5 + 40353608*q^7 + 387440173*q^9 + 2357947692*q^11 +
O(q^12),
q^2 + 512*q^4 + 19684*q^6 + 262144*q^8 + 1953126*q^10 + O(q^12)
]
```

[75]:
```
M1.cuspidal_subspace().basis()
```

[75]:
```
[
q + 246*q^5 - 3136*q^7 + 15885*q^9 - 45696*q^11 + O(q^12),
q^2 + 16*q^4 - 156*q^6 + 256*q^8 + 870*q^10 + O(q^12),
q^3 - 4*q^5 - 14*q^7 + 72*q^9 + 67*q^11 + O(q^12)
]
```

[95]:
```
M1.eisenstein_subspace()
```

[95]:
```
Eisenstein subspace of dimension 3 of Modular Forms space of dimension 6 for
Congruence Subgroup Gamma1(4) of weight 10 over Rational Field
```

[115]:
```
M1.modular_symbols()
```

[115]:
```
Modular Symbols space of dimension 9 for Gamma_1(4) of weight 10 with sign 0
over Rational Field
```

[116]:
```
M1.new_subspace()
```

[116]:
```
Modular Forms subspace of dimension 1 of Modular Forms space of dimension 6 for
Congruence Subgroup Gamma1(4) of weight 10 over Rational Field
```

[86]:
```
ModularForms(Gamma0(11),2).character()
```

[86]:
```
Dirichlet character modulo 11 of conductor 1 mapping 2 |--> 1
```

[71]:
```
[ModularForms(Gamma1(7),k).dimension() for k in [2,3,4,5,6]]
```

[71]:
```
[5, 7, 9, 11, 13]
```

[72]:
```
chi = DirichletGroup(109, CyclotomicField(3)).0
```

[73]:
```
ModularForms(chi, 2, base_ring = CyclotomicField(15))
```

[73]:
```
Modular Forms space of dimension 10, character [zeta3 + 1] and weight 2 over
Cyclotomic Field of order 15 and degree 8
```

[ ]:
```

```

[81]:
```
M2=ModularForms(11,2,base_ring=GF(13))
```

[82]:
```
M2.basis()
```

[82]:
```
[
q + 11*q^2 + 12*q^3 + 2*q^4 + q^5 + O(q^6),
1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + O(q^6)
]
```

[83]:
```
ModularForms(11,2).basis()
```

[83]:
```
[
q - 2*q^2 - q^3 + 2*q^4 + q^5 + O(q^6),
1 + 12/5*q + 36/5*q^2 + 48/5*q^3 + 84/5*q^4 + 72/5*q^5 + O(q^6)
]
```

[89]:
```
M2.cuspidal_submodule()
```

[89]:
```
Cuspidal subspace of dimension 1 of Modular Forms space of dimension 2 for
Congruence Subgroup Gamma0(11) of weight 2 over Finite Field of size 13
```

[105]:
```
chi = DirichletGroup(25,QQ).0
```

[106]:
```
n = ModularForms(chi,2)
```

[107]:
```
n.basis()
```

[107]:
```
[
1 + O(q^6),
q + O(q^6),
```

```
q^2 + O(q^6),
q^3 + O(q^6),
q^4 + O(q^6),
q^5 + O(q^6)
]
```

[108]:
```
n.set_precision(20)
```

[109]:
```
n.basis()
```

[109]:
```
[
1 + 10*q^10 + 20*q^15 + O(q^20),
q + 5*q^6 + q^9 + 12*q^11 - 3*q^14 + 17*q^16 + 8*q^19 + O(q^20),
q^2 + 4*q^7 - q^8 + 8*q^12 + 2*q^13 + 10*q^17 - 5*q^18 + O(q^20),
q^3 + q^7 + 3*q^8 - q^12 + 5*q^13 + 3*q^17 + 6*q^18 + O(q^20),
q^4 - q^6 + 2*q^9 + 3*q^14 - 2*q^16 + 4*q^19 + O(q^20),
q^5 + q^10 + 2*q^15 + O(q^20)
]
```

[117]:
```
e = DirichletGroup(27,CyclotomicField(3)).0**2
```

[119]:
```
M = ModularForms(e,2,prec=10).eisenstein_subspace()
```

[120]:
```
M.eisenstein_series()
```

[120]:
```
[
-1/3*zeta6 - 1/3 + q + (2*zeta6 - 1)*q^2 + q^3 + (-2*zeta6 - 1)*q^4 + (-5*zeta6
+ 1)*q^5 + O(q^6),
-1/3*zeta6 - 1/3 + q^3 + O(q^6),
q + (-2*zeta6 + 1)*q^2 + (-2*zeta6 - 1)*q^4 + (5*zeta6 - 1)*q^5 + O(q^6),
q + (zeta6 + 1)*q^2 + 3*q^3 + (zeta6 + 2)*q^4 + (-zeta6 + 5)*q^5 + O(q^6),
q^3 + O(q^6),
q + (-zeta6 - 1)*q^2 + (zeta6 + 2)*q^4 + (zeta6 - 5)*q^5 + O(q^6)
]
```

[121]:
```
M.eisenstein_subspace().T(2).matrix().fcp()
```

[121]:
```
(x + 2*zeta3 + 1) * (x + zeta3 + 2) * (x - zeta3 - 2)^2 * (x - 2*zeta3 - 1)^2
```

2.3 Cusp Forms

[14]:
```
S= CuspForms(Gamma0(4), 6)
```

[15]:
```
S
```

[15]: Cuspidal subspace of dimension 1 of Modular Forms space of dimension 4 for Congruence Subgroup Gamma0(4) of weight 6 over Rational Field

[111]:
```
ModularForms(Gamma0(27),2).eisenstein_series()
```

[111]:
```
[
q^3 + O(q^6),
q - 3*q^2 + 7*q^4 - 6*q^5 + O(q^6),
1/12 + q + 3*q^2 + q^3 + 7*q^4 + 6*q^5 + O(q^6),
1/3 + q + 3*q^2 + 4*q^3 + 7*q^4 + 6*q^5 + O(q^6),
13/12 + q + 3*q^2 + 4*q^3 + 7*q^4 + 6*q^5 + O(q^6)
]
```

[112]:
```
ModularForms(Gamma1(5),3).eisenstein_series()
```

[112]:
```
[
-1/5*zeta4 - 2/5 + q + (4*zeta4 + 1)*q^2 + (-9*zeta4 + 1)*q^3 + (4*zeta4 -
15)*q^4 + q^5 + O(q^6),
q + (zeta4 + 4)*q^2 + (-zeta4 + 9)*q^3 + (4*zeta4 + 15)*q^4 + 25*q^5 + O(q^6),
1/5*zeta4 - 2/5 + q + (-4*zeta4 + 1)*q^2 + (9*zeta4 + 1)*q^3 + (-4*zeta4 -
15)*q^4 + q^5 + O(q^6),
q + (-zeta4 + 4)*q^2 + (zeta4 + 9)*q^3 + (-4*zeta4 + 15)*q^4 + 25*q^5 + O(q^6)
]
```

[98]:
```
M=ModularForms(Gamma0(12), 6, prec=12)
```

[99]:
```
M
```

[99]: Modular Forms space of dimension 13 for Congruence Subgroup Gamma0(12) of weight 6 over Rational Field

[100]:
```
M.cuspidal_subspace().echelon_basis()
```

[100]:
```
[
q + 39*q^9 - 192*q^11 + O(q^12),
q^2 + 4*q^8 - 30*q^10 + O(q^12),
q^3 - 12*q^9 + O(q^12),
q^4 - 6*q^8 + O(q^12),
q^5 - 15*q^9 + 38*q^11 + O(q^12),
q^6 - 4*q^8 + 4*q^10 + O(q^12),
q^7 - 6*q^9 + 15*q^11 + O(q^12)
]
```

[101]:
```
M.integral_basis()
```

[101]:
```
[
1 + O(q^12),
```

```
q + O(q^12),
q^2 + O(q^12),
q^3 + O(q^12),
q^4 + O(q^12),
q^5 + O(q^12),
q^6 + O(q^12),
q^7 + O(q^12),
q^8 + O(q^12),
q^9 + O(q^12),
q^10 + O(q^12),
q^11 + O(q^12),
O(q^12)
]
```

[103]:
```
S = ModularForms(11,2).cuspidal_submodule()
```

[104]:
```
S.q_expansion_basis(12)
```

[104]:
```
[
q - 2*q^2 - q^3 + 2*q^4 + q^5 + 2*q^6 - 2*q^7 - 2*q^9 - 2*q^10 + q^11 + O(q^12)
]
```

[ ]:
```

```

2.4 Eisenstein Series

[124]:
```
E1 = EisensteinForms(25, 4)
```

[125]:
```
E.new_eisenstein_series()
```

[125]:
```
[q + (-2*zeta6 + 1)*q^2 + (-2*zeta6 - 1)*q^4 + (5*zeta6 - 1)*q^5 + O(q^6),
 q + (-zeta6 - 1)*q^2 + (zeta6 + 2)*q^4 + (zeta6 - 5)*q^5 + O(q^6)]
```

[126]:
```
from sage.modular.modform.eis_series_cython import Ek_ZZ
```

[131]:
```
Ek_ZZ(4,15)
```

[131]:
```
[1, 1, 9, 28, 73, 126, 252, 344, 585, 757, 1134, 1332, 2044, 2198, 3096]
```

[ ]:
```

```

[ ]:
```

```

2.5 Newforms

[ ]:

[ ]:

[78]:
```
Newforms(Gamma0(11), 2)
```

[78]:
```
[q - 2*q^2 - q^3 + 2*q^4 + q^5 + O(q^6)]
```

[ ]:

2.6 Hecke Operators

[114]:
```
ModularForms(17,4).basis()
```

[114]:
```
[
q + 2*q^5 + O(q^6),
q^2 - 3/2*q^5 + O(q^6),
q^3 + O(q^6),
q^4 - 1/2*q^5 + O(q^6),
1 + O(q^6),
q + 9*q^2 + 28*q^3 + 73*q^4 + 126*q^5 + O(q^6)
]
```

[113]:
```
ModularForms(17,4).hecke_polynomial(2)
```

[113]:
```
x^6 - 16*x^5 + 18*x^4 + 608*x^3 - 1371*x^2 - 4968*x + 7776
```

[122]:
```
E = ModularForms(e,2,prec=10).eisenstein_subspace()
```

[132]:
```
E.eisenstein_subspace().T(2).matrix().fcp()
```

[132]:
```
(x + 2*zeta3 + 1) * (x + zeta3 + 2) * (x - zeta3 - 2)^2 * (x - 2*zeta3 - 1)^2
```

[162]:
```
hecke_operator_on_basis(ModularForms(1,12).q_expansion_basis(30), 3, 12)
```

[162]:
```
[   252      0]
[     0 177148]
```

[163]:
```
M = ModularForms(1,12);
hecke_operator_on_qexp(M.basis()[0], 3, 12)
```

[163]:
```
252*q - 6048*q^2 + 63504*q^3 - 370944*q^4 + O(q^5)
```

[ ]:

[ ]:

2.7 Ring of Modular Forms

[ ]:

[ ]:

[136]:
```
M=ModularFormsRing(Gamma0(12))
```

[137]:
```
M.generators()
```

[137]:
```
[(2, 1 + 24*q^6 + O(q^10)),
 (2, q + 6*q^5 + 8*q^7 - 3*q^9 + O(q^10)),
 (2, q^2 + 5*q^6 - 2*q^8 + O(q^10)),
 (2, q^3 + 4*q^9 + O(q^10)),
 (2, q^4 - 2*q^6 + 3*q^8 + O(q^10))]
```

[138]:
```
M2 = ModularFormsRing(1)
```

[142]:
```
E4 = M2.0; E6=M2.1
```

[143]:
```
E4
```

[143]:
```
q + 11*q^2 + 12*q^3 + 2*q^4 + q^5 + O(q^6)
```

[144]:
```
E6
```

[144]:
```
1 + 12*q^2 + 12*q^3 + 12*q^4 + 12*q^5 + O(q^6)
```

[ ]:

2.8 Elliptic Curves

[145]:
```
E = EllipticCurve('11a')
```

[146]:
```
E
```

[146]:
```
Elliptic Curve defined by y^2 + y = x^3 - x^2 - 10*x - 20 over Rational Field
```

[153]:
```
f2 = E.modular_form()
```

[154]:
```
f2
```

[154]: q - 2*q^2 - q^3 + 2*q^4 + q^5 + O(q^6)

[155]:
```
f2.has_cm()
```

[155]: False

[156]:
```
E = EllipticCurve([-1, 0])
```

[157]:
```
E
```

[157]: Elliptic Curve defined by y^2 = x^3 - x over Rational Field

[159]:
```
f2 = E.modular_form()
```

[160]:
```
f2
```

[160]: q - 2*q^5 + O(q^6)

[161]:
```
f2.has_cm()
```

[161]: True

2.9 Modular Symbols

[164]:
```
set_modsym_print_mode ('modular');
```

[165]:
```
M = ModularSymbols(11, 2)
```

[166]:
```
M.basis()
```

[166]: ({Infinity, 0}, {-1/8, 0}, {-1/9, 0})

[223]:
```
[ModularSymbols(Gamma1(7),k).dimension() for k in [2,3,4,5]]
```

[223]: [5, 8, 12, 16]

[167]:
```
M = ModularSymbols(23,2,base_ring=QQ)
```

[168]:
```
M.T(2)
```

[168]: Hecke operator T_2 on Modular Symbols space of dimension 5 for Gamma_0(23) of weight 2 with sign 0 over Rational Field

[169]:
```
M.T(2).matrix()
```

[169]:
```
[ 3  0  0  0 -1]
[ 0  0  1 -1  0]
[ 0  0  1 -1  1]
[ 0 -1  2 -2  1]
[ 0 -1  1  0 -1]
```

[170]:
```
M.T(2).charpoly('x').factor()
```

[170]:
```
(x - 3) * (x^2 + x - 1)^2
```

[171]:
```
M1 = ModularSymbols(11, 4)
```

[172]:
```
M1.basis()
```

[172]:
```
(X^2*{0, Infinity},
 36*X^2*{-1/6, 0} + 12*X*Y*{-1/6, 0} + Y^2*{-1/6, 0},
 49*X^2*{-1/7, 0} + 14*X*Y*{-1/7, 0} + Y^2*{-1/7, 0},
 64*X^2*{-1/8, 0} + 16*X*Y*{-1/8, 0} + Y^2*{-1/8, 0},
 81*X^2*{-1/9, 0} + 18*X*Y*{-1/9, 0} + Y^2*{-1/9, 0},
 100*X^2*{-1/10, 0} + 20*X*Y*{-1/10, 0} + Y^2*{-1/10, 0})
```

[173]:
```
M.T(5).matrix()
```

[173]:
```
[ 6  0  0  0 -2]
[ 0  0  2 -2  0]
[ 0  0  2 -2  2]
[ 0 -2  4 -4  2]
[ 0 -2  2  0 -2]
```

[174]:
```
M1.cuspidal_subspace()
```

[174]:
```
Modular Symbols subspace of dimension 4 of Modular Symbols space of dimension 6
for Gamma_0(11) of weight 4 with sign 0 over Rational Field
```

[176]:
```
ModularForms(11, 4).basis()
```

[176]:
```
[
q + 3*q^3 - 6*q^4 - 7*q^5 + O(q^6),
q^2 - 4*q^3 + 2*q^4 + 8*q^5 + O(q^6),
1 + O(q^6),
q + 9*q^2 + 28*q^3 + 73*q^4 + 126*q^5 + O(q^6)
]
```

[ ]:
```
```

[177]:
```
m = ModularSymbols(Gamma1(3),12); m.dimension()
```

[177]: 8

[178]:
```
m.cuspidal_subspace().dimension()
```

[178]: 6

[179]:
```
m.cuspidal_subspace().new_subspace().dimension()
```

[179]: 2

[181]:
```
CuspForms(Gamma1(3), 12, prec=10).basis()
```

[181]:
```
[
q - 176*q^4 + 2430*q^5 - 5832*q^6 - 19336*q^7 + 101088*q^8 - 107163*q^9 +
O(q^10),
q^2 + 54*q^4 - 100*q^5 - 243*q^6 - 108*q^7 + 692*q^8 + 2916*q^9 + O(q^10),
q^3 - 24*q^6 + 252*q^9 + O(q^10)
]
```

[183]:
```
[m.cuspidal_subspace().T(n).matrix() for n in [2..10]]
```

[183]:
```
[
[     -24    236/3     50/3   -286/3       14  -623/30]  [       3411
-1364    -14965/8     -7723/8     -6639/40      -521/80]
[       0      462     -243     -243        0 -1701/20]  [          0
-58743/7     -38637/7     -38637/7            0    60507/175]
[       0     -246     -225      447      -81   1671/20]  [     -25920
9528/7      53847/7       6408/7         1242   128259/350]
[       0     -246      447     -225       81     51/20]  [      25920
49848/7     -13752/7      33687/7        -1242  -243441/350]
[       0     1800     2100    -3900      726     -690]  [     207360
-12360       -63900       -7740       -10179        -2790]
[       0     3600    -1800    -1800        0     -654],  [          0
-70800       -48600       -48600            0         2673],

[    -1472       4248        900      -5148        756    -5607/5]
[        0      24772     -13122     -13122          0  -45927/10]
[        0     -13284     -12326      24138      -4374   45117/10]
[        0     -13284      24138     -12326       4374    1377/10]
[        0      97200     113400    -210600      39028     -37260]
[        0     194400     -97200     -97200          0     -35492],

[     4830 -23600/3  -5000/3  28600/3    -1400    6230/3]  [        -81864
13620          40845          46335          2907/5          26013/5]
[        0   -43770    24300    24300        0      8505]  [             0
583146/7       1340631/7       1340631/7                0     8658333/700]
[        0    24600    24930   -44700     8100     -8355]  [        622080
```

```
189774/7      -950427/7      -914139/7         -10125 -20368287/700]
[        0     24600    -44700     24930     -8100      -255]  [       -622080
-777906/7      -430299/7      -466587/7          10125  11251413/700]
[        0  -180000  -210000    390000    -70170     69000]  [      -4976640
-140760        1023300        1133460         62046         234630]
[        0  -360000   180000    180000          0     67830], [            0
824400        1603800        1603800             0          88938],

[  -16744     -8496     -1800     10296     -1512   11214/5]
[        0    -69232     26244     26244         0   45927/5]
[        0     26568      4964    -48276      8748  -45117/5]
[        0     26568    -48276      4964     -8748   -1377/5]
[        0   -194400   -226800    421200    -97744     74520]
[        0   -388800    194400    194400         0     51296],

[     84480    163312/3     34600/3  -197912/3       9688 -215558/15]   [
682425      -114336    -845595/2  -1042533/2   -10017/10     -248787/4]
[          0      420792     -168156     -168156          0  -294273/5]   [
0      -874719     -2099520      -2099520             0 -4021893/25]
[          0     -170232      -54612      309324      -56052    289083/5]   [
-6531840      -374328      1175229      1534140        76788  8399457/25]
[          0     -170232      309324      -54612      56052      8823/5]   [
6531840      1077192       808380       449469      -76788 -4196043/25]
[          0     1245600     1453200    -2698800     603480     -477480]   [
52254720      2134080     -9979200    -13322880     -555255     -2715120]
[          0     2491200    -1245600    -1245600          0     -351480],  [
0     -7344000    -17496000    -17496000             0     -1340631],

[  -115920    -233640     -49500     283140     -41580      61677]
[         0  -1559340     721710     721710          0   505197/2]
[         0    730620     481050   -1327590     240570  -496287/2]
[         0    730620   -1327590     481050    -240570   -15147/2]
[         0  -5346000   -6237000   11583000   -2343420    2049300]
[         0 -10692000    5346000    5346000          0    1755180]
]
```

[184]:
```
ModularSymbols(2, 8)[1].q_eigenform(5, 'a')
```

[184]:
```
q - 8*q^2 + 12*q^3 + 64*q^4 + O(q^5)
```

[185]:
```
M = ModularSymbols(1, 24).cuspidal_submodule()
```

[190]:
```
M.q_expansion_module(9, QQ)
```

[190]:
```
Vector space of degree 9 and dimension 2 over Rational Field
Basis matrix:
[           0            1            0       195660     12080128     44656110
```

```
-982499328  -147247240 22106234880]
[           0           0           1          -48         1080       -15040
143820      -985824     4857920]
```

[186]:
```
M.q_expansion_basis(8, algorithm='eigen')
```

[186]:
```
[
q + 195660*q^3 + 12080128*q^4 + 44656110*q^5 - 982499328*q^6 - 147247240*q^7 +
O(q^8),
q^2 - 48*q^3 + 1080*q^4 - 15040*q^5 + 143820*q^6 - 985824*q^7 + O(q^8)
]
```

[187]:
```
CuspForms(1, 24).basis()
```

[187]:
```
[
q + 195660*q^3 + 12080128*q^4 + 44656110*q^5 + O(q^6),
q^2 - 48*q^3 + 1080*q^4 - 15040*q^5 + O(q^6)
]
```

[ ]:

[192]:
```
M = ModularSymbols(11,4,base_ring=GF(2)); M.basis()
```

[192]:
```
(X*Y*{Infinity, 0},
 X*Y*{-1/8, 0},
 X^2*{-1/9, 0} + X*Y*{-1/9, 0},
 X^2*{0, Infinity},
 Y^2*{-1/8, 0},
 X^2*{-1/9, 0} + Y^2*{-1/9, 0},
 Y^2*{-1/10, 0})
```

[ ]:

[197]:
```
M = ModularSymbols(11, 2)
```

[198]:
```
M
```

[198]:
```
Modular Symbols space of dimension 3 for Gamma_0(11) of weight 2 with sign 0
over Rational Field
```

[199]:
```
M.basis()
```

[199]:
```
({Infinity, 0}, {-1/8, 0}, {-1/9, 0})
```

[228]:
```
M.cuspidal_submodule().basis()
```

[228]:
```
()
```

[200]:
```
M.modular_symbol([2/11, oo])
```

[200]:
```
-{-1/9, 0}
```

[203]:
```
M.modular_symbol([0, -2/5, 0])
```

[203]:
```
-{-1/8, 0} + 2*{-1/9, 0}
```

[ ]:
```

```

[229]:
```
M = ModularSymbols(8, 4)
```

[230]:
```
M.basis()
```

[230]:
```
(X^2*{0, Infinity},
 36*X^2*{-1/6, 0} + 12*X*Y*{-1/6, 0} + Y^2*{-1/6, 0},
 49*X^2*{-1/7, 0} + 14*X*Y*{-1/7, 0} + Y^2*{-1/7, 0},
 X^2*{0, 1/2},
 9*X^2*{1/3, 1/2} - 6*X*Y*{1/3, 1/2} + Y^2*{1/3, 1/2},
 X^2*{0, 1/4})
```

[231]:
```
M.cuspidal_submodule().basis()
```

[231]:
```
(36*X^2*{-1/6, 0} + 12*X*Y*{-1/6, 0} + Y^2*{-1/6, 0} - 49*X^2*{-1/7, 0} -
14*X*Y*{-1/7, 0} - Y^2*{-1/7, 0},
 X^2*{0, 1/2} - 9*X^2*{1/3, 1/2} + 6*X*Y*{1/3, 1/2} - Y^2*{1/3, 1/2})
```

[ ]:
```

```

[232]:
```
M.modular_symbol([0, -2/5, 0])
```

[232]:
```
-576*X^2*{-1/6, 0} - 192*X*Y*{-1/6, 0} - 16*Y^2*{-1/6, 0} - 392*X^2*{-1/7, 0} -
112*X*Y*{-1/7, 0} - 8*Y^2*{-1/7, 0} + 88*X^2*{0, 1/2} - 792*X^2*{1/3, 1/2} +
528*X*Y*{1/3, 1/2} - 88*Y^2*{1/3, 1/2}
```

[233]:
```
M.modular_symbol([2, -2/5, 0])
```

[233]:
```
-72*X^2*{-1/6, 0} - 24*X*Y*{-1/6, 0} - 2*Y^2*{-1/6, 0} - 98*X^2*{-1/7, 0} -
28*X*Y*{-1/7, 0} - 2*Y^2*{-1/7, 0} + 14*X^2*{0, 1/2} - 126*X^2*{1/3, 1/2} +
84*X*Y*{1/3, 1/2} - 14*Y^2*{1/3, 1/2}
```

[234]:
```
set_modsym_print_mode('manin')
```

[235]:
```
M.modular_symbol([2, -2/5, 0])
```

[235]:
```
-2*[X^2,(1,6)] - 2*[X^2,(1,7)] + 14*[X^2,(2,1)] - 14*[X^2,(2,3)]
```

[236]:
```
M.modular_symbol([0, -2/5, 0])
```

[236]: -16*[X^2,(1,6)] - 8*[X^2,(1,7)] + 88*[X^2,(2,1)] - 88*[X^2,(2,3)]

[237]:
```
set_modsym_print_mode('modular')
```

[238]:
```
M.modular_symbol([2, -2/5, 0])
```

[238]: -72*X^2*{-1/6, 0} - 24*X*Y*{-1/6, 0} - 2*Y^2*{-1/6, 0} - 98*X^2*{-1/7, 0} - 28*X*Y*{-1/7, 0} - 2*Y^2*{-1/7, 0} + 14*X^2*{0, 1/2} - 126*X^2*{1/3, 1/2} + 84*X*Y*{1/3, 1/2} - 14*Y^2*{1/3, 1/2}

[ ]:

2.10 Manin Symbols

[239]:
```
from sage.modular.modsym.manin_symbol import ManinSymbol;
from sage.modular.modsym.manin_symbol_list import ManinSymbolList_gamma0
```

[ ]:

[224]:
```
A=ModularSymbols(11,2).manin_symbols()
```

[225]:
```
A
```

[225]: Manin Symbol List of weight 2 for Gamma0(11)

[226]:
```
len(A)
```

[226]: 12

[194]:
```
ModularSymbols(11,2).manin_symbols_basis()
```

[194]: [(1,0), (1,8), (1,9)]

[ ]:

[195]:
```
ModularSymbols(11,6).manin_symbols()
```

[195]: Manin Symbol List of weight 6 for Gamma0(11)

[196]:
```
ModularSymbols(11,6).manin_symbols_basis()
```

[196]:
```
[[X^4,(0,1)],
 [X^4,(1,2)],
 [X^4,(1,3)],
```

```
[X^4,(1,4)],
[X^4,(1,5)],
[X^4,(1,6)],
[X^4,(1,7)],
[X^4,(1,8)],
[X^4,(1,9)],
[X^4,(1,10)]]
```

[ ]: