Modular Forms Project: Eta Products

Henri Cohen,
Université de Bordeaux, Institut de Mathématiques de Bordeaux,
351 Cours de la Libération, 33405 TALENCE Cedex, FRANCE

July 8, 2018

1 Answers and Programs

Exercise 1.

(1). If $24 = a_1 + a_2 + \ldots + a_r$ is a partition of 24, we let I to be the set of distinct integers a_i, and for each $m \in I$ we let $r_m > 0$ be the number of a_i equal to m. It is clear that this is a bijection between the partitions of 24 and $\mathcal{E}$.

(2). The command `numbpart(24)` tells us that $\mathcal{E}$ has 1575 elements.

Exercise 2.

(1). In `Pari/GP` the simplest is to use the second optional argument to the `partitions` command: simply write `vector(24,k,[k/2,#partitions(24,,[k,k])])` whose output will be

[[1/2, 1], [1, 12], [3/2, 48], [2, 108], [5/2, 164], [3, 199], [7/2, 201], [4, 186], [9/2, 157], [5, 128], [11/2, 99], [6, 77], [13/2, 56], [7, 42], [15/2, 30], [8, 22], [17/2, 15], [9, 11], [19/2, 7], [10, 5], [21/2, 3], [11, 2], [23/2, 1], [12, 1]]

(2). Assume that `V` is a partition (as a `Vecsmall`) corresponding to an eta product of integral weight. To test whether it is multiplicative for $mn \leq 100$ we can write for instance the following:

```
/* Transform eta product Vecsmall into matrix (not
   essential, but more legible). */
VStomat(V)=
{
  my(S=Set(V),M=Mat([]),m,rm);
  for(i=1,#S,
    m=S[i]; rm=0;
    for(j=1,#V,if(V[j]==m,rm++));
    M=concat(M,Mat([m,rm]~))
  );
  return (M~);
}

/* Compute expansion of eta product given by Vecsmall
   or matrix to L terms. */
etaprod(V,L)=
```

```
{
  my(P=x,T,V1,V2);
  if (type(V)=="t_VECSMALL",V=VStomat(V));
  V1=V[,1];V2=V[,2];
  for (i=1,#V1,P*=eta(x^V1[i]+O(x^(L+1)))^V2[i]);
  P;
}

/* Optional parameter L, default 100 */
tsthecke(V,L=100)=
{
  my(E,P,Pn);
  P=etaprod(V,L);
  for(n=2,sqrtint(L),
    Pn=polcoeff(P,n);
    for(m=n+1,L\n,
      if(gcd(m,n)==1,
        if (polcoeff(P,m)*Pn!=polcoeff(P,m*n), return(0))
      )
    )
  );
  return (1);
}
```

To answer the question, we write the following program:

```
dohecke()=
{
  my(VH1=[]);
  forpart(V=24,
    if ((#V)%2==0 && tsthecke(V),VH1=concat(VH1,[VStomat(V)]));
  );
  for(i=1,#VH1,
    print(i,": ",VH1[i],", ",mfparams(mffrometaquo(VH1[i]))[1..3])
  );
  return (VH1);
}
```

Thus, executing the command `VH1=dohecke();` outputs:

```
1: [1, 1; 23, 1], [23, 1, -23]
2: [2, 1; 22, 1], [44, 1, -11]
3: [3, 1; 21, 1], [63, 1, -7]
4: [4, 1; 20, 1], [80, 1, -20]
5: [6, 1; 18, 1], [108, 1, -3]
6: [8, 1; 16, 1], [128, 1, -8]
7: Mat([12, 2]), [144, 1, -4]
8: [1, 2; 11, 2], [11, 2, 1]
9: [1, 1; 2, 1; 7, 1; 14, 1], [14, 2, 1]
10: [1, 1; 3, 1; 5, 1; 15, 1], [15, 2, 1]
```

```
11: [2, 2; 10, 2], [20, 2, 1]
12: [2, 1; 4, 1; 6, 1; 12, 1], [24, 2, 1]
13: [3, 2; 9, 2], [27, 2, 1]
14: [4, 2; 8, 2], [32, 2, 1]
15: Mat([6, 4]), [36, 2, 1]
16: [1, 3; 7, 3], [7, 3, -7]
17: [1, 2; 2, 1; 4, 1; 8, 2], [8, 3, -8]
18: [2, 3; 6, 3], [12, 3, -3]
19: Mat([4, 6]), [16, 3, -4]
20: [1, 4; 5, 4], [5, 4, 1]
21: [1, 2; 2, 2; 3, 2; 6, 2], [6, 4, 1]
22: [2, 4; 4, 4], [8, 4, 1]
23: Mat([3, 8]), [9, 4, 1]
24: [1, 4; 2, 2; 4, 4], [4, 5, -4]
25: [1, 6; 3, 6], [3, 6, 1]
26: Mat([2, 12]), [4, 6, 1]
27: [1, 8; 2, 8], [2, 8, 1]
28: Mat([1, 24]), [1, 12, 1]
```

We thus see that there are 28 potential Hecke eigenforms, 7 in weight 1, 8 in weight 2 (all with trivial character, and thus corresponding to elliptic curves over $\mathbb{Q}$ of respective conductors 11, 14, 15, 20, 24, 27, 32, and 36, which are in fact unique up to isgogeny), 4 in weight 3, 4 in weight 4, 1 in weight 5, 2 in weight 6, 1 in weight 8, and 1 in weight 12.

Exercise 3.

To test multiplicativity in half-integral weight we can write the following:

```
tsthecke2(V,L)=
{
  P=etaprod(V,L);
  forsquarefree(tt=1,30,
    t=tt[1]; at=polcoeff(P,t);
    if (at,
      limn=sqrtint(sqrtint(L\t));
      for (n=2,limn,
        Pn=polcoeff(P,t*n^2);
        for(m=n+1,sqrtint(L\(t*n^2)),
          if(gcd(m,n)==1,
            if (polcoeff(P,t*m^2)*Pn!=at*polcoeff(P,t*m^2*n^2), return(0))
          )
        )
      ),
      for (n=2,sqrtint(L\t),if(polcoeff(P,t*n^2),return(0)))
    )
  );
  return (1);
}
```

Even though the exercise asks us to test the condition only up to $tm^2n^2 \leq 1000$, it is safer to write the following:

```
dohecke2()=
{
  my(VH2=[]);
  forpart(V=24,
    if((#V)%2==1,
      if(tsthecke2(V,1000),
        if(tsthecke2(V,3000),
          if(tsthecke2(V,10000),
            VH2=concat(VH2,[VStomat(V)])
          )
        )
      )
    )
  );
  for(i=1,#VH2,
    print(i,": ",VH2[i],", ",mfparams(mffrometaquo(VH2[i]))[1..3])
  );
  return (VH2);
}
```

This program first tests if the multiplicativity condition is satisfied with $tm^2n^2 \leq 1000$, and only if the answer is positive, it tests up to 3000, and then up to 10000. In fact, 1000 is sufficient (i.e., the test up to 10000 is superfluous), but we do not know this in advance; also, we have not *proved* that multiplicativity always holds.

Thus, executing the command `VH2=dohecke2();` outputs:

```
1: Mat([24, 1]), [576, 1/2, 12]
2: [1, 2; 22, 1], [176, 3/2, 44]
3: [2, 2; 20, 1], [160, 3/2, 40]
4: [2, 1; 4, 1; 18, 1], [864, 3/2, 8]
5: [2, 1; 8, 1; 14, 1], [448, 3/2, 28]
6: [2, 1; 11, 2], [176, 3/2, 1]
7: [3, 2; 18, 1], [432, 3/2, 1]
8: [3, 1; 9, 1; 12, 1], [864, 3/2, 8]
9: [4, 2; 16, 1], [128, 3/2, 8]
10: [4, 1; 8, 1; 12, 1], [576, 3/2, 12]
11: [4, 1; 10, 2], [160, 3/2, 8]
12: [6, 2; 12, 1], [288, 3/2, 24]
13: [6, 1; 9, 2], [432, 3/2, 12]
14: Mat([8, 3]), [64, 3/2, 1]
15: [1, 3; 6, 1; 15, 1], [720, 5/2, 5]
16: [1, 2; 4, 2; 14, 1], [28, 5/2, 28]
17: [2, 4; 16, 1], [128, 5/2, 8]
18: [2, 3; 4, 1; 14, 1], [224, 5/2, 56]
19: [2, 3; 6, 1; 12, 1], [96, 5/2, 8]
20: [2, 2; 4, 2; 12, 1], [288, 5/2, 24]
21: [2, 2; 4, 1; 8, 2], [16, 5/2, 8]
```

```
22: [2, 1; 4, 1; 6, 3], [96, 5/2, 24]
23: [3, 4; 12, 1], [288, 5/2, 24]
24: [3, 3; 6, 1; 9, 1], [432, 5/2, 1]
25: [3, 2; 6, 3], [144, 5/2, 12]
26: [4, 4; 8, 1], [64, 5/2, 1]
27: [4, 3; 6, 2], [288, 5/2, 8]
28: [1, 3; 3, 1; 6, 3], [144, 7/2, 1]
29: [1, 2; 2, 2; 6, 3], [48, 7/2, 12]
30: [2, 4; 4, 2; 8, 1], [64, 7/2, 1]
31: [2, 3; 3, 2; 6, 2], [48, 7/2, 1]
32: [2, 2; 4, 5], [32, 7/2, 8]
33: [3, 6; 6, 1], [144, 7/2, 12]
34: [1, 6; 6, 3], [48, 9/2, 12]
35: [1, 4; 4, 5], [32, 9/2, 8]
36: [1, 3; 2, 3; 5, 3], [80, 9/2, 5]
37: [1, 2; 2, 3; 4, 4], [16, 9/2, 1]
38: [2, 8; 8, 1], [64, 9/2, 1]
39: [2, 6; 4, 3], [32, 9/2, 8]
40: [2, 3; 3, 6], [48, 9/2, 1]
41: [1, 2; 2, 7; 4, 2], [4, 11/2, 1]
42: [2, 10; 4, 1], [32, 11/2, 8]
43: [1, 5; 2, 5; 3, 3], [48, 13/2, 12]
44: [1, 2; 2, 11], [16, 13/2, 1]
45: [1, 6; 2, 9], [16, 15/2, 1]
```

Here, we thus have 45 (possible) Hecke eigenforms, more precisely $(1, 13, 13, 6, 7, 2, 2, 1)$ in weights $(1/2, 3/2, 5/2, 7/2, 9/2, 11/2, 13/2, 15/2)$.

Exercise 4.

(1). We write the following program:

```
tstshimura(V,L=50)=
{
  my(P,F,mf,N,k,s,mf2,G,S,ad,D,SD);
  P=etaprod(V,L); F=mffrometaquo(V); mf=mfinit(F,1);
  [N,k]=mfparams(F);
  if (k==1/2,return([0,P,mf,F,0,0]));
  if((k-1/2)%2,s=-1,s=1);
  [mf2,G]=mfshimura(mf,F,1);
  S=mfcoefs(G,L);
  for(Da=3,L,
    ad=polcoeff(P,Da);
    D=s*Da;
    if(isfundamental(D) && gcd(N,D)==1,
      SD=mfcoefs(mfshimura(mf,F,D)[2],L);
      if (SD!=ad*S,return([0,P,mf,F,mf2,G]))
    )
  );
  return([S,P,mf,F,mf2,G]);
}
```

In this program we not only return a yes or no answer, but also a number of other quantities that can be use in other programs without being recomputed. Note that we exclude weight 1/2 since the Shimura lift as explained above is not applicable in that case.

To obtain the indices that correspond to eta products having proportional shimura lifts, we simply write the following

```
/* VH2 is the output of dohecke2 */
doshimura(VH2)=
{
  my(VS=[]);
  for(i=2,#VH2,
    if (tstshimura(VH2[i])[1],VS=concat(VS,i))
  );
  return (VS);
}
```

Thus, executing the command `VS=doshimura(VH2)` outputs (after 1 minute or so) the 28 corresponding indices:

[3, 9, 10, 11, 12, 14, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 28, 30, 32, 34, 35, 37, 38, 39, 41, 42, 43, 45]

(2). We do exactly the same (but simpler) for the Kohnen condition:

```
tstkohnen(V,L=100)=
{
  my(P=etaprod(V,L));
  forstep(i=2,L,[1,3],if(polcoeff(P,i),return(0)));
  return(1);
}

/* VH2 is the output of dohecke2(). */
dokohnen(VH2)=
{
  my(VK=[],r);
  for(i=1,#VH2,
    if(tstkohnen(VH2[i]),VK=concat(VK,i))
  );
  return(VK);
}
```

Thus, executing the command `VS=dokohnen(VH2)` outputs

```
[1, 9, 10, 14, 26]
```

Exercise 5.

For this exercise, we can write the following program:

```
tstwaldspurger(V,L=50)=
{
  my(S,P,mf,F,mf2,G,N,k,s,LG,ad2,D,LT,VW=[]);
```

```
  [S,P,mf,F,mf2,G]=tstshimura(V,L);
  if (!S,return(0));
  [N,k]=mfparams(F); if((k-1/2)%2,s=-1,s=1);
  LG=lfunmf(mf2,G);
  for(Da=3,L,
    ad2=polcoeff(P,Da)^2;
    if (ad2,
      D=s*Da;
      if(isfundamental(D) && gcd(N,D)==1,
        LT=lfuntwist(LG,D);
        VW=concat(VW,Da^(k-1)*lfun(LT,k-1/2)/ad2)
      )
    )
  );
  if(#VW==0||vecmin(VW)<10^(-10),return(0));
  if(abs(vecmax(VW)/vecmin(VW)-1)<10^(-10),
    return (vecmax(VW)),
    return (0)
  );
}
```

A few comments: first, it is of course necessary to redo the computation of the `tstshimura`, so as to have available the accessory data such as `mf`, `F`, `mf2`, `G`, etc... Second, because of the stringent conditions that we included for the discriminants, it may happen that `VW` is empty, and the `vecmin` and `vecmax` functions do not work in that case. Finally, it may also happen that `vecmin(VW)` is very small, and we exclude this since either there is a loss of accuracy, or the Waldspurger condition is not satisfied.

Note that twisting by a large discriminant D multiplies the level by D^2, which can become quite expensive.

As usual, to obtain the required indices we simply write:

```
dowaldspurger(VH2)=
{
  my(VW=[],r);
  for(i=1,#VH2,
    if(tstwaldspurger(VH2[i]),VW=concat(VW,i))
  );
  return(VW);
}
```

Thus, executing the command `VW=dowaldspurger(VH2)` outputs

```
[26, 30, 37, 38, 41, 45]
```

Note that in the test we have not added the condition that $F \in S_k(\Gamma_0(N))$ with no character, but this is indeed the case for the six eta products that we have found.

For fun, if we also want to see the constant given by Waldspurger's theorem, we can also write:

```
for(i=1,#VW,ind=VW[i];print(ind,": ",tstwaldspurger(VH2[ind])));
```

which outputs:

```
26: 1.1136967437017574953710011825926461770
30: 2.5781528542418427075113039314054421657
37: 1.3625850619310736674744619518461820898
38: 1.5853344174521043236770250724804121170
41: 2.6502916177466143161549593439949418366
45: 6.9437451200064545322567536626723148945
```

Exercise 6.

We see that the index 26 is the only one occurring in all our searches, and the corresponding eta product has matrix $\left(\begin{smallmatrix}4&4\\8&1\end{smallmatrix}\right)$ which corresponds to $F(\tau) = \eta(4\tau)^4\eta(8\tau)$.

(1). We write the following:

```
F=mffrometaquo([4,4;8,1]);mf=mfinit(F,1);
mfparams(F)
Ser(mfcoefs(F,30),q)
```

which outputs the parameters `[64,5/2,1,y]`, which says that $F \in S_{5/2}(\Gamma_0(64))$, and then the expansion

```
q - 4*q^5 + q^9 + 12*q^13 - 8*q^17 - 8*q^21 - 7*q^25 + 4*q^29 + O(q^31)
```

(2). We simply write

```
[mf2,G]=mfshimura(mf,F,1); mfparams(G)
Ser(mfcoefs(G,20),q)
```

which outputs the parameters `[16,4,1,y]`, which says that $G \in S_4(\Gamma_0(16))$ as predicted by Kohnen's theorem, and then the expansion

```
q + 4*q^3 - 2*q^5 - 24*q^7 - 11*q^9 + 44*q^11 + 22*q^13 - 8*q^15\
  + 50*q^17 - 44*q^19 + O(q^21)
```

(3). We have already computed above that the Waldspurger constant is approximately equal to

```
1.1136967437017574953710011825926461770
```

(4.) Using the given commands, we find that

$< G, G > = 8.2179771764814246623334306747451700813E - 5$ and

$< F, F > = 0.00072827889789335095595678337078492247121$ which gives

$$\pi^2 \frac{< G, G >}{< F, F >} = 1.1136967437017574953710011825926461779\ .$$

Exercise 7.

We could write specific commands for solving this exercise, but it is more useful to write a general `GP` script which tests whether a given modular form given in `Pari/GP` format is an eta quotient or not, as follows:

```
mfisetaquo(F)=
{
  [N,k]=mfparams(F);
  S=Ser(mfcoefs(F,N+10)); /* +10 for safety */
  vS=valuation(S,x); S/=x^vS;
  R=vector(N); /* R will hold the r_m's */
  for(m=1,N,
    rm=-polcoeff(S,m); R[m]=rm; S/=eta(x^m+O(x^(N+10)))^rm;
  );
/* Here our potential eta quotient is
  $\prod_{1\le m\le N}\eta(m\tau)^{rm}$. */
/* First do sanity checks. */
  weight=vecsum(R)/2; if(weight!=k,return(0));
  val=sum(m=1,N,m*R[m])/24; if(val!=vS,return(0));
/* Now complete check. */
  M=Mat([]);
  for(m=1,N,if(R[m],M=concat(M,Mat([m,R[m]]~))));
  M=M~; /* matrix of eta quotient. */
  G=mffrometaquo(M);
  if (!mfisequal(F,G),return(0));
  return (M);
}
```

Note that this simple-minded script could be improved in several ways, but will be sufficient for us. We thus write `mfisetaquo(G)`, which answers:

```
[2 -4]

[4 16]

[8 -4]
```

This in fact *proves* that

$$G(\tau) = \frac{\eta(4\tau)^{16}}{\eta(2\tau)^4\eta(8\tau)^4}\ .$$

If we do not want to rely on `Pari/GP`'s command `mfisequal`, we note that both sides belong to $S_4(\Gamma_0(16))$, and the Sturm bound (in `Pari/GP` simply `mfsturm([16,4])`) tells us that if the first 9 Fourier coefficients are equal, they are equal, and this is the case since by construction the first 16 are equal.