

Supplement to:

“Old Meets New: Connecting Two  
Infinite Families of Congruences

Modulo Powers of 5 for Generalized  
Frobenius Partition Functions,”

Frank G. Garvan, University of Florida

James A. Sellers, University of  
Minnesota Duluth

Nicolas Allen Smoot, University of  
Vienna

---

Some Remarks Regarding the functions  $x$ ,  $y$ ,  $t$

We start with some preliminary definitions. We give explicit expressions for  $x$ ,  $y$ , and  $t$ . We also give the expression for  $t$  in terms of  $x$ , which we will prove in the section “Level-Reduction Analysis” below, and the expression for  $z_{i=1}/(1+5x)$ , which we proved in Section 3 of the main paper:

$$\ln[6] := Xq = \frac{q \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^{10}, q^{10}]^3}{\text{QPochhammer}[q, q]^3 \text{QPochhammer}[q^5, q^5]};$$

$$\ln[6] := Yq = \left( q^2 \text{QPochhammer}[q^2, q^2]^2 \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^5, q^5] \text{QPochhammer}[q^{20}, q^{20}]^3 \right) / \left( \text{QPochhammer}[q, q]^5 \text{QPochhammer}[q^{10}, q^{10}]^2 \right);$$

$$\ln[6] := Tq = \frac{q \text{QPochhammer}[q^5, q^5]^6}{\text{QPochhammer}[q, q]^6};$$

$$\ln[6] := tR = \frac{x + 8x^2 + 16x^3}{1 + 5x};$$

$$\ln[6] := zi = 1 - 125t + 120x + 400x^2;$$

We give an expression for our function  $A^{(0)}$  defined in our text. We have

$$U_0^{(0)}(f) = U_5(A^{(0)}f),$$

$$U_1^{(0)}(f) = U_5(f).$$

Here, for  $a=0,1$  and  $b=0,1$ , we denote  $p_a^{(b)}$  as  $p_a^{(b)}$  expressed in terms of  $t, x, y$  (the latter considered as indeterminates).

$$\ln[6] := A0q = \left( \text{QPochhammer}[q^4, q^4]^2 \text{QPochhammer}[q^{25}, q^{25}]^2 \text{QPochhammer}[q^{50}, q^{50}] \right) / \left( q^4 \text{QPochhammer}[q, q]^2 \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^{100}, q^{100}]^2 \right);$$

$$\ln[6] := p00n = 1 + 5t + 64x + 600tx + 208x^2 + 2000tx^2 + 4y + 1000ty + 12500t^2y - 992xy - 12000txy - 3520x^2y - 40000tx^2y;$$

$$\ln[6] := p01n = 1 + 9t + 8x + 64tx + 16x^2 + 80tx^2 + 4y + 8ty + 500t^2y + 32xy - 480txy + 64x^2y - 1600tx^2y;$$

We denote  $P_{ab}$ , as  $p_a^{(b)}$  expressed in terms of  $t, x, y$  (the latter taken as their  $q$ -Pochhammer quotients).

$$\ln[6] := P00 = p00n / . t \rightarrow Tq / . x \rightarrow Xq / . y \rightarrow Yq;$$

```
In[ ]:= P01 = p01n /. t -> Tq /. x -> Xq /. y -> Yq;
```

The module generators used by Paule and Radu have the following form:

```
In[ ]:= p11n = t + 25 t^2 - 8 t x - 4 y - 8 t y - 500 t^2 y - 32 x y + 480 t x y - 64 x^2 y + 1600 t x^2 y;
```

```
In[ ]:= p10n = 85 t + 625 t^2 - 24 x - 64 x^2 - 4 y -
          1000 t y - 12500 t^2 y + 992 x y + 12000 t x y + 3520 x^2 y + 40000 t x^2 y;
```

If we take  $p_a^{(b)} + p_a^{(1-b)}$ , we get the following (notice that the contribution of  $y$  vanishes altogether):

```
In[ ]:= p01n + p11n // Expand
```

```
Out[ ]:= 1 + 10 t + 25 t^2 + 8 x + 56 t x + 16 x^2 + 80 t x^2
```

```
In[ ]:= p00n + p10n // Expand
```

```
Out[ ]:= 1 + 90 t + 625 t^2 + 40 x + 600 t x + 144 x^2 + 2000 t x^2
```

## Cusp Calculations for $X_0(N)$

We want to examine  $x$  and show that it is modular over  $X_0(10)$  (indeed, that it is a Hauptmodul at  $[0]$ ).

We also want to derive an expression of  $t$  in terms of  $x$ .

We provide here a set of representatives for the cusps of each of the two surfaces  $X_0(20)$  and  $X_0(100)$ :

```
In[ ]:= cusps20 =
```

```
{
  1 / 20, 1 / 10, 1 / 5, 1 / 4, 1 / 2, 1 / 1
};
```

```
In[ ]:= cusps100 =
```

```
{
  1 / 100, 1 / 50, 1 / 25, 1 / 20, 1 / 10, 3 / 20,
  1 / 5, 1 / 4, 3 / 10, 7 / 20, 2 / 5, 9 / 20,
  1 / 2, 3 / 5, 7 / 10, 4 / 5, 9 / 10, 1 / 1
};
```

We next give a formula which counts the order of a modular eta quotient at a given cusp:

```
In[ ]:= etaCuspOrder[c_, N_, s_] :=
  Sum[
    (N / (24 c GCD[c, N / c])) s[[i]]  $\frac{\text{GCD}[\text{Divisors}[N][[i]], c]^2}{\text{Divisors}[N][[i]}}$ ,
    {i, 1, Length[Divisors[N]]}
  ]
```

Notice that in this formula, a given eta quotient  $s$  is represented by a vector. One constructs this vector by the powers of each eta function  $\eta(\delta\tau)$  in which  $\delta$  is a divisor of  $N$ .

For example, one can show that our function  $A$  is modular over  $X_0(100)$ .

```
In[ ]:= A0q
Out[ ]:=  $\frac{\text{QPochhammer}[q^4, q^4]^2 \text{QPochhammer}[q^{25}, q^{25}]^2 \text{QPochhammer}[q^{50}, q^{50}]}{q^4 \text{QPochhammer}[q, q]^2 \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^{100}, q^{100}]^2}$ 
```

To it we assign the vector

```
In[ ]:= A0v = {-2, -1, 2, 0, 0, 0, 2, 1, -2};
```

We assign similar corresponding vectors to our functions for  $x$ ,  $y$ , and  $t$ . Indeed, we assign two in these cases---one corresponding to  $X_0(20)$  and one corresponding to  $X_0(100)$ .

`In[ ]:= Xq`

$$\text{Out[ ]} = \frac{q \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^{10}, q^{10}]^3}{\text{QPochhammer}[q, q]^3 \text{QPochhammer}[q^5, q^5]}$$

`In[ ]:= xv = {-3, 1, 0, -1, 3, 0};`

`In[ ]:= xv100 = {-3, 1, 0, -1, 3, 0, 0, 0, 0};`

`In[ ]:= Yq`

$$\text{Out[ ]} = \frac{\left( q^2 \text{QPochhammer}[q^2, q^2]^2 \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^5, q^5] \text{QPochhammer}[q^{20}, q^{20}]^3 \right)}{\left( \text{QPochhammer}[q, q]^5 \text{QPochhammer}[q^{10}, q^{10}]^2 \right)}$$

`In[ ]:= yv = {-5, 2, 1, 1, -2, 3};`

`In[ ]:= yv100 = {-5, 2, 1, 1, -2, 3, 0, 0, 0};`

`In[ ]:= Tq`

$$\text{Out[ ]} = \frac{q \text{QPochhammer}[q^5, q^5]^6}{\text{QPochhammer}[q, q]^6}$$

`In[ ]:= tv = {-6, 0, 0, 6, 0, 0};`

`In[ ]:= tv100 = {-6, 0, 0, 6, 0, 0, 0, 0, 0};`

We can now examine the orders of x, y, and t at a given cusp:

```
In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, xv]
    },
    {k, 1, Length[cusps20]}
  ]
]
{ 1/20, 1 }
{ 1/10, 1 }
{ 1/5, 0 }
{ 1/4, 0 }
{ 1/2, 0 }
{ 1, -2 }
```

```

In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, yv]
    },
    {k, 1, Length[cusps20]}
  ]
]

```

Out[ ]:=

```

{
  {1/20, 2}
  {1/10, 0}
  {1/5, 0}
  {1/4, 1}
  {1/2, 0}
  {1, -3}
}

```

```

In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, tv]
    },
    {k, 1, Length[cusps20]}
  ]
]

```

Out[ ]:=

```

{
  {1/20, 1}
  {1/10, 1}
  {1/5, 4}
  {1/4, -1}
  {1/2, -1}
  {1, -4}
}

```

It is useful to define a function which has positive order at every cusp except  $[1/20]$  (equivalently,  $[\infty]$ ):

$$\text{In[ ]:= } \mu\theta = \frac{\text{QPochhammer}[q^4, q^4]^4 \text{QPochhammer}[q^{10}, q^{10}]^8}{q^6 \text{QPochhammer}[q^{20}, q^{20}]^{12}};$$

```

In[ ]:= mu0v = {0, 0, 4, 0, 8, -12};

```

```

In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, mu0v]
    },
    {k, 1, Length[cusps20]}
  ]
]

```

```

Out[ ]:=
{
  {1/20, -6},
  {1/10, 1},
  {1/5, 1},
  {1/4, 2},
  {1/2, 1},
  {1, 1}
}

```

```

In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, {-5, 5, 0, 1, -1, 0}],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, xv]
    },
    {k, 1, Length[cusps20]}
  ]
]

```

```

Out[ ]:=
{
  {1/20, 0, 1},
  {1/10, 0, 1},
  {1/5, 0, 0},
  {1/4, 1, 0},
  {1/2, 1, 0},
  {1, -2, -2}
}

```

Therefore, the function which we denote by  $z$ , or  $Zq$  in code,

$$\text{In[ ]:= } Zq = \frac{QPochhammer[q^2, q^2]^5 QPochhammer[q^5, q^5]}{QPochhammer[q, q]^5 QPochhammer[q^{10}, q^{10}]} ;$$

must be a scalar multiple of  $x$  plus some complex number. We quickly find that

```
In[ ]:= Series[
  Zq - (1 + 5 Xq),
  {q, 0, 100}
]
Out[ ]:= 0[q]^101
```

Now we compare the orders of  $z$  with those of  $t$ :

```
In[ ]:= Column[
  Table[
    {
      cusps20[[k]],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, {-5, 5, 0, 1, -1, 0}],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, tv],
      etaCuspOrder[Denominator[cusps20[[k]]], 20, 6 mu0v]
    },
    {k, 1, Length[cusps20]}
  ]
]
Out[ ]:= {1/20, 0, 1, -36}
          {1/10, 0, 1, 6}
          {1/5, 0, 4, 6}
          {1/4, 1, -1, 12}
          {1/2, 1, -1, 6}
          {1, -2, -4, 6}
```

So  $z$  must be some combination of powers of  $x$  and  $y$  by powers of  $x$ . One guesses from



```
In[ ]:= Series[
  Zq Tq - (Xq + 8 Xq^2 + 16 Xq^3),
  {q, 0, 100}
]
Out[ ]:= O[q]^101
```

that  $t$  must have the form we defined above:

```
In[ ]:= tR
Out[ ]:= 
$$\frac{x + 8x^2 + 16x^3}{1 + 5x}$$

```

We prove this by multiplying by a sufficient power of  $\mu_0$  to push everything up to the cusp at  $[1/20]$ , and checking to see that the principal parts and constants are indeed killed:

```
In[ ]:= Series[
  (mu0^6) Zq Tq - (mu0^6) (Xq + 8 Xq^2 + 16 Xq^3),
  {q, 0, 100}
]
Out[ ]:= O[q]^101
```

---

## Cusp Analysis for Witness Identity in Section 4.1

We proceed in a manner related to the previous section of this notebook. First we give the representative cusps for the weight 60 classical modular curve:

```
In[ ]:= cusps60 =
  {
    1 / 60, 1 / 30, 1 / 20, 1 / 15, 1 / 12, 1 / 10,
    1 / 6, 1 / 5, 1 / 4, 1 / 3, 1 / 2, 1 / 1
  };
```

Next we give the vector representations and the eta quotient representations of the functions p1, p2, p3 and t that are found in the identity:

$$\text{In[ ]:= } \mathbf{p1v} = \{-1, -1, -2, 1, 6, 2, -2, -1, -1, -1, -1, 1\};$$

$$\text{In[ ]:= } \mathbf{p2v} = \{-1, -1, -2, 1, 1, 2, -1, -1, -2, -1, 4, 1\};$$

$$\text{In[ ]:= } \mathbf{p3v} = \{0, -1, 1, 1, -2, 2, 1, -1, 1, -1, -2, 1\};$$

$$\text{In[ ]:= } \mathbf{t60v} = \{-6, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0\};$$

$$\text{In[ ]:= } \mathbf{p1} = \left( \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^5, q^5]^6 \text{QPochhammer}[q^6, q^6]^2 \text{QPochhammer}[q^{60}, q^{60}] \right) / \\ \left( \text{QPochhammer}[q, q] \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^3, q^3]^2 \text{QPochhammer}[q^{10}, q^{10}]^2 \right. \\ \left. \text{QPochhammer}[q^{12}, q^{12}] \text{QPochhammer}[q^{15}, q^{15}] \text{QPochhammer}[q^{20}, q^{20}] \text{QPochhammer}[q^{30}, q^{30}] \right);$$

$$\text{In[ ]:= } \mathbf{p2} = \left( q^5 \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^5, q^5] \right. \\ \left. \text{QPochhammer}[q^6, q^6]^2 \text{QPochhammer}[q^{30}, q^{30}]^4 \text{QPochhammer}[q^{60}, q^{60}] \right) / \\ \left( \text{QPochhammer}[q, q] \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^3, q^3]^2 \text{QPochhammer}[q^{10}, q^{10}] \right. \\ \left. \text{QPochhammer}[q^{12}, q^{12}] \text{QPochhammer}[q^{15}, q^{15}]^2 \text{QPochhammer}[q^{20}, q^{20}] \right);$$

$$\text{In[ ]:= } \mathbf{p3} = \left( \text{QPochhammer}[q^3, q^3] \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^6, q^6]^2 \right. \\ \left. \text{QPochhammer}[q^{10}, q^{10}] \text{QPochhammer}[q^{15}, q^{15}] \text{QPochhammer}[q^{60}, q^{60}] \right) / \\ \left( \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^5, q^5]^2 \text{QPochhammer}[q^{12}, q^{12}] \right. \\ \left. \text{QPochhammer}[q^{20}, q^{20}] \text{QPochhammer}[q^{30}, q^{30}]^2 \right);$$

$$\text{In[ ]:= } \mathbf{Tq}$$

$$\text{Out[ ]:= } \frac{q \text{QPochhammer}[q^5, q^5]^6}{\text{QPochhammer}[q, q]^6}$$

We compute the orders of the constituent products of functions on the right-hand side of the identity. Note that t is multiplied by all other terms:

```

In[ ]:= Column[
  Table[
    {
      cusps60[[k]],
      etaCuspOrder[Denominator[cusps60[[k]]], 60, p1v + t60v],
      etaCuspOrder[Denominator[cusps60[[k]]], 60, p2v + t60v],
      etaCuspOrder[Denominator[cusps60[[k]]], 60, p3v + t60v],
      etaCuspOrder[Denominator[cusps60[[k]]], 60, t60v]
    },
    {k, 1, Length[cusps60]}
  ]
]

```

Out[ ]:=

```

{
  {1/60, 1, 6, 1, 1},
  {1/30, 0, 5, 0, 1},
  {1/20, 2, 2, 1, 3},
  {1/15, 4, 4, 4, 4},
  {1/12, -2, -1, 0, -1},
  {1/10, 3, 3, 2, 3},
  {1/6, -1, 0, 1, -1},
  {1/5, 22, 12, 8, 12},
  {1/4, -3, -3, -2, -3},
  {1/3, -8, -8, 0, -4},
  {1/2, -4, -4, -3, -3},
  {1, -14, -16, -12, -12}
}

```

We want to push these functions to the cusp at infinity, thus simplifying our calculations. The highest possible order pole is -16. Consider the following eta quotient:

```
In[ ]:= mu60v = {0, 1, 0, -1, 0, -1, -1, 3, 0, 3, 5, -9};
```

```
In[ ]:= mu60 =
```

$$\frac{\left( \text{QPochhammer}[q^2, q^2] \text{QPochhammer}[q^{12}, q^{12}]^3 \text{QPochhammer}[q^{20}, q^{20}]^3 \text{QPochhammer}[q^{30}, q^{30}]^5 \right)}{\left( q^{13} \text{QPochhammer}[q^4, q^4] \text{QPochhammer}[q^6, q^6] \text{QPochhammer}[q^{10}, q^{10}] \text{QPochhammer}[q^{60}, q^{60}]^9 \right)};$$

If we examine its 16th power, we find that it can overpower any poles on the right-hand side of our identity aside from infinity. Notice that the induced pole at infinity will have order -208:

```

In[ ]:= Column[
  Table[
    {
      cusps60[[k]],
      etaCuspOrder[Denominator[cusps60[[k]]], 60, 16 mu60v]
    },
    {k, 1, Length[cusps60]}
  ]
]
{ 1/60, -208 }
{ 1/30, 16 }
{ 1/20, 16 }
{ 1/15, 16 }
{ 1/12, 48 }
Out[ ]:= { 1/10, 16 }
{ 1/6, 16 }
{ 1/5, 16 }
{ 1/4, 16 }
{ 1/3, 16 }
{ 1/2, 16 }
{ 1, 16 }

```

We now consider the left-hand side of our identity. First we define the generating function:

```

In[ ]:= phi2 = (QPochhammer[q^4, q^4] QPochhammer[q^6, q^6]^2) /
  (QPochhammer[q, q] QPochhammer[q^2, q^2] QPochhammer[q^3, q^3] QPochhammer[q^12, q^12]);

```

Anticipating that we will multiply on by our 16th power of mu60, and that we will therefore need 208 terms to compensate for, we will expand this function to take the first  $208 \cdot 5 = 1040$  terms. Indeed, we take a few more to adjust for a shift in the progression:

```
In[6]:= phi2S = Series[phi2, {q, 0, 1045}];
```

We have an inverted and dilated version of the generating function as a prefactor on the left-hand side. Let us define that factor:

```
In[7]:= phi2q5 = (QPochhammer[q^20, q^20] QPochhammer[q^30, q^30]^2) /
  (QPochhammer[q^5, q^5] QPochhammer[q^10, q^10] QPochhammer[q^15, q^15] QPochhammer[q^60, q^60]);
```

We now use a key theorem to provide lower bounds for poles of an eta quotient modular function in which a linear progression is taken.

The theorem is Theorem 47 taken from Silviu Radu's paper, "An Algorithmic Approach to Ramanujan--Kolberg Identities," published in the Journal of Symbolic Computation, Volume 68, pp. 225-253 (2015).

We define the functions p and Pstar (Here the term Pmr is simply 1).

```
In[3]:= p[y_, M_, r_, m_] :=
  Min[
    Table[
      1
      Sum[
        24
          r[[i]] (
            GCD[
              Divisors[M][[i]]
              (Numerator[y] + GCD[m^2 - 1, 24] Denominator[y]), m Denominator[y]
            ] ^ 2 / (Divisors[M][[i]] m)
          ),
      {i, 1, Length[Divisors[M]]}
    ],
    {1, 0, m - 1}
  ]
```

```
In[4]:= pStar[y_, N_, s_] :=
```

$$\frac{1}{24} \text{Sum} \left[ s[[i]] \frac{\text{GCD}[\text{Divisors}[N][[i]], \text{Denominator}[y]]^2}{\text{Divisors}[N][[i]]}, \{i, 1, \text{Length}[\text{Divisors}[N]]\} \right]$$

```
In[6]:= Column[
```

```
Table[
```

```
{
```

```
cusps60[[k]],
```

$$N \left[ \frac{60}{\text{GCD}[(\text{Denominator}[\text{cusps60}[[k]])]^2, 60]} \right. \\ \left. (\text{p}[\text{repMat}[60][[k]], 12, \{-1, -1, -1, 1, 2, -1\}, 5] + \text{pStar}[\text{repMat}[60][[k]], 60, \{0, 0, 0, 0, 1, 0, 1, 0, 1, -1, -2, 1\}]) \right]$$

```
},
```

```
{k, 1, Length[repMat[60]]}
```

```
]
```

```
]
```

$$\left\{ \frac{1}{60}, 0.4 \right\}$$

$$\left\{ \frac{1}{30}, -0.8 \right\}$$

$$\left\{ \frac{1}{20}, -0.4 \right\}$$

$$\left\{ \frac{1}{15}, 1.6 \right\}$$

$$\left\{ \frac{1}{12}, -2. \right\}$$

```
Out[6]=
```

$$\left\{ \frac{1}{10}, 0.8 \right\}$$

$$\left\{ \frac{1}{6}, 0. \right\}$$

$$\left\{ \frac{1}{5}, 3.2 \right\}$$

$$\left\{ \frac{1}{4}, 0. \right\}$$

$$\left\{ \frac{1}{3}, -8. \right\}$$

$$\left\{ \frac{1}{2}, -4. \right\}$$

$$\{1, -16. \}$$

Once again, the highest order pole to overcome is -16. Therefore, multiplying on by the 16th power of mu60 will induce a function with a pole only at infinity.

We now take the left-hand side of our identity and multiply on by this power of  $\mu_{60}$ , and take the full principal part together with the constant term (and even the first positive order term):

```
In[ ]:= Lhs60 =
Series[
(mu60^(16)) ((1 / phi25) Sum[Coefficient[phi25, q, 5 n + 3] q^(n + 1), {n, 0, 208}]),
{q, 0, 1}
];
```

We do the same with the right-hand side, being careful to multiply by the same power of  $\mu_{60}$ :

```
In[ ]:= Rhs60 =
Series[
(mu60^(16)) (5 Tq (-1 + 5 p1 + 60 p2 - 3 p3)),
{q, 0, 1}
];
```

Finally we subtract:

```
In[ ]:= Lhs60 - Rhs60
Out[ ]:= 0 [q]^2
```

The difference of these two modular functions is holomorphic with no poles at all, and thus must be a constant. Since we have accounted for the constants, the difference must be 0, and the identity is proved.

---

## Computations for the Atkin--Lehner Involution

Here we will walk through the computations needed to prove Theorem 5.2. We will only consider the

computations for  $c\psi_{2,0}(n)$ , since Paule and Radu had already considered  $c\psi_{2,1}(n)$ . In any case, the latter relations may be demonstrated in an analogous manner.

First, we note that, as discussed in Section 3, our relevant functions may be built as elements of a rank 6  $\mathbb{Z}[t]$  module with generators  $1, x, x^2, y, xy, x^2y$ . As such, we build a function which will take any polynomial in  $x$  and reduce it so that all powers of  $x$  are 2 or less.

```
In[6]:= tReduce[f_] :=
Module[
{
fn = f,
j = Exponent[f, x]
},
While[
j ≥ 3,
Set[
fn, Sum[x^k Coefficient[fn, x, k], {k, 0, 2}] + Sum[x^(k-3) (
 $\frac{1}{16} (t + 5xt - x - 8x^2)$ 
)
Coefficient[fn, x, k], {k, 3, Exponent[fn, x]}] // Expand
];
Set[j, Exponent[fn, x]]
];
fn
]
```

Similarly, we will reduce any power of  $y$  greater than 2 to a polynomial in  $x$  and powers of  $t$  in which the power of  $y$  is either 0 or 1.



```

In[ ]:= yReduce[f_] :=
Module[
{
fn = f,
j = Exponent[f, y]
},
While[
j ≥ 2,
Set[
fn,
Sum[y^k Coefficient[fn, y, k], {k, 0, 1}] + Sum[y^(k - 2) (
 $\frac{1}{4} (x^2 + 5 x^3 - y - 6 x y)$ 
) Coefficient[fn, y, k], {k, 2, Exponent[fn, y]}] // Expand
];
Set[j, Exponent[fn, y]]
];
fn
]

```

We showed in Theorem 5.1 that  $x, y, z=1+5x$ , all have certain transformations via our chosen Atkin-Lehner involution. We can build up the formula of Theorem 5.1 using the effects of the involution on  $x, 1/(1+5x)$ , and  $y$ .

Notice that by  $ziw$ , we mean the effect of the involution on  $1/(1+5x)$ .

$$\text{In[ ]:= } \mathbf{xw} = \frac{1 + 10x + 20x^2 + 4y}{(1 + 5x)^2};$$

$$\text{In[ ]:= } \mathbf{ziw} = \frac{1 + 5x + 5y}{1 + 5x};$$

$$\text{In[ ]:= } \mathbf{yw} = \frac{1 + 14x + 40x^2 + (20 + 80x)y}{(1 + 5x)^3};$$

$$\text{In[ ]:= } \mathbf{AL}[m_, n_, j_] := 4^{n-2j-m} (-1)^{m+n} xw^m ziw^n yw^j$$

We want to convert this into how the involution affects polynomials in  $x, y, t$ .

For example, if we take  $xw$  and recall that we have  $z_i=1/(1+5x)$ , then we have

`In[ ]:= (1 + 10 x + 20 x^2 + 4 y) zi ^2 // Expand`

`Out[ ]:= 1 - 250 t + 15 625 t^2 + 250 x - 32 500 t x + 156 250 t^2 x + 17 620 x^2 - 405 000 t x^2 + 312 500 t^2 x^2 + 252 800 x^3 - 1 600 000 t x^3 + 1 424 000 x^4 - 2 000 000 t x^4 + 3 520 000 x^5 + 3 200 000 x^6 + 4 y - 1000 t y + 62 500 t^2 y + 960 x y - 120 000 t x y + 60 800 x^2 y - 400 000 t x^2 y + 384 000 x^3 y + 640 000 x^4 y`

This expression contains powers of  $x$  which are greater than 2. So we apply `tReduce`. This is the effect of the involution on  $x$  (not yet counting associated powers of -1 and 2).

`In[ ]:= tReduce [(1 + 10 x + 20 x^2 + 4 y) zi ^2 // Expand]`

`Out[ ]:= 1 - 200 t - 3125 t^2 + 200 x + 3000 t x + 720 x^2 + 10 000 t x^2 + 4 y + 3000 t y + 62 500 t^2 y - 3040 x y - 60 000 t x y - 11 200 x^2 y - 200 000 t x^2 y`

`In[ ]:= xW = 1 - 200 t - 3125 t^2 + 200 x + 3000 t x + 720 x^2 + 10 000 t x^2 + 4 y + 3000 t y + 62 500 t^2 y - 3040 x y - 60 000 t x y - 11 200 x^2 y - 200 000 t x^2 y;`

We can get the effect of the involution on  $1/(1+5x)$  in a similar manner:

`In[ ]:= tReduce [(1 + 5 x + 5 y) zi // Expand]`

`Out[ ]:= 1 + 5 y - 625 t y + 600 x y + 2000 x^2 y`

`In[ ]:= ziW = 1 + 5 y - 625 t y + 600 x y + 2000 x^2 y;`

Finally, we do the same with  $y$ :

`In[ ]:= tReduce [(1 + 14 x + 40 x^2 + (20 + 80 x) y) zi ^3 // Expand]`

`Out[ ]:= 1 + 375 t + 34 375 t^2 + 390 625 t^3 - 376 x - 34 000 t x - 375 000 t^2 x - 1520 x^2 - 120 000 t x^2 - 1 250 000 t^2 x^2 + 20 y - 5500 t y - 562 500 t^2 y - 7 812 500 t^3 y + 5280 x y + 560 000 t x y + 7 500 000 t^2 x y + 20 800 x^2 y + 2 000 000 t x^2 y + 25 000 000 t^2 x^2 y`

```
In[ ]:= yW = 1 + 375 t + 34 375 t2 + 390 625 t3 - 376 x - 34 000 t x - 375 000 t2 x - 1520 x2 -
120 000 t x2 - 1 250 000 t2 x2 + 20 y - 5500 t y - 562 500 t2 y - 7 812 500 t3 y + 5280 x y +
560 000 t x y + 7 500 000 t2 x y + 20 800 x2 y + 2 000 000 t x2 y + 25 000 000 t2 x2 y;
```

There are various ways of getting the effect on t (including simple direct computation). One way is to examine how we can express t in terms of x:

```
In[ ]:= tR
Out[ ]:= 
$$\frac{x + 8 x^2 + 16 x^3}{1 + 5 x}$$

```

We now apply our formula AL[m,n,j] to each rational monomial in this expression:

```
In[ ]:= AL [1, 1, 0] + 8 AL [2, 1, 0] + 16 AL [3, 1, 0] // Together
Out[ ]:= 
$$\frac{(1 + 10 x + 20 x^2 + 4 y) (1 + 5 x + 5 y) (25 x^4 - 40 x^2 y + 16 y^2)}{(1 + 5 x)^7}$$

```

Multiplying this out will induce large powers of x and y, together with a denominator in x. So instead, we replace 1/(1+5x) with zi, then we apply yReduce and tReduce. This gives us the effect of the involution on t:

```
In[ ]:= tReduce [yReduce [(1 + 10 x + 20 x2 + 4 y) (1 + 5 x + 5 y) (25 x4 - 40 x2 y + 16 y2) zi ^7]]
Out[ ]:= 136 t + 14 000 t2 + 312 500 t3 + 1 953 125 t4 - 136 x - 13 880 t x - 305 000 t2 x -
1 875 000 t3 x - 528 x2 - 50 000 t x2 - 1 050 000 t2 x2 - 6 250 000 t3 x2 - 1980 t y -
225 000 t2 y - 5 625 000 t3 y - 39 062 500 t4 y + 1984 x y + 223 200 t x y + 5 500 000 t2 x y +
37 500 000 t3 x y + 7680 x2 y + 808 000 t x2 y + 19 000 000 t2 x2 y + 125 000 000 t3 x2 y

In[ ]:= tW = 136 t + 14 000 t2 + 312 500 t3 + 1 953 125 t4 - 136 x - 13 880 t x - 305 000 t2 x -
1 875 000 t3 x - 528 x2 - 50 000 t x2 - 1 050 000 t2 x2 - 6 250 000 t3 x2 - 1980 t y -
225 000 t2 y - 5 625 000 t3 y - 39 062 500 t4 y + 1984 x y + 223 200 t x y + 5 500 000 t2 x y +
37 500 000 t3 x y + 7680 x2 y + 808 000 t x2 y + 19 000 000 t2 x2 y + 125 000 000 t3 x2 y;
```

So now we have an alternative expression for how the involution affects a polynomial in  $x, t, y$ :

```
In[ ]:= ALt[m_, n_, j_] :=
  4^(-2 j - m) (-1)^(m) tReduce[yReduce[xW^m tW^n yW^j]]
```

## $c\psi_{2,0}(n)$ : Odd Index, Progression $4n+2$

We will use ALt to study the effect of the involution (henceforth called  $W$ ) on the arithmetic progressions of  $c\psi_{2,0}(n)$ . As shown in Section 5, we only need consider  $p_1^{(0)}$ , and the necessary behavior for  $p_0^{(0)}$  will follow.

We begin by reminding ourselves of  $p_1^{(0)}$ :

```
In[ ]:= p01n
Out[ ]:= 1 + 9 t + 8 x + 64 t x + 16 x^2 + 80 t x^2 + 4 y + 8 t y + 500 t^2 y + 32 x y - 480 t x y + 64 x^2 y - 1600 t x^2 y
```

We apply  $W$  to  $p_1^{(0)}$  and note its form:

```
In[ ]:= p01w =
  ALt[0, 0, 0] + 9 ALt[0, 1, 0] + 8 ALt[1, 0, 0] + 64 ALt[1, 1, 0] +
  16 ALt[2, 0, 0] + 80 ALt[2, 1, 0] + 4 ALt[0, 0, 1] + 8 ALt[0, 1, 1] + 500 ALt[0, 2, 1] +
  32 ALt[1, 0, 1] - 480 ALt[1, 1, 1] + 64 ALt[2, 0, 1] - 1600 ALt[2, 1, 1] // Together
Out[ ]:= 5200 t + 2 674 400 t^2 + 320 787 500 t^3 + 15 859 765 625 t^4 + 393 457 031 250 t^5 + 5 200 195 312 500 t^6 +
  35 095 214 843 750 t^7 + 95 367 431 640 625 t^8 - 5200 x - 2 669 488 t x - 318 698 000 t^2 x -
  15 660 250 000 t^3 x - 385 890 625 000 t^4 x - 5 064 453 125 000 t^5 x - 33 935 546 875 000 t^6 x -
  91 552 734 375 000 t^7 x - 20 512 x^2 - 10 132 640 t x^2 - 1 170 740 000 t^2 x^2 - 56 027 500 000 t^3 x^2 -
  1 350 781 250 000 t^4 x^2 - 17 402 343 750 000 t^5 x^2 - 114 746 093 750 000 t^6 x^2 - 305 175 781 250 000 t^7 x^2 -
  75 964 t y - 39 945 500 t^2 y - 5 031 000 000 t^3 y - 262 117 187 500 t^4 y - 6 843 750 000 000 t^5 y -
  94 970 703 125 000 t^6 y - 671 386 718 750 000 t^7 y - 1 907 348 632 812 500 t^8 y + 75 968 x y +
  39 873 760 t x y + 4 999 640 000 t^2 x y + 258 945 000 000 t^3 x y + 6 715 937 500 000 t^4 x y +
  92 539 062 500 000 t^5 x y + 649 414 062 500 000 t^6 x y + 1 831 054 687 500 000 t^7 x y + 299 520 x^2 y +
  151 489 600 t x^2 y + 18 398 000 000 t^2 x^2 y + 928 150 000 000 t^3 x^2 y + 23 546 875 000 000 t^4 x^2 y +
  318 359 375 000 000 t^5 x^2 y + 2 197 265 625 000 000 t^6 x^2 y + 6 103 515 625 000 000 t^7 x^2 y
```

We construct the list of monomials in this expression:

```
In[ ]:= mLP01w =
      MonomialList[p01w]
Out[ ]:= {-1907348632812500 t8 y, 95367431640625 t8, 6103515625000000 t7 x2 y,
-305175781250000 t7 x2, 1831054687500000 t7 x y, -91552734375000 t7 x,
-671386718750000 t7 y, 35095214843750 t7, 2197265625000000 t6 x2 y,
-114746093750000 t6 x2, 649414062500000 t6 x y, -33935546875000 t6 x,
-94970703125000 t6 y, 5200195312500 t6, 318359375000000 t5 x2 y, -17402343750000 t5 x2,
92539062500000 t5 x y, -5064453125000 t5 x, -684375000000 t5 y, 393457031250 t5,
23546875000000 t4 x2 y, -1350781250000 t4 x2, 6715937500000 t4 x y, -385890625000 t4 x,
-262117187500 t4 y, 15859765625 t4, 928150000000 t3 x2 y, -56027500000 t3 x2,
258945000000 t3 x y, -15660250000 t3 x, -5031000000 t3 y, 320787500 t3,
18398000000 t2 x2 y, -1170740000 t2 x2, 4999640000 t2 x y, -318698000 t2 x,
-39945500 t2 y, 2674400 t2, 151489600 t x2 y, -10132640 t x2, 39873760 t x y,
-2669488 t x, -75964 t y, 5200 t, 299520 x2 y, -20512 x2, 75968 x y, -5200 x}
```

We also construct the list of coefficients:

```
In[ ]:= mLP01wC = {-1907348632812500, 95367431640625, 6103515625000000, -305175781250000,
1831054687500000, -915527343750000, -671386718750000, 35095214843750,
2197265625000000, -114746093750000, 649414062500000, -33935546875000,
-94970703125000, 5200195312500, 318359375000000, -17402343750000,
92539062500000, -5064453125000, -684375000000, 393457031250, 23546875000000,
-1350781250000, 6715937500000, -385890625000, -262117187500, 15859765625,
928150000000, -56027500000, 258945000000, -15660250000, -5031000000, 320787500,
18398000000, -1170740000, 4999640000, -318698000, -39945500, 2674400, 151489600,
-10132640, 39873760, -2669488, -75964, 5200, 299520, -20512, 75968, -5200};
```

Of course, we have already examined  $t, y, x$  by considering their associated vectors, which we remind the reader of here:

```
In[ ]:= tv
Out[ ]:= {-6, 0, 0, 6, 0, 0}
```

```
In[e]:= yv
```

```
Out[e]= { -5, 2, 1, 1, -2, 3 }
```

```
In[e]:= xv
```

```
Out[e]= { -3, 1, 0, -1, 3, 0 }
```

Now we are interested in the arithmetic progression  $4n+2$ . The functions  $t$ ,  $x$ , each begin with  $q^1$ , and  $y$  begins with  $q^2$ . So the associated progression  $4n+2$  in a given monomial with powers of  $t$ ,  $x$ ,  $y$  can be given as

```
In[e]:= rem2[nt_, nx_, ny_] :=  

Mod[2 - (nt + nx + 2 ny), 4]
```

With  $nt$ ,  $nx$ ,  $ny$  denoting the powers of  $t$ ,  $x$ ,  $y$  (respectively) in the monomial. Thus, `rem2` will give us the key progression to examine mod 4 in the given associated infinite products for  $t$ ,  $x$ ,  $y$ .

Now we must appeal to software that has already been developed to study arithmetic progressions in eta quotients. The user will need the software `RaduRK` to explicitly work through the following steps (though we will illustrate them as we go).

```
<< RaduRK
```

```
math4ti2: Mathematica interface to 4ti2 (http://www.4ti2.de/)  

Copyright (C) 2017, Ralf Hemmecke <ralf@hemmecke.org>  

Copyright (C) 2017, Silviu Radu <sradu@risc.jku.at>
```

```
RaduRK: Ramanujan--Kolberg Program Version 2.8  

Copyright (C) 2020, Nicolas Allen Smoot <nicolas.smoot@risc.jku.at>  

Research Institute for Symbolic Computation  

Johannes Kepler Universität, Linz
```

The first thing we will do is determine the associated modular curve that we will have to work on in examining the progression  $4n+2$ . We build a modified version of `minN` for this purpose. Notice that we

do not just have the power of  $t$ ,  $x$ , and  $y$  in the given monomial, but also the presence of  $\phi(-q^5)$  (here indicated by the vector  $(0,0,0,2,-1,0)$  indexed over the powers of 20).

```
In[6]:= MN2[f_] :=
  minN[
    20,
    Exponent[f, t] tv + Exponent[f, x] xv + Exponent[f, y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem2[Exponent[f, t], Exponent[f, x], Exponent[f, y]]
  ]
```

Now if we examine the necessary level of the modular curve for each monomial, we have a fortunate pattern:

```
In[6]:= Column[
  Table[
    {j, MN2[mLp01w[[j]]]},
    {j, 1, Length[mLp01w]}
  ]
]
```

```
{1, 40}  
{2, 40}  
{3, 40}  
{4, 40}  
{5, 40}  
{6, 40}  
{7, 40}  
{8, 40}  
{9, 40}  
{10, 40}  
{11, 40}  
{12, 40}  
{13, 40}  
{14, 40}  
{15, 40}  
{16, 40}  
{17, 40}  
{18, 40}  
{19, 40}  
{20, 40}  
{21, 40}  
{22, 40}  
{23, 40}  
Out[*]= {24, 40}  
{25, 40}  
{26, 40}  
{27, 40}  
{28, 40}  
{29, 40}  
{30, 40}  
{31, 40}  
{32, 40}  
{33, 40}  
{34, 40}  
{35, 40}  
{36, 40}  
{37, 40}  
{38, 40}  
{39, 40}  
{40, 40}  
{41, 40}  
{42, 40}  
{43, 40}  
{44, 40}  
{45, 40}  
{46, 40}  
{47, 40}  
{48, 40}
```



We will therefore work over the modular curve  $X_0(40)$ . This has genus 3 (this can be computed through formulae in Diamond--Shurman, or through consulting OEIS sequence A001617). As such, we can build an algebra basis for the functions over this curve with a pole only at  $[\infty]$  with functions which have orders -4, -5, -6, -7 at  $[\infty]$ . Again appealing to RaduRK, we can use the algebra commands AB and etaGenerators to build the following algebra basis (which may be checked directly through the etaCuspOrder function we defined in the previous section):

```

In[ ]:= ab40 =
{
  QPochhammer[q^4, q^4]^3 QPochhammer[q^20, q^20]
  /
  q^4 QPochhammer[q^8, q^8] QPochhammer[q^40, q^40]^3,
{
  1,
  (QPochhammer[q^2, q^2]^3 QPochhammer[q^5, q^5] QPochhammer[q^20, q^20]^2) /
  (q^5 QPochhammer[q, q] QPochhammer[q^10, q^10] QPochhammer[q^40, q^40]^4),
  QPochhammer[q^8, q^8] QPochhammer[q^10, q^10]^5
  /
  q^6 QPochhammer[q^2, q^2] QPochhammer[q^40, q^40]^5,
  (QPochhammer[q^4, q^4]^4 QPochhammer[q^20, q^20]^8) /
  (q^7 QPochhammer[q^2, q^2]^2 QPochhammer[q^10, q^10]^2 QPochhammer[q^40, q^40]^8)
}
};

```

The associated Ramanujan--Kolberg identity for taking the associated progression may consist of multiple different progressions, which would substantially complicate our approach. We can show that this will not be a problem by building a modified Pmr function:

```

In[6]:= Pmr2[f_] :=
  Pmr [
    40,
    20,
    Exponent[f, t] tv + Exponent[f, x] xv + Exponent[f, y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem2[Exponent[f, t], Exponent[f, x], Exponent[f, y]]
  ]

```

If we examine every monomial we have with its associated progression, we find that we will never have multiple different progressions, since the length of Pmr2 is always just 1:

```

In[6]:= Column[
  Table[
    {j, Length[Pmr2[mLp01w[[j]]]}},
    {j, 1, Length[mLp01w]}
  ]
]

```

```
{1, 1}
{2, 1}
{3, 1}
{4, 1}
{5, 1}
{6, 1}
{7, 1}
{8, 1}
{9, 1}
{10, 1}
{11, 1}
{12, 1}
{13, 1}
{14, 1}
{15, 1}
{16, 1}
{17, 1}
{18, 1}
{19, 1}
{20, 1}
{21, 1}
{22, 1}
{23, 1}
Out[*]= {24, 1}
{25, 1}
{26, 1}
{27, 1}
{28, 1}
{29, 1}
{30, 1}
{31, 1}
{32, 1}
{33, 1}
{34, 1}
{35, 1}
{36, 1}
{37, 1}
{38, 1}
{39, 1}
{40, 1}
{41, 1}
{42, 1}
{43, 1}
{44, 1}
{45, 1}
{46, 1}
{47, 1}
{48, 1}
```

We will have to multiply the generating function for each associated progression by some eta quotient over the divisors of 40. We give the associated system of equations here (which may be checked by comparing to Theorem 45 of Radu's paper "An Algorithmic Approach to Ramanujan--Kolberg Identities").

The function `sSys[N_,M_,r_,m_,j_,PM_]` gives the associated system of equations that need to be solved. `N` here will be 40 (the level of our modular curve). `M` will be 20 (taking an eta quotient over the divisors of which our monomial in `t, x, y`, will be expressed), and `r` is the associated vector over the divisors of 20. Here `m` will be 4 and `j` will be the associated `rem2` progression mod 4. `PM` will just be the set `{rem2}`. This is taken from key steps in the buildup of a given Ramanujan--Kolberg identity.

```

In[ ]:= w[r_] :=
  Sum[
    r[[d]],
    {d, 1, Length[r]}
  ]

In[ ]:= sigI[r_, N_] :=
  Sum[
    Divisors[N][[d]] × r[[d]],
    {d, 1, Length[Divisors[N]]}
  ]

In[ ]:= sig0[r_, N_] :=
  Sum[
    
$$\frac{N}{\text{Divisors}[N][[d]]} r[[d]],$$

    {d, 1, Length[Divisors[N]]}
  ]

v[N_, M_, r_, m_, j2_, PM_] :=
  Sum[
    
$$\frac{1}{m} (1 - m^2) (24 \text{PM}[[j]] + \text{sigI}[r, M]),$$

    {j, 1, Length[PM]}
  ]

In[ ]:= primeF[N_] :=
  Table[
    FactorInteger[N][[j]][[1]],
    {j, 1, Length[FactorInteger[N]]}
  ]

```

```

In[ ]:= pp[N_, i_, d_] :=
  If[
    Mod[IntegerExponent[d, primeF[N][[i]]], 2] == 1,
    1,
    0
  ]

prodS[N_, M_, r_, m_, j2_, PM_] :=
  Product[
    (m Divisors[M][[i]])^(Abs[r[[i]]]), {i, 1, Length[Divisors[M]]}^(Length[PM])
  ]

sSys[N_, M_, r_, m_, j2_, PM_] :=
  Join[
    {
      Sum[s[j], {j, 1, Length[Divisors[N]]}] == -Length[PM] × w[r],
      Sum[Divisors[N][[j]] × s[j], {j, 1, Length[Divisors[N]]}] + 24 mu11 ==
        Mod[-v[N, M, r, m, j2, PM] - Length[PM] m sigI[r, M], 24],
      Sum[ $\frac{N}{\text{Divisors}[N][[j]]} s[j], \{j, \text{Length}[\text{Divisors}[N]]\}] + 24 \mu_{22} ==$ 
        Mod[-Length[PM]  $\frac{m N}{M}$  sig0[r, M], 24]
    },
    Table[Sum[pp[NM, i, Divisors[N][[j]]] × s[j], {j, 1, Length[Divisors[N]]}] + 2 ep[i] ==
      pp[NM, i, prodS[N, M, r, m, j2, PM]], {i, 1, Length[primeF[NM]]}]
  ]

```

More good news: we will have to multiply the generating function for each associated progression by some eta quotient over the divisors of 40. But we can show that a single eta quotient will suffice for every single monomial that we have. To do this, we check whether the system of equations for our first monomial matches all of the others:

```

In[ ]:= Table[
  sSys[
    40,
    20,
    Exponent[mLp01w[[1]], t] tv + Exponent[mLp01w[[1]], x] xv +
      Exponent[mLp01w[[1]], y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem2[Exponent[mLp01w[[1]], t], Exponent[mLp01w[[1]], x], Exponent[mLp01w[[1]], y]],
    {rem2[Exponent[mLp01w[[1]], t], Exponent[mLp01w[[1]], x], Exponent[mLp01w[[1]], y]]}
  ] = sSys[
    40,
    20,
    Exponent[mLp01w[[j1]], t] tv + Exponent[mLp01w[[j1]], x] xv +
      Exponent[mLp01w[[j1]], y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem2[Exponent[mLp01w[[j1]], t],
      Exponent[mLp01w[[j1]], x], Exponent[mLp01w[[j1]], y]],
    {rem2[Exponent[mLp01w[[j1]], t], Exponent[mLp01w[[j1]], x],
      Exponent[mLp01w[[j1]], y]]}
  ],
  {j1, 1, Length[mLp01w]}
]

Out[ ]:= {True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True}

```

One may solve this system of equations, together with the inequalities in Theorem 47 of Radu's paper "An Algorithmic Approach to Ramanujan--Kolberg Identities," to get an eta quotient which will induce nonnegative order at every cusp except  $[\infty]$ . For the sake of simplicity we simply give one such eta quotient, given by the vector

(219, -93, -16, -45, 44, 18, 128, -256) indexed over the divisors of 40. We define the function

```

pStar[y_, N_, s_] :=
  
$$\frac{1}{24} \text{Sum}\left[ \frac{\text{GCD}[\text{Divisors}[N][[i]], y[[2]][[1]]]^2}{\text{Divisors}[N][[i]]}, \{i, 1, \text{Length}[\text{Divisors}[N]]\} \right]$$


```

again in keeping with Theorem 47 of Radu's paper "An Algorithmic Approach to Ramanujan--Kolberg Identities." We then put together the associated lower bound on the order of the associated generating function (keeping track of the progression  $4n+2$ ) for each monomial, with our normalizing eta quotient. Notice that the orders are all only at  $[\infty]$ :

```

In[*]:= Column[
  Table[
    Table[
      N[
        
$$\frac{40}{\text{GCD}[(\text{repMat}[40][[11]][[2]][[1]])^2, 40]} ($$

        p[repMat[40][[11]], 20, Exponent[mLp01w[[j1]], t] tv + Exponent[mLp01w[[j1]], x] xv +
        Exponent[mLp01w[[j1]], y] yv + {0, 0, 0, 2, -1, 0}, 4] +
        pStar[repMat[40][[11]], 40, {219, -93, -16, -45, 44, 18, 128, -256}]
      )
    ],
    {11, 1, Length[repMat[40]]}
  ],
  {j1, 1, Length[mLp01w]}
]

```

```

{-306., 3., 4.5, 0.75, 18., 9.75, 12., 0.}
{-306.5, 2.5, 3.5, 0.5, 14., 9.5, 11.5, 24.}
{-305.75, 3.25, 5., 1., 20., 10., 12.5, 0.}
{-306.25, 2.75, 4., 0.75, 16., 9.75, 12., 24.}
{-306., 3., 4.5, 1., 18., 10., 12.5, 16.}
{-306.5, 2.5, 3.5, 0.75, 14., 9.75, 12., 40.}
{-306.25, 2.75, 4., 1., 16., 10., 12.5, 32.}
{-306.75, 2.25, 3., 0.75, 12., 9.75, 12., 56.}
{-306., 3., 4.5, 1.25, 18., 10.25, 13., 32.}
{-306.5, 2.5, 3.5, 1., 14., 10., 12.5, 56.}
{-306.25, 2.75, 4., 1.25, 16., 10.25, 13., 48.}
{-306.75, 2.25, 3., 1., 12., 10., 12.5, 72.}
{-306.5, 2.5, 3.5, 1.25, 14., 10.25, 13., 64.}
{-307., 2., 2.5, 1., 10., 10., 12.5, 88.}
{-306.25, 2.75, 4., 1.5, 16., 10.5, 13.5, 64.}
{-306.75, 2.25, 3., 1.25, 12., 10.25, 13., 88.}
{-306.5, 2.5, 3.5, 1.5, 14., 10.5, 13.5, 80.}
{-307., 2., 2.5, 1.25, 10., 10.25, 13., 104.}
{-306.75, 2.25, 3., 1.5, 12., 10.5, 13.5, 96.}
{-307.25, 1.75, 2., 1.25, 8., 10.25, 13., 120.}
{-306.5, 2.5, 3.5, 1.75, 14., 10.75, 14., 96.}
{-307., 2., 2.5, 1.5, 10., 10.5, 13.5, 120.}
{-306.75, 2.25, 3., 1.75, 12., 10.75, 14., 112.}
Out[*]={-307.25, 1.75, 2., 1.5, 8., 10.5, 13.5, 136.}
{-307., 2., 2.5, 1.75, 10., 10.75, 14., 128.}
{-307.5, 1.5, 1.5, 1.5, 6., 10.5, 13.5, 152.}
{-306.75, 2.25, 3., 2., 12., 11., 14.5, 128.}
{-307.25, 1.75, 2., 1.75, 8., 10.75, 14., 152.}
{-307., 2., 2.5, 2., 10., 11., 14.5, 144.}
{-307.5, 1.5, 1.5, 1.75, 6., 10.75, 14., 168.}
{-307.25, 1.75, 2., 2., 8., 11., 14.5, 160.}
{-307.75, 1.25, 1., 1.75, 4., 10.75, 14., 184.}
{-307., 2., 2.5, 2.25, 10., 11.25, 15., 160.}
{-307.5, 1.5, 1.5, 2., 6., 11., 14.5, 184.}
{-307.25, 1.75, 2., 2.25, 8., 11.25, 15., 176.}
{-307.75, 1.25, 1., 2., 4., 11., 14.5, 200.}
{-307.5, 1.5, 1.5, 2.25, 6., 11.25, 15., 192.}
{-308., 1., 0.5, 2., 2., 11., 14.5, 216.}
{-307.25, 1.75, 2., 2.5, 8., 11.5, 15.5, 192.}
{-307.75, 1.25, 1., 2.25, 4., 11.25, 15., 216.}
{-307.5, 1.5, 1.5, 2.5, 6., 11.5, 15.5, 208.}
{-308., 1., 0.5, 2.25, 2., 11.25, 15., 232.}
{-307.75, 1.25, 1., 2.5, 4., 11.5, 15.5, 224.}
{-308.25, 0.75, 0., 2.25, 0., 11.25, 15., 248.}
{-307.5, 1.5, 1.5, 2.75, 6., 11.75, 16., 224.}
{-308., 1., 0.5, 2.5, 2., 11.5, 15.5, 248.}
{-307.75, 1.25, 1., 2.75, 4., 11.75, 16., 240.}
{-308.25, 0.75, 0., 2.5, 0., 11.5, 15.5, 264.}

```



We can therefore express each of these functions as elements enumerated by our algebra basis ab40. To properly give the principal parts and constants of each function, we define the following functions, as necessary in the construction of RaduRK:

```

In[ ]:= f1[s_, N_, M_, r_, m_, PM_] :=
  Product[
    Eta[Divisors[N][[i]]]^(s[[i]]),
    {i, 1, Length[Divisors[N]]}
  ] × Product[
    q^ $\left(\frac{24 \text{PM}[[i]] + \text{sigI}[r, M]}{24 m}\right)$ ,
    {i, 1, Length[PM]}
  ]

```

```

In[6]:= fLHS0[s_, N_, M_, r_, m_, PM_] :=
Module[
{
L =
Series[Product[QPochhammer[q^(Divisors[M][[d]])], q^(Divisors[M][[d]])]^(r[[d])],
{d, 1, Length[Divisors[M]}], {q, 0, m (pord[f1[s, N, M, r, m, PM]] + 1)}
],
g =
Normal[
Series[
f1[s, N, M, r, m, PM],
{q, 0, 0}
]
]
},
1;
If[
s == {},
1,
Normal[
Series[
g Product[
Sum[
Coefficient[L, q, m n + PM[[i]]] q^n,
{n, 0, pord[f1[s, N, M, r, m, PM]]}
],
{i, 1, Length[PM]}
],
{q, 0, 0}
]
]
]
]

In[6]:= fLHS[s_, N_, M_, r_, m_, PM_] :=
If[s == {} || s == {0}, "No Membership", fLHS0[s, N, M, r, m, PM]]

```

We now construct a table of the principal parts and constants of each function:

```

In[ ]:= fH =
  Table[
    {
      j1,
      fLHS[
        {219, -93, -16, -45, 44, 18, 128, -256},
        40,
        20,
        Exponent[mLp01w[[j1]], t] tv + Exponent[mLp01w[[j1]], x] xv +
          Exponent[mLp01w[[j1]], y] yv + {0, 0, 0, 2, -1, 0},
        4,
        {rem2[Exponent[mLp01w[[j1]], t],
          Exponent[mLp01w[[j1]], x], Exponent[mLp01w[[j1]], y]}]
      },
    {j1, 1, Length[mLp01w]}
  ];

```

Finally, we construct a table which contains the expression of each function as a member of the algebra basis (Notice that here, “t” represents our placeholder, in this case corresponding to `ab40[[1]]`, not our `Hauptmodul t`):

This took approximately seven hours’ computation time (as did checking the similar case for  $4n+3$  below).

```
In[ ]:= mW =
```

```
Table[
  {
    j1,
    MW[
      ab40[[1]], ab40[[2]],
      fH[[j1]][[2]]
    ]
  },
  {j1, 1, Length[mLp01w]}
]
```

```
Out[ ]:=
```

```
{ {1,
  {196 159 429 230 833 773 869 868 419 475 239 575 503 198 607 639 501 078 528 000 000 000 000 000 :
   000 000 000 000 000 000 000 000 000 000 000 000 000 +
   2 550 072 580 000 839 060 308 289 453 178 114 481 541 581 899 313 514 020 864 000 000 000 000 :
   000 000 000 000 000 000 000 000 000 000 000 000 000 t + ... 74 ... + 5 367 006 440 t76,
  { ... 2 ... , ... 1 ... }}, { ... 46 ... , ... 1 ... } }
```

large output

show less

show more

show all

set size limit...

There is one more consideration: each of these monomials has a given integer coefficient, which we recorded in the table mLp01wC. We combine each element of mLp01wC with the function for the associated progression mod 4 of the monomial. We don't want to worry about dealing with the actual eta quotients of ab40, so we replace them with indeterminates z5, z6, z7.

Each of these functions has an eta quotient multiplied on to ensure modularity. However, we went to the trouble of making sure that these multipliers were all the same (the eta quotient given by the vector (219, -93, -16, -45, 44, 18, 128, -256) over the divisors of 40). As such, the multiplier will have no effect if the progression comes out to be zero.

```
In[ ]:= Sum[
```

```
  mLp01wC[[j1]] × mW[[j1]][[2]].{1, z5, z6, z7},
  {j1, 1, Length[mLp01w]}
] // Together
```

```
Out[ ]:= 0
```

Which it does!

## $c\psi_{2,0}(n)$ : Odd Index, Progression $4n+3$

We work through the same steps as above for the progression  $4n+3$ . We of course will modify our key progression and variation on the minN programs:

```

In[ ]:= rem3[nt_, nx_, ny_] :=
  Mod[3 - (nt + nx + 2 ny), 4]

In[ ]:= MN3[f_] :=
  minN[
    20,
    Exponent[f, t] tv + Exponent[f, x] xv + Exponent[f, y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem3[Exponent[f, t], Exponent[f, x], Exponent[f, y]]
  ]

```

We also of course have a different multiplier for our ultimate identities, which we will give here as V3:

```

In[ ]:= V3 = {224, -107, -7, -48, 42, 26, 123, -254};

```

Otherwise, the steps are largely identical to those above.

```

In[ ]:= Column[
  Table[
    {j, MN3[mLp01w[[j]]}],
    {j, 1, Length[mLp01w]}
  ]
]

```

```
{1, 40}  
{2, 40}  
{3, 40}  
{4, 40}  
{5, 40}  
{6, 40}  
{7, 40}  
{8, 40}  
{9, 40}  
{10, 40}  
{11, 40}  
{12, 40}  
{13, 40}  
{14, 40}  
{15, 40}  
{16, 40}  
{17, 40}  
{18, 40}  
{19, 40}  
{20, 40}  
{21, 40}  
{22, 40}  
{23, 40}  
Out[*]= {24, 40}  
{25, 40}  
{26, 40}  
{27, 40}  
{28, 40}  
{29, 40}  
{30, 40}  
{31, 40}  
{32, 40}  
{33, 40}  
{34, 40}  
{35, 40}  
{36, 40}  
{37, 40}  
{38, 40}  
{39, 40}  
{40, 40}  
{41, 40}  
{42, 40}  
{43, 40}  
{44, 40}  
{45, 40}  
{46, 40}  
{47, 40}  
{48, 40}
```

```

In[ ]:= Pmr3[f_] :=
  Pmr [
    40,
    20,
    Exponent[f, t] tv + Exponent[f, x] xv + Exponent[f, y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem3[Exponent[f, t], Exponent[f, x], Exponent[f, y]]
  ]

```

```

In[ ]:= Column [
  Table [
    {j, Length[Pmr3[mLp01w[[j]]]}},
    {j, 1, Length[mLp01w]}
  ]
]

```

```
{1, 1}
{2, 1}
{3, 1}
{4, 1}
{5, 1}
{6, 1}
{7, 1}
{8, 1}
{9, 1}
{10, 1}
{11, 1}
{12, 1}
{13, 1}
{14, 1}
{15, 1}
{16, 1}
{17, 1}
{18, 1}
{19, 1}
{20, 1}
{21, 1}
{22, 1}
{23, 1}
Out[*]= {24, 1}
{25, 1}
{26, 1}
{27, 1}
{28, 1}
{29, 1}
{30, 1}
{31, 1}
{32, 1}
{33, 1}
{34, 1}
{35, 1}
{36, 1}
{37, 1}
{38, 1}
{39, 1}
{40, 1}
{41, 1}
{42, 1}
{43, 1}
{44, 1}
{45, 1}
{46, 1}
{47, 1}
{48, 1}
```



```

In[ ]:= Table[
  sSys[
    40,
    20,
    Exponent[mLp01w[[1]], t] tv + Exponent[mLp01w[[1]], x] xv +
      Exponent[mLp01w[[1]], y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem3[Exponent[mLp01w[[1]], t], Exponent[mLp01w[[1]], x], Exponent[mLp01w[[1]], y]],
    {rem3[Exponent[mLp01w[[1]], t], Exponent[mLp01w[[1]], x], Exponent[mLp01w[[1]], y]]}
  ] == sSys[
    40,
    20,
    Exponent[mLp01w[[j1]], t] tv + Exponent[mLp01w[[j1]], x] xv +
      Exponent[mLp01w[[j1]], y] yv + {0, 0, 0, 2, -1, 0},
    4,
    rem3[Exponent[mLp01w[[j1]], t],
      Exponent[mLp01w[[j1]], x], Exponent[mLp01w[[j1]], y]],
    {rem3[Exponent[mLp01w[[j1]], t], Exponent[mLp01w[[j1]], x],
      Exponent[mLp01w[[j1]], y]]}
  ],
  {j1, 1, Length[mLp01w]}
]

Out[ ]:= {True, True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True, True,
  True, True, True, True, True, True, True, True, True, True, True}

```

```

In[ ]:= Column[
  Table[
    Table[
      N[
        
$$\frac{40}{\text{GCD}[(\text{repMat}[40][[11]][[2]][[1]])^2, 40]} ($$

        p[repMat[40][[11]], 20, Exponent[mLp01w[[j1]], t] tv +
        Exponent[mLp01w[[j1]], x] xv + Exponent[mLp01w[[j1]], y] yv +
        {0, 0, 0, 2, -1, 0}, 4] + pStar[repMat[40][[11]], 40, V3]
        )
      ],
    {11, 1, Length[repMat[40]]}
  ],
  {j1, 1, Length[mLp01w]}
]
]

```

```

{-304.25, 2.75, 7., 0.5, 18., 11.5, 6.5, 0.}
{-304.75, 2.25, 6., 0.25, 14., 11.25, 6., 24.}
{-304., 3., 7.5, 0.75, 20., 11.75, 7., 0.}
{-304.5, 2.5, 6.5, 0.5, 16., 11.5, 6.5, 24.}
{-304.25, 2.75, 7., 0.75, 18., 11.75, 7., 16.}
{-304.75, 2.25, 6., 0.5, 14., 11.5, 6.5, 40.}
{-304.5, 2.5, 6.5, 0.75, 16., 11.75, 7., 32.}
{-305., 2., 5.5, 0.5, 12., 11.5, 6.5, 56.}
{-304.25, 2.75, 7., 1., 18., 12., 7.5, 32.}
{-304.75, 2.25, 6., 0.75, 14., 11.75, 7., 56.}
{-304.5, 2.5, 6.5, 1., 16., 12., 7.5, 48.}
{-305., 2., 5.5, 0.75, 12., 11.75, 7., 72.}
{-304.75, 2.25, 6., 1., 14., 12., 7.5, 64.}
{-305.25, 1.75, 5., 0.75, 10., 11.75, 7., 88.}
{-304.5, 2.5, 6.5, 1.25, 16., 12.25, 8., 64.}
{-305., 2., 5.5, 1., 12., 12., 7.5, 88.}
{-304.75, 2.25, 6., 1.25, 14., 12.25, 8., 80.}
{-305.25, 1.75, 5., 1., 10., 12., 7.5, 104.}
{-305., 2., 5.5, 1.25, 12., 12.25, 8., 96.}
{-305.5, 1.5, 4.5, 1., 8., 12., 7.5, 120.}
{-304.75, 2.25, 6., 1.5, 14., 12.5, 8.5, 96.}
{-305.25, 1.75, 5., 1.25, 10., 12.25, 8., 120.}
{-305., 2., 5.5, 1.5, 12., 12.5, 8.5, 112.}
{-305.5, 1.5, 4.5, 1.25, 8., 12.25, 8., 136.}
Out[*]=
{-305.25, 1.75, 5., 1.5, 10., 12.5, 8.5, 128.}
{-305.75, 1.25, 4., 1.25, 6., 12.25, 8., 152.}
{-305., 2., 5.5, 1.75, 12., 12.75, 9., 128.}
{-305.5, 1.5, 4.5, 1.5, 8., 12.5, 8.5, 152.}
{-305.25, 1.75, 5., 1.75, 10., 12.75, 9., 144.}
{-305.75, 1.25, 4., 1.5, 6., 12.5, 8.5, 168.}
{-305.5, 1.5, 4.5, 1.75, 8., 12.75, 9., 160.}
{-306., 1., 3.5, 1.5, 4., 12.5, 8.5, 184.}
{-305.25, 1.75, 5., 2., 10., 13., 9.5, 160.}
{-305.75, 1.25, 4., 1.75, 6., 12.75, 9., 184.}
{-305.5, 1.5, 4.5, 2., 8., 13., 9.5, 176.}
{-306., 1., 3.5, 1.75, 4., 12.75, 9., 200.}
{-305.75, 1.25, 4., 2., 6., 13., 9.5, 192.}
{-306.25, 0.75, 3., 1.75, 2., 12.75, 9., 216.}
{-305.5, 1.5, 4.5, 2.25, 8., 13.25, 10., 192.}
{-306., 1., 3.5, 2., 4., 13., 9.5, 216.}
{-305.75, 1.25, 4., 2.25, 6., 13.25, 10., 208.}
{-306.25, 0.75, 3., 2., 2., 13., 9.5, 232.}
{-306., 1., 3.5, 2.25, 4., 13.25, 10., 224.}
{-306.5, 0.5, 2.5, 2., 0., 13., 9.5, 248.}
{-305.75, 1.25, 4., 2.5, 6., 13.5, 10.5, 224.}
{-306.25, 0.75, 3., 2.25, 2., 13.25, 10., 248.}
{-306., 1., 3.5, 2.5, 4., 13.5, 10.5, 240.}
{-306.5, 0.5, 2.5, 2.25, 0., 13.25, 10., 264.}

```

```
In[ ]:= fH3 =
  Table[
    {
      j1,
      fLHS[
        V3,
        40,
        20,
        Exponent[mLp01w[[j1]], t] tv + Exponent[mLp01w[[j1]], x] xv +
          Exponent[mLp01w[[j1]], y] yv + {0, 0, 0, 2, -1, 0},
        4,
        {rem3[Exponent[mLp01w[[j1]], t],
          Exponent[mLp01w[[j1]], x], Exponent[mLp01w[[j1]], y]}]
    },
    {j1, 1, Length[mLp01w]}
  ];
```

```
In[ ]:= mW3 =
  Table[
    {
      j1,
      MW[
        ab40[[1]], ab40[[2]],
        fH3[[j1]][[2]]
      ]
    },
    {j1, 1, Length[mLp01w]}
  ]
```

Out[ ]:=

```
{ {1,
  {19 615 942 923 083 377 386 986 841 947 523 957 550 319 860 763 950 107 852 800 000 000 000 000 :
    000 000 000 000 000 000 000 000 000 000 000 000 000 +
    253 045 663 707 775 568 292 130 261 123 059 052 399 126 203 854 956 391 301 120 000 000 000 000 :
    000 000 000 000 000 000 000 000 000 000 000 000 000 t + ... 74 ... + 53 t76,
    ... 1 ... , ... 1 ... , ... 1 ... }}, ... 47 ... }
```

large output    [show less](#)    [show more](#)    [show all](#)    [set size limit...](#)

```
In[ ]:= Sum[
  mLp01wC[[j1]] × mW3[[j1]][[2]].{1, z5, z6, z7},
  {j1, 1, Length[mLp01w]}
] // Together
```

Out[ ]:= 0