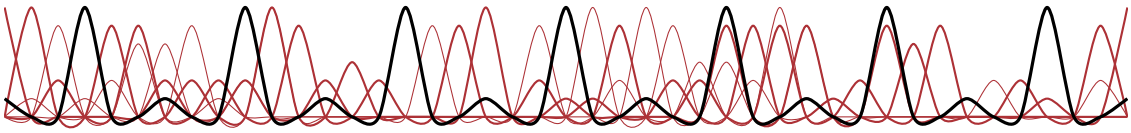
Taylor Expansions of Modular Forms at CM Points

ETH zürich

TAYLOR EXPANSIONS OF MODULAR FORMS AT CM POINTS



RYAN RUEGER

A semester thesis submitted to the Swiss Federal Institute of Technology Zurich, November 2023.

Supervised by Dr. M. Schwagenscheidt.

Contents

Introduction

The Fourier expansion, or q-expansion, of a modular forms is well-known. However, as holomorphic functions, modular forms also have a Taylor expansion at every point in the complex upper half-plane $\mathbb{H}$. Unfortunately, the naïve Taylor expansion at any given point z_0 in $\mathbb{H}$ only has a radius of convergence $r = \mathrm{Im}(z_0)$, which is unsatisfying, since the domain of definition of a modular form is the entirety of $\mathbb{H}$.

We will develop a method for computing the Taylor expansion of a modular form at complex-multiplication points (CM points) that is valid on the whole of $\mathbb{H}$ in two steps. First we will find a Taylor expansion of f that is valid on $\mathbb{H}$ as a function of the derivatives $f^{(r)}$ of f; then we will use a recursive procedure to calculate the derivatives of f, making the computation numerically easy without employing any analytic tools. Importantly, these recursions can be easily computed using computer algebra programs such as Sage.

In [OR12], O'Sullivan and Risager reference the work of Damerell to note that the Fourier coefficients of a cusp form when evaluated at a CM point in $\mathbb{H}$ are almost algebraic. To be more precise, given a cusp form f, a CM point ζ in $\mathcal{H}$ there exist complex numbers κ, λ so that $c_f(\zeta, n)/\kappa\lambda^n$ is algebraic for all n.

In particular, for ι and ρ we will show that

$$c_f(n,\rho) = \frac{(2\pi\iota)^n}{n!}\left(\sqrt{3}\right)^{n+k/2}\left(2\sqrt{3}\Omega_{-3}^6\right)^{(k+2n)/4} r_n(0)$$
$$c_f(n,\iota) = \frac{(2\pi\iota)^n}{n!}2^{n+k/2}\left(\Omega_{-4}^4\right)^{(k+2n)/6} q_n(0).$$

where Ω_D is the *Chowla-Selberg* period and $r_n(t), q_n(t)$ recursively defined polynomials.

In Section 1 we give an overview of the theory of complex multiplication to understand the meaning of a CM point. We show that the CM points τ in $\mathbb{H}$ parametrise the elliptic curves over $\mathbb{C}$ with endomorphism rings isomorphic to the order $\mathbb{Z}\tau + \mathbb{Z}$ in $\mathbb{Q}(\tau)$. In Section 2 we develop the theory required to compute the Taylor expansion of a modular form at any CM point in $\mathbb{H}$. We note that at $\zeta_{-4} = \iota$ and $\zeta_{-3} = \rho$ the expressions are particularly easy to compute. Finally, in Section 3 we present methods to detect non-vanishing behaviour of Fourier coefficients.

Using the methods of Zagier and O'Sullivan-Risager, we prove in Subsection 3.8 that the non-trivial Fourier coefficients of the j-function expanded at ι and ρ are eventually non-zero. Around ρ the expansion can be particularly compactly described. In fact, we will see that

$$q_{6n}(0) \equiv 6 \pmod 7 \quad \text{and} \quad q_{6n+3}(0) \equiv 1 \pmod 7$$

for $n > 1$, where $q_n(t)$ are recursively defined polynomials satisfying the description for $c_j(n,\iota)$ as above. In particular, we see that the non-trivial Fourier coefficients of j expanded at ρ are all non-zero except for the first one.

The graphic on the title page is a stylised version of Figure 3.2, a graph depicting the behaviour of the coefficients of the sequence $q_n(t) \pmod 7$ for E_4^3 over one period. Each line depicts the coefficient of a different monomial t^k in $q_n(t) \pmod 7$.

1
Complex-multiplication

The theory of complex multiplication arises naturally in the study of endomorphism rings of elliptic curves. To see this we briefly review some theory of elliptic curves in the following to establish a bijection between elliptic curves E defined over $\mathbb{C}$ and lattices Ω in $\mathbb{C}$. From here, we will see that endomorphisms of an elliptic curve E are in bijection with $\{\lambda \in \mathbb{C} \mid \lambda\Omega \subseteq \Omega\}$ — justifying the nomenclature "complex multiplication".

1.1 Elliptic Curves and their representations

As Milne points out in *Elliptic Curves* [Mil06], there are many equivalent definitions. At their heart, *elliptic curves* over a field k are one-dimensional regular (irreducible[1]) projective varieties with a group law, but can be expressed more concretely as projective varieties cut out from $\mathbb{P}^3_k$ by equations in *Weierstrass form*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \tag{1.1}$$

whose coefficients a_i lie in k. We also call k-triples $(x : y : z)$ that satisfy the equation (1.1) *k-rational points* on the curve; in part, to distinguish from triples $(\overline{x} : \overline{y} : \overline{z})$ in some extension F/k that may also satisfy the equation.

In the projective variety formulation of an elliptic curve, the group law can be expressed geometrically, or via explicit formulas given by Silverman and Tate in *Rational points on Elliptic curves* [ST15] Chapter 1.4. Importantly, the explicit formulas show that addition can be expressed as a regular map, and so is compatible with the variety structure.

More precisely, since they are complete [Mil17, Th. 7.22, pg 158] and connected (they are irreducible) with a group law described by regular maps, elliptic curves are instances of abelian varieties. As such, we emphasise that a *regular map* of elliptic curves is a morphism in the category of varieties, and we will call a *homomorphism* a morphism in the category of abelian varieties. That is, a homomorphism is a regular map which is also a morphism of groups. It turns out, however, that these notions are not too different: a regular map is a homomorphism if and only if it sends 0 to 0, a direct consequence of [Mil08, Cor. 1.2, pg. 9].

The neutral element of the group is often denoted by $\mathcal{O}$ and called the *point at infinity* for geometric reasons. We use the notation E/k to denote an elliptic curve E over the field k; this is consistent with the notation of field extensions, because of the contravariant equivalence of categories of 1-dimensional regular projective curves over k and field extensions F/k of transcendence degree 1 [Vak23, Th. 16.3.3, pg.436; Sil09, Rem. 2.5, pg.22].

Finally, we note that many authors introduce the language of isogenies in the context of elliptic curves. An *isogeny* is a surjective homomorphism between abelian varieties of the same dimension. However due to their relatively simple structure, any non-constant homomorphism of elliptic curves is automatically surjective. Indeed, its image is dense (*dominant*) [Gal18, Lem. 8.1.3, pg. 145] and finite [Gal18, Th. 8.1.5, pg. 146], hence surjective [Mil17, Prop. 2.41, pg. 50].

[1] For some authors, all projective varieties are automatically irreducible.

Whilst the representation (1.2) is more concrete, computations are still not necessarily very easy. For example, we would like to compute the j-invariant of an elliptic curve, as an easy function of its coefficients. To that end, we perform the following reductions.

When k has characteristic $\mathrm{char}(k) \neq 2, 3$, we can replace Y with $Y - a_1X/2$ to eliminate the XYZ term in (1.1)

$$\begin{aligned}&(Y - a_1X/2)^2Z + a_1X(Y - a_1X/2)Z + a_3(Y + a_1X/2)Z^2\\ &= Y^2Z - a_1XYZ + a_1^2X^2Z/4 + a_1XYZ - a_1^2X^2Z/2 + a_3YZ^2 + a_1a_3XZ^2/2\\ &= Y^2Z - a_1^2X^2Z/4 + a_3YZ^2 + a_1a_3XZ^2/2.\end{aligned}$$

Rearranging, we obtain the equation

$$Y^2Z + a_3YZ^2 = X^3 + (a_2 + a_1^2/4)X^2Z + (a_4 - a_1a_3/2)XZ^2 + a_6Z^3.$$

We can now complete the square in terms of Y, eliminating the YZ^2 term and *depress* the cubic in X, eliminating the X^2Z term. The required substitutions for this procedure replace Y with $Y - a_3Z/2$ and X with $X - (a_2 + a_1^2/4)/3\,Z$ respectively. We result in an equation of the following form

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{1.2}$$

and call this the *reduced* or *short* Weierstrass form of an elliptic curve. We use the notation $\mathcal{E}(a, b)$ to denote the variety cut out by the equation (1.2). Notably, the point at infinity $\mathcal{O}$ is given by $(0 : 1 : 0)$ in homogeneous coordinates.

The k-rational points on $\mathcal{E}(a, b)$ in the line $\{Z = 1, Y = 0\} = \{(x : 0 : 1) \mid x \in k\}$, are exactly roots of $X^3 + aX + b$ in k, and so uniquely determined by a, b.

We note that each of the substitutions induces an isomorphism of varieties. Indeed, by [Mil17, Prop. 6.20, pg. 138] it suffices to show that they are described by homogeneous polynomials in each component and so is their inverse. Clearly both of these requirements are satisfied.

The converse converse, namely that every variety $\mathcal{E}(a, b)$ is an elliptic curve is *not* true in general, but can easily be recovered as in the following

Theorem 1.1 (Existence of the reduced Weierstrass form [Mil06, Th. 1.2(a), pg. 50])**.** *Let k be a field with characteristic* $\mathrm{char}(k) \neq 2, 3$*. Every elliptic curve E over k is isomorphic to some $\mathcal{E}(a, b)$ and conversely every $\mathcal{E}(a, b)$ is an elliptic curve if $4a^3 + 27b^2 \neq 0$.*

The expression $4a^3+27b^2$ is the discriminant of the depressed cubic X^3+aX+b and so the condition $4a^3+27b^2 \neq 0$ essentially enforces regularity of the curve: it requires the cubic X^3+aX+b to have distinct roots and ensures that the curve has no singularities (geometrically described as *cusps*) in the plane $\{Z = 1\}$.

Our initial goal was to find a way to describe elliptic curves (up to isomorphism) which allow us to compute attached constants like the j-invariant, or the discriminant Δ directly from coefficients of the equation. Unfortunately, the Weierstrass reduced form of an elliptic curve is not unique, and so any definition of j, Δ based on the coefficients of the reduced Weierstrass form would *a priori* not be well-defined.

The following theorem describes the in what sense the representation is uniqe.

Theorem 1.2 (Uniqueness of the reduced Weierstrass form [Mil06, Th. 1.2(b), pg. 50]). *Let k be a field with characteristic* $\mathrm{char}(k) \neq 2,3$. *For non-zero u in k, the map $\varphi(u)\colon \mathcal{E}(a,b) \to \mathcal{E}(u^4a, u^6b); (x:y:z) \mapsto (u^{-2}x : u^{-3}y : z)$ is an isomorphism of abelian varieties. Conversely, if $\varphi\colon \mathcal{E}(a,b) \to \mathcal{E}(a',b')$ is a group-law preserving isomorphism, then $\varphi = \varphi(u)$ for some non-zero u in k. Consequently, $\mathcal{E}(a,b), \mathcal{E}(a',b')$ are isomorphic as abelian varieties if and only if $a' = u^4a, b' = u^6b$ for some non-zero u in k.*

An immediate consequence of this theorem, is that the definition of the j-invariant

$$j(E) \overset{\text{def.}}{=} j(\mathcal{E}(a,b)) = 1728\frac{(4a)^3}{4a^3 + 27b^2}$$

of an elliptic curve $E \cong \mathcal{E}(a,b)$ is good. Indeed, if $E \cong \mathcal{E}(a,b) \cong \mathcal{E}(a',b')$, then $a' = u^4a, b' = u^6b$ for u non-zero in k and

$$j(\mathcal{E}(a,b)) = 1728\frac{(4a)^3}{4a^3 + 27b^2} = 1728\frac{u^{12}(4a)^3}{4u^{12}a^3 + 27u^{12}b^2} = 1728\frac{a'^3}{4(4a')^3 + 27b'^2} = j(\mathcal{E}(a',b')).$$

This theorem also shows how we can extend the notion of isomorphism. We continue to say that two elliptic curves defined over k (that is, cut out from $\mathbb{P}^3_k$) are isomorphic *over k*, if there is a map $(x:y:z) \mapsto (u^2x : u^3y : z)$ with u non-zero in k; we say that the curves (still defined over k) are isomorphic *over $\overline{k}$* if there exists a non-zero w in $\overline{k}$ so that $E \to E'; (x:y:z) \mapsto (w^2x : w^3y : z)$ is an isomorphism. Milne gives a concrete example in [Mil06, Rem. 2.2, pg. 52]. For c in k not a square, the curve

$$Y^2Z = X^3 + c^2XZ^2 + c^3Z^3 \quad \text{is not isomorphic to} \quad Y^2Z = X^3 + XZ^2 + Z^3$$

because the desired isomorphism $\varphi(c^{1/2})\colon \mathcal{E}(1,1) \to \mathcal{E}(c^2,c^3) = (x:y:z) \mapsto (c^{-1}x : c^{-3/2}y : z)$ does not exist over k (if c is not a square k does not contain $c^{1/2}$). It does, however, exist over $\overline{k}$. This inspires the easy

Lemma 1.3. *Let k be a field with characteristic $k \neq 2,3$. Two elliptic curves E, E' are isomorphic over $\overline{k}$ if and only if $j(E) = j(E')$.*

The necessity is clear from our definition, namely $j(E) \overset{\text{def.}}{=} j(\mathcal{E}(a,b))$ where $E \cong \mathcal{E}(a,b)$. For sufficiency, we separate the cases $j(E) = 0$ and $j(E) \neq 0$. Let $E \cong \mathcal{E}(a,b)$ and $E' \cong \mathcal{E}(a',b')$. If $j(E) = j(E') = 0$ then $a = a' = 0$ and $E \cong \mathcal{E}(0,b), E' \cong \mathcal{E}(0,b')$. These curves are clearly isomorphic via $\varphi((b'/b)^{1/2})$. If $j(E) \neq 0$, $a \neq 0$ and a similar method works. The isomorphism $\varphi((a'/a)^{1/4})$ sends $\mathcal{E}(a,b)$ to $\mathcal{E}(a',(a'/a)^{3/2}b)$. Writing $b'' = (a'/a)^{3/2}b$ we note that $j(\mathcal{E}(a,b)) = j(\mathcal{E}(a',b'')) = j(\mathcal{E}(a',b'))$. The last equality tells us, that $b'' = \pm b'$ and by choosing $u = (\pm a'/a)^{1/4}$ appropriately, we obtain $\varphi(u)\colon \mathcal{E}(a,b) \xrightarrow{\sim} \mathcal{E}(a',b'') = \mathcal{E}(a',b')$.

Finally, we will transform the reduced Weierstrass form one more time, for future convenience. Again k is a field with $\mathrm{char}(k) \neq 2,3$. Replacing Y with $2Y$ in (1.2) we obtain the equation $Y^2Z = 4X^3 + 4aXZ^2 + 4bz^3$. If we write $A = -4a, B = -4b$, we have $ZY^2 = 4X^3 - AXZ^2 - BZ^3$ and the condition $4a^3 + 27b^2 \neq 0$ becomes $A^3 - 27B^2 \neq 0$. We will use the notation $E(a,b)$ to denote the projective variety cut out by $Y^2Z = 4X^3 + aX + b$. We note that Theorem 1.2 still applies with the same conditions on curves described by $E(a,b)$.

We summarise this section in the following

Theorem 1.4 (Representations of elliptic curves)**.** *Let k be a field with characteristic* $\mathrm{char}(k) \neq 2, 3$*. Let u be non-zero in k. Here, isomorphisms are in the category of abelian varieties. Let $E(a,b)$ be the projective variety cut out from $\mathbb{P}^3_k$ by $ZY^2 = 4X^3 - aXZ^2 - bZ^3$ then*

(i) Every elliptic curve is isomorphic to some $E(a,b)$.

(ii) $E(a,b)$ is an elliptic curve if and only if $a^3 - 27b^2 \neq 0$.

(iii) $\varphi\colon E(a,b) \to E(a',b')$ is an isomorphism if and only if $\varphi = (x:y:z) \mapsto (u^2x : u^3y : z)$.

(iv) The elliptic curves $E(a,b), E(a',b')$ are isomorphic if and only if $a' = u^4a, b' = u^6b$.

In other words we have the bijection of sets

$$\begin{aligned}
&\{\textit{Elliptic curves over } k\}/_{\textit{isomorphism}} \\
\cong &\{E(a,b) \mid a,b \in k, a^3 - 27b^2 \neq 0\}/\{E(a,b) \sim E(u^4a, u^6b) \mid u \in k \setminus \{0\}\} \\
\cong &\{(a,b) \in k^2 \mid a^3 - 27b^2 \neq 0\}/\{(a,b) \sim (u^4a, u^6b) \mid u \in k \setminus \{0\}\}
\end{aligned}$$

This bijection is explicitly described by the linear transformations illustrated in this section.

1.2 Elliptic curves and Lattices

The content of Theorem 1.4 allows us to show that any elliptic curve is isomorphic as a group the quotient $\mathbb{C}/\Omega$ for a lattice Ω uniquely determined by E in relatively few steps. In other words elliptic curves are in bijection with the lattices in $\mathbb{C}$.

We recall that a lattice $\Omega \subseteq \mathbb{C}$ is a free $\mathbb{Z}$-module generated by an $\mathbb{R}$-basis of $\mathbb{C}$. That is, a lattice is of the form $\Omega = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ where ω_1, ω_2 are an $\mathbb{R}$-basis of $\mathbb{C}$. The quotient group $\mathbb{C}/\Omega$ is also called a *complex torus* and can be understood as a complex manifold. It is well known that the *Eisenstein series* of weight $k \geq 2$ for the lattice Ω

$$G_k(\Omega) = \sum_{\substack{\omega \in \Omega \\ \omega \neq 0}} \omega^{-k}$$

converges. From this, we define the *Weierstrass invariants* $g_2(\Omega) = 60G_4(\Omega)$ and $g_3(\Omega) = 140G_6(\Omega)$ of the lattice Ω. These are invariants in the following sense.

Theorem 1.5 (Uniformisation Theorem)**.**

(i) [KK07, Cor. F, pg. 39] *Every lattice $\Omega \subseteq \mathbb{C}$ depends only on its Weierstrass invariants.*

(ii) [KK07, Cor. on pg. 56] *The complex numbers a, b are the Weierstrass invariants $g_2(\Omega), g_3(\Omega)$ of a lattice $\Omega \subseteq \mathbb{C}$ if and only if $a^3 - 27b^2 \neq 0$.*

This means we can unambiguously write $\Omega(a,b)$ for the lattice corresponding to the Weierstrass invariants a, b for every complex a, b with $a^3 - 27b^2 \neq 0$. We immediately get the bijection

$$\begin{aligned}
\{E(a,b)/\mathbb{C} \mid a,b \in \mathbb{C}, a^3 - 27b^2 \neq 0\} &\to \{\Omega(a,b) \subseteq \mathbb{C} \mid a,b \in \mathbb{C}, a^3 - 27b^2 \neq 0\} \\
E(a,b) &\mapsto \Omega(a,b).
\end{aligned}$$

Since the $E(a,b)/\mathbb{C}$ are (not necessarily different) representatives of isomorphism classes of elliptic curves (over $\mathbb{C}$), we expect to extend this bijection somehow to a bijection between isomorphism classes of elliptic curves (over $\mathbb{C}$) and lattices up to a yet to be determined relation.

We note that the Weierstrass invariants are homogeneous of degree -4 and -6 respectively. That is, for Ω a lattice in $\mathbb{C}$ and λ a non-zero scalar in $\mathbb{C}$ we have $G_k(\lambda\Omega) = \lambda^{-k}G_k(\Omega)$ and so $g_2(\lambda\Omega) = \lambda^{-4}g_2(\Omega), g_3(\lambda\Omega) = \lambda^{-6}g_3(\Omega)$. We call two lattices Ω, Ω' *homothetic* if $\Omega' = \lambda\Omega$ for some (non-zero) λ in $\mathbb{C}$. So using the uniqueness provided by the uniformisation theorem, we conclude that two lattices are Ω, Ω' homothetic if and only if their Weierstrass invariants satisfy $g_2(\Omega) = \lambda^4 g_2(\Omega'), g_3(\Omega) = \lambda^6(\Omega')$ for some non-zero λ in $\mathbb{C}$.

This insight gives us the important

Theorem 1.6. *The set of elliptic curves over $\mathbb{C}$ up to isomorphism is in bijection to the lattices $\Omega \subseteq \mathbb{C}$ up to homothety via the map*

$$\{\text{Elliptic curves over } \mathbb{C}\}/_{\text{isomorphism}} \to \{\text{Lattices } \Omega \subseteq \mathbb{C}\}/_{\text{homothety}}$$
$$[E(a,b)] \mapsto [\Omega(a,b)]$$

In fact, together with the meromorphic *Weierstrass $\wp$-function*

$$\wp_\Omega(z) = \frac{1}{z^2} + \sum_{\substack{\omega\in\Omega \\ \omega\neq 0}} \frac{1}{(z-\omega)^2} - \frac{1}{\omega^2}$$

we can go further to show that

Lemma 1.7 ([KK07, Satz on pg. 64])**.** *Given complex numbers a, b satisfying $a^3 - 27b^2 \neq 0$ we obtain an isomorphism of groups*

$$\mathbb{C}/_{\Omega(a,b)} \to E(a,b) \qquad z \mapsto (\wp_{\Omega(a,b)}(z) : \wp'_{\Omega(a,b)}(z) : 1) \tag{1.3}$$

where $\Omega(a,b)$ is the lattice corresponding to the Weierstrass invariants a, b and 0 in $\mathbb{C}/_{\Omega(a,b)}$ is sent to the point at infinity $\mathcal{O}$.

A key ingredient in the proof is that the Weierstrass $\wp$-function satisfies the differential equation

$$\wp'_\Omega(z)^2 = 4\wp_\Omega(z)^2 - g_2(\Omega)\wp_\Omega(z) - g_3(\Omega)$$

Some authors use more delicate language to phrase this correspondence as one of equivalent categories, for instance [Sut21, Lecture 17] and [Sil09, Th. 5.3, pg. 175].

It is an immediate consequence of the lemma, is that we can also attach the same j-invariant to a lattice $\Omega(a,b)$ with the same formula

$$j(\Omega(a,b)) = 1728\frac{(4a)^3}{4a^3 + 27b^2} = j(E(a,b)).$$

We also obtain a further immediate

Corollary 1.8 (of Lemma 1.3)**.** *Two lattices Ω, Ω' in $\mathbb{C}$ are homothetic if and only if $j(\Omega) = j(\Omega')$.*

If $\Omega = \lambda\Omega'$ then it is clear that $j(\Omega) = j(\Omega')$. Conversely, if $j(\Omega) = j(\Omega')$ set $E = E(g_2(\Omega), g_3(\Omega))$ and $E' = E(g_2(\Omega'), g_3(\Omega'))$. Then $j(E) = j(E')$, so by Lemma 1.3 $E \cong E'$ and Ω, Ω' are homothetic by Theorem 1.6.

1.3 Interlude: Lattice representations and modularity of the j-invariant

Finally, before we move on, we make a short remark on the representation of homethety classes of lattices and how $\mathrm{SL}_2(\mathbb{Z})\backslash\mathbb{H}$ is a moduli space for the space of elliptic curves.

We defined a lattice in $\mathbb{C}$ to be a free $\mathbb{Z}$-module of the form $\Omega = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2 \subseteq \mathbb{C}$ with ω_1, ω_2 an $\mathbb{R}$-basis of $\mathbb{C}$. This means $\tau = \omega_1/\omega_2$ is not real and we may assume, by replacing ω_1 with $-\omega_1$ if necessary, that $\mathrm{Im}(\tau) > 0$, that is, τ lies in the complex upper-half plane $\mathbb{H}$. Note that exchanging ω_1 with $-\omega_1$ does not change the lattice Ω. Consequently, we can write $\Omega = \omega_2^{-1}(\mathbb{Z}\tau + \mathbb{Z})$ and conclude that every homothety class of lattices in $\mathbb{C}$ can be represented by a τ in $\mathbb{H}$ with the lattice $\mathbb{Z}\tau + \mathbb{Z}$.

Recalling the $\mathrm{GL}_2(\mathbb{R})$-action on $\mathbb{C}$, we also obtain the following

Lemma 1.9. *The lattices $\Omega = \mathbb{Z}\tau + \mathbb{Z}$ and $\Omega' = \mathbb{Z}\tau' + \mathbb{Z}$ with τ, τ' in $\mathbb{H}$ are homothetic if and only if $\tau' = \gamma\tau$ for some γ in $\mathrm{SL}_2(\mathbb{Z})$.*

Indeed, if $\tau' = \gamma\tau$ and $\gamma = (a, b; c, d)$ in $\mathrm{SL}_2(\mathbb{Z})$, then

$$\begin{aligned}\mathbb{Z}\tau' + \mathbb{Z} &= \mathbb{Z}\left(\frac{a\tau + b}{c\tau + b}\right) + \mathbb{Z} \\ &= (c\tau + b)^{-1}\left(\mathbb{Z}(a\tau + b) + \mathbb{Z}(c\tau + d)\right) \\ &= (c\tau + b)^{-1}\left((\mathbb{Z}a + \mathbb{Z}c)\tau + (\mathbb{Z}b + \mathbb{Z}d)\right) \\ &= (c\tau + b)^{-1}\left(\mathbb{Z}\tau + \mathbb{Z}\right)\end{aligned}$$

where the penultimate equality holds because a, c and b, d are coprime. Conversely, suppose $\mathbb{Z}\tau' + \mathbb{Z} = \lambda(\mathbb{Z}\tau + \mathbb{Z}) = \mathbb{Z}\lambda\tau + \mathbb{Z}\lambda$ for some λ in $\mathbb{C}$. Then there exist integers a, b, c, d so that $\tau' = a\lambda\tau + b\lambda$ and $1 = c\lambda\tau + \lambda d$. Forming the matrix $\gamma = (a, b; c, d)$ we obtain $\tau' = \gamma\tau$ by substitution. By symmetry we also have $\delta = (e, f; g, h)$ with integral entries, $\lambda\tau = e\tau' + f$, $\lambda = g\tau' + h$ and $\tau = \delta\tau'$. In matrix form, we can write this as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \lambda\tau \\ \lambda \end{pmatrix} = \begin{pmatrix} \tau' \\ 1 \end{pmatrix}, \tau' = \gamma\tau = \frac{a\tau + b}{c\tau + d} \quad \text{and} \quad \begin{pmatrix} e & f \\ g & h \end{pmatrix}\begin{pmatrix} \tau' \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda\tau \\ \lambda \end{pmatrix}, \tau = \delta\tau' = \frac{e\tau' + f}{g\tau' + h}.$$

So if we write $\gamma\delta = (j, k; l, m)$, then

$$\begin{pmatrix} j & k \\ l & m \end{pmatrix}\begin{pmatrix} \tau' \\ 1 \end{pmatrix} = \begin{pmatrix} \tau' \\ 1 \end{pmatrix} \quad \text{so} \quad j\tau' + k = \tau' \quad \text{and} \quad l\tau' + m = 1$$

since Ω' is a lattice by assumption, $\tau', 1$ is an $\mathbb{R}$-basis of $\mathbb{C}$ and in particular linearly independent over $\mathbb{R}$. Consequently, $j = 1, k = 0$ and $l = 0, m = 1$. In other words, γ, δ invertible over $\mathbb{Z}$ and so $\det(\gamma), \det(\delta) = \pm 1$. Finally, note that

$$\mathrm{Im}(\tau') = \mathrm{Im}(\gamma\tau) = \mathrm{Im}\left(\frac{a\tau + b}{c\tau + d}\right) = \frac{\mathrm{Im}\left((a\tau + b)(c\overline{\tau} + d)\right)}{|c\tau + d|^2} = \frac{\mathrm{Im}\left(bc\overline{\tau} + ad\tau\right)}{|c\tau + d|^2} = \frac{\det(\gamma)\,\mathrm{Im}(\tau)}{|c\tau + d|^2}.$$

Since $\tau' = \gamma\tau$ lies in $\mathbb{H}$, we conclude that $\det(\gamma) > 0$, so $\det(\gamma) = 1$ and γ lies in $\mathrm{SL}_2(\mathbb{Z})$.

Representing a lattice $\Omega = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ by $\Omega = \omega_2(\mathbb{Z}\omega_1/\omega_2 + \mathbb{Z})$ with $\tau = \omega_1/\omega_2$ in $\mathbb{H}$ illustrates how the j-invariant and the Eisenstein series can be expressed as functions $\mathbb{H} \to \mathbb{C}$ by sending $\tau \mapsto j(\mathbb{Z}\tau + \mathbb{Z})$ and $\tau \mapsto G_k(\mathbb{Z}\tau + \mathbb{Z})$. Moreover, the statement on homothety shows that the

j-invariant and G_k are weakly modular of weight 1 and k respectively. Indeed

$$j(\gamma\tau) = j(\mathbb{Z}\gamma\tau + \mathbb{Z}) = j((c\tau+d)^{-1}(\mathbb{Z}\tau+\mathbb{Z})) = j(\mathbb{Z}\tau+\mathbb{Z}) = j(\tau)$$
$$G_k(\gamma\tau) = G_k(\mathbb{Z}\gamma\tau + \mathbb{Z}) = G_k((c\tau+d)^{-1}(\mathbb{Z}\tau+\mathbb{Z})) = (c\tau+d)^k G_k(\mathbb{Z}\tau+\mathbb{Z}) = (c\tau+d)^k G_k(\tau).$$

Here we are switching between viewing j, G_k as functions of lattices and as functions on $\mathbb{H}$.

Moreover, we see that the j-function is injective up to the $\mathrm{SL}_2(\mathbb{Z})$ action. Indeed, $j(\tau) = j(\tau')$ if and only if $\mathbb{Z}\tau + \mathbb{Z}$ and $\mathbb{Z}\tau' + \mathbb{Z}$ are homothetic by Corollary 1.8, which they are if and only if $\tau' = \gamma\tau$ for γ in $\mathrm{SL}_2(\mathbb{Z})$ by Lemma 1.9. Conversely, it is well-known that the j-function is surjective [KK07, Satz C, pg. 55].

This yields what is known as a *moduli space*. The space $\mathrm{SL}_2(\mathbb{Z})\backslash\mathbb{H}$ parametrises the space of elliptic curves up to isomorphism.

1.4 Classification of Endomorphisms of Elliptic Curves I

If we view $E(a,b)/\mathbb{C}$ in the plane $\{Z=1\} \cong \mathbb{C}^2$, we see that the map (1.3) is a biholomorphic map. Silverman formalises this idea in [Sil09, Prop. 5.2, pg. 174] when proving a categorical equivalence between complex tori and elliptic curves over $\mathbb{C}$. This means every endomorphism of an elliptic curve $E \cong E(a,b)$ induces an endomorphism on the corresponding complex torus $\mathbb{C}/\Omega$ where $\Omega = \Omega(a,b)$. More precisely, every endomorphism $E \to E$ induces a holomorphic additive group morphism $\varphi\colon \mathbb{C}/\Omega \to \mathbb{C}/\Omega$.

Let us study these maps. Since $\mathbb{C}$ is simply connected, we can lift φ to a holomorphic map $\Phi\colon \mathbb{C} \to \mathbb{C}$ which commutes with the quotient map $\pi\colon \mathbb{C} \to \mathbb{C}/\Omega$ in the usual way $\pi \circ \Phi = \pi \circ \varphi$. Since φ is a group morphism $\varphi(0) = 0$, so $\Phi(0)$ lies in Ω and without loss of generality we choose $\Phi(0) = 0$. For every ω in Ω the image of the map $z \mapsto \Phi(z+\omega) - \Phi(z)$ is contained in Ω. Indeed, $\pi\circ(\Phi(z+\omega)-\Phi(z)) \overset{!}{=} \varphi(\pi(z+\omega)) - \varphi(\pi(z)) = \pi(0)$. So, because Ω is discrete and $\mathbb{C}$ connected we conclude that $z \mapsto \Phi(z+\omega)-\Phi(z)$ is constant for all ω and $\Phi'(z+\omega) = \Phi'(z)$ for all z in $\mathbb{C}$ and ω in Ω. As such, Φ' is an entire holomorphic elliptic function and so constant. Indeed, as a continuous (holomorphic) function it is bounded the closure of any fundamental parallelogram; because it is elliptic, its behaviour on all of $\mathbb{C}$ is described by its behaviour on a fundamental parallelogram; and Liouville's theorem on bounded functions tells us that bounded entire functions are constant. Therefore $\Phi(z) = \alpha z + \beta$. However, since we chose $\Phi(0) = 0$, we know $\beta = 0$, so $\Phi(z) = \alpha z$ and $\varphi(\pi(z)) = \pi(\alpha z)$.

Conversely, the map $\Phi(z) = \alpha z$ induces an endomorphism on $\mathbb{C}/\Omega$ if $\Phi(\Omega) = \alpha\Omega \subseteq \Omega$. Indeed, the map is well-defined and surely a group morphism.

In summary, for an elliptic curve $E/\mathbb{C} \cong E(a,b)$ we have the bijection

$$\begin{aligned}
&\mathrm{End}_{\text{elliptic curves}}(E) \\
&\cong \mathrm{End}_{\text{complex tori}}\left(\mathbb{C}/\Omega(a,b)\right) \\
&\cong \left\{\varphi_\alpha\colon \mathbb{C}/\Omega(a,b) \to \mathbb{C}/\Omega(a,b); \pi(z) \mapsto \pi(\alpha z) \mid \alpha \in \mathbb{C}, \alpha\Omega(a,b) \subseteq \Omega(a,b)\right\} \\
&\cong \{\alpha \in \mathbb{C} \mid \alpha\Omega(a,b) \subseteq \Omega(a,b)\}\,.
\end{aligned} \tag{1.4}$$

In fact, since the relationship between curves and lattices is functorial, this is an isomorphism of rings induced by (1.3). The second isomorphism is also one of rings via the obvious map. We note

that (1.4) does not depend on the choice of representative $E(a,b)$ of the isomorphism class of E. Indeed, if we chose $E(a',b') \cong E(a,b)$, then the lattices $\Omega = \Omega(a,b), \Omega' = \Omega(a',b')$ are homothetic $\Omega = \lambda\Omega'$ and the rings

$$\{\alpha \in \mathbb{C} \mid \alpha\Omega \subseteq \Omega\} = \{\alpha \in \mathbb{C} \mid \alpha\lambda\Omega \subseteq \lambda\Omega\} = \{\alpha \in \mathbb{C} \mid \alpha\Omega' \subseteq \Omega'\} \tag{1.5}$$

are equal. Moreover, in light of (1.4) we write $\mathrm{End}(\Omega) = \{\alpha \in \mathbb{C} \mid \alpha\Omega \subseteq \Omega\}$ for every lattice $\Omega \subseteq \mathbb{C}$.

We now want to classify the endomorphisms of an elliptic curve over $\mathbb{C}$. Let $E/\mathbb{C} \cong E(a,b)$ be an elliptic curve and $\Omega = \Omega(a,b)$ a corresponding lattice.

We first understand that $\mathrm{End}(E)$ contains a subring isomorphic to $\mathbb{Z}$. Indeed, since Ω is a lattice it is a $\mathbb{Z}$-module and $\alpha\Omega \subseteq \Omega$ holds for every integer α and so yields a well-defined map φ_α. Moreover, the identification $\mathbb{Z} \to \mathrm{End}(E)$ sending $\alpha \mapsto \varphi_\alpha$ is injective. Suppose $\varphi_\alpha = \varphi_\beta$, then the image of $d\colon z \mapsto (\alpha - \beta)z$ lies in Ω for all z which in turn holds if and only if d is constant because Ω is discrete, $\mathbb{C}$ connected and d continuous. Clearly d is only constant when $\alpha = \beta$.

Now suppose α is a non-integer element of $\mathrm{End}(E)$. Let $\Omega(a,b) = \mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ with $\tau = \omega_1/\omega_2$ in $\mathbb{H}$. Then there exist integers k,l,m,n so that

$$\alpha\omega_1 = k\omega_1 + l\omega_2 \quad \text{and} \quad \alpha\omega_2 = m\omega_1 + n\omega_2 \quad \text{so} \quad \alpha\tau = k\tau + l \quad \text{and} \quad \alpha = m\tau + n.$$

Solving for τ and α respectively, we garner

$$m\tau^2 + (n-k)\tau - l = 0 \quad \text{and} \quad \alpha^2 - (k+n)\alpha + kn - ml = 0. \tag{1.6}$$

We immediately note that if $\mathbb{Z} \subsetneq \mathrm{End}(E)$, then Ω is homothetic to $\mathbb{Z}\tau + \mathbb{Z}$ with an algebraic τ. Or framed in the contrapositive, if $\Omega(a,b)$ is homothetic to $\mathbb{Z}\tau + \mathbb{Z}$ with τ transcendental, then $\mathrm{End}(E) = \mathbb{Z}$.

To understand what this means for the structure of the endomorphism ring, we perform a short detour.

1.5 Interlude: Orders in Quadratic Number Fields

Let k be a *number field*, that is a finite separable (and therefore algebraic) extension of $\mathbb{Q}$. If the extension $k/\mathbb{Q}$ is of degree 2, we call k a *quadratic* number field. We define its *ring of integers* $\mathcal{O}_k$ to be the integral closure of $\mathbb{Z} \subseteq k$.

From our study of endomorphisms of elliptic curves, we are primarily interested in imaginary quadratic number fields, that is number fields of the form $\mathbb{Q}(\sqrt{D})$ for $D < 0$ a squarefree integer. We note that this is not particularly restricting: every quadratic number field is of the form $\mathbb{Q}(\sqrt{D})$ for a unique squarefree integer D not $0, 1$. More formally stated, the quadratic number fields are exactly the splitting field of the polynomials $X^2 - D$ for squarefree integers D, not $0, 1$. If $D > 0$ we call $\mathbb{Q}(\sqrt{D})$ a *real* quadratic number field, else *imaginary* and we uncanonically choose $\sqrt{D}$ in $\iota\mathbb{R}$. The ring of integers of a quadratic number field $\mathbb{Q}(\sqrt{D})$ is given by [Cox13, Eq. 5.14, pg.92]

$$\mathcal{O}_D = \mathbb{Z}[w_D] = \mathbb{Z} + \mathbb{Z}w_D \quad \text{where} \quad w_D = \frac{d_D + \sqrt{d_D}}{2} \quad \text{and} \quad d_D = \begin{cases} D & \text{if } D \equiv 1 \pmod 4 \\ 4D & \text{else.} \end{cases}$$

is called the *discriminant* of k.

An *order* of a number field k is a subring $\mathcal{O}$, finitely generated as a $\mathbb{Z}$-module by a $\mathbb{Q}$-basis of k. Equivalently, it is a subring with $\mathcal{O} \otimes \mathbb{Q} = k$. If k is quadratic (not necessarily imaginary), we can classify the orders rather compactly in the following

Theorem 1.10 (Classification of orders in quadratic number fields [Sut21, Th. 13.27])**.** *The orders in $\mathbb{Q}(\sqrt{D})$ are precisely the rings $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_D$ with f a positive integer called the* conductor *and is equal to the index $[\mathcal{O} : \mathcal{O}_D]$.*

Importantly, the ring of integers $\mathcal{O}_D = \mathbb{Z} + 1 \cdot \mathcal{O}_D$ of $\mathbb{Q}(\sqrt{D})$ is the unique maximal order and so the index $[\mathcal{O} : \mathcal{O}_D]$ makes sense. Together with our description of the ring of integers, this means we may write any order $\mathcal{O}$ in $\mathbb{Q}(\sqrt{D})$ as

$$\mathcal{O} = \mathbb{Z} + f\mathcal{O}_D = \mathbb{Z} + f\mathbb{Z}w_D.$$

An immediate consequence is that any non-integer element $\alpha = a + bw_D$ of $\mathcal{O}_D$ generates an order $\mathcal{O} = \mathbb{Z}[\alpha] \subseteq \mathcal{O}_D$. Indeed,

$$\mathbb{Z}[\alpha] = \mathbb{Z} + \mathbb{Z}\alpha = \mathbb{Z} + \mathbb{Z}(a + bw_D) = \mathbb{Z} + b\mathbb{Z}w_D$$

is an order with conductor $|b|$. (Our condition that α is not an integer ensures $b \neq 0$).

1.6 Classification of Endomorphisms of Elliptic Curves II

We continue where we left off, before the interlude on orders.

From Equation 1.6, we see that $k = \mathbb{Q}(\tau)$ is an imaginary quadratic number field. Indeed, since τ lies in $\mathbb{H}$, $k \neq \mathbb{Q}$; and since τ satisfies a polynomial of degree two, $k/\mathbb{Q}$ must be an extension of degree two. Moreover, since $\operatorname{Im}(\tau) > 0$, we know it must be an imaginary extension. So let us write $\mathbb{Q}(\tau) = \mathbb{Q}(\sqrt{D})$ from now on.

We assumed $\operatorname{End}(E)$ contained a non-integer element α and showed that it is integral over $\mathbb{Z}$. Since it is integral and contained in $\mathbb{Q}(\tau) = \mathbb{Q}(\sqrt{D})$ (because $\alpha = m\tau + n$), we conclude that α lies in the ring of integers $\mathcal{O}_D$ of $\mathbb{Q}(\sqrt{D})$. As such, $\operatorname{End}(E)$ contains a subring which is isomorphic to $\mathbb{Z}[\alpha]$ which is an order in $\mathbb{Q}(\sqrt{D})$. Conversely, because α was an arbitrary non-integer in $\operatorname{End}(E)$, we have shown that $\operatorname{End}(E)$ is isomorphic to a subring of $\mathcal{O}_D$. More explicitly, we have shown that

$$\operatorname{End}(E) = \bigcup \{\mathbb{Z}[\beta] \mid \beta \in \operatorname{End}(E)\} \subseteq \mathcal{O}_D.$$

With $\alpha = a + bw_D$ as before, this gives us the following setup

$$\mathbb{Z} \subseteq \mathbb{Z}[\alpha] = \mathbb{Z} + b\mathbb{Z}w_D \subseteq \operatorname{End}(E) \subseteq \mathcal{O}_D = \mathbb{Z} + \mathbb{Z}w_D$$

This allows us to show that $\operatorname{End}(E)$ is in fact an order in $\mathcal{O}_D$. Indeed,

$$M = \{b' \mid a' + b'w_D \in \operatorname{End}(E), a', b' \in \mathbb{Z}\};$$

is clearly a (principal) ideal of $\mathbb{Z}$ and it is non-zero because it contains $b \neq 0$. So $M = f\mathbb{Z}$ for some non-zero integer f and $\operatorname{End}(E) = \mathbb{Z} + f\mathbb{Z}w_D$ is an order.

Hence, we have shown that the endomorphism ring of any elliptic curve over $\mathbb{C}$ is (isomorphic to) either $\mathbb{Z}$ or an order in an imaginary quadratic field. This kind of classification can be extended to elliptic curves over any field k and is given by

Theorem 1.11 (The Deuring Correspondence [Sil09, Cor. 9.4, p. 102]). *Let E be an elliptic curve defined over a field k of characteristic p. The ring* $\mathrm{End}(E)$ *is isomorphic to one of the following*

(i) (Only if $p = 0$) The integers; or

(ii) an order $\mathcal{O}$ in a imaginary quadratic field ordinary*; or*

(iii) (Only if $p \neq 0$) a maximal order in the quaternion algebra ramified at p and ∞.

If $p \neq 0$ and $\mathrm{End}(E/k)$ is an order (case (ii)), then we call E ordinary*, else we call it* supersingular*.*

1.7 The class-group action on elliptic curves with complex multiplication

We have now established that every elliptic curve $E/\mathbb{C}$ has and endomorphism ring $\mathrm{End}(E)$ which is either $\mathbb{Z}$ or an order $\mathcal{O}$ in an imaginary quadratic number field. Notably, orders are also lattices and so, when $E \cong E(a,b)$ has complex multiplication by $\mathrm{End}(E) = \mathcal{O}$, we can investigate the relationship between the lattice $\Omega = \Omega(a,b)$ corresponding to E and the lattice $\mathcal{O}$. In particular, given an order $\mathcal{O}$ we will find all the elliptic curves with complex multiplication by $\mathcal{O}$.

To forgo the additional step of going through elliptic curves, we will phrase elliptic curves only in terms of lattices and use the notation $\mathrm{End}(\Omega) = \{\alpha \in \mathbb{C} \mid \alpha\Omega \subseteq \Omega\}$ as introduced after (1.4). Moreover, to make the following more precise, for every imaginary quadratic number field $\mathbb{Q}(\sqrt{D})$ we choose the embedding into $\mathbb{C}$ so that $\sqrt{D} = \iota\sqrt{-D}$ in $\iota\mathbb{R}$.

We can reformulate the result of the first paragraph now as follows. We have the map

$$\begin{aligned}\{\text{Lattices } \Omega \subseteq \mathbb{C}\} &\to \{\mathbb{Z}\} \cup \{\mathcal{O} \mid \mathcal{O} \text{ an order in an imaginary quadratic field}\}\\ \Omega &\mapsto \mathrm{End}(\Omega).\end{aligned}$$

It is invariant under homothety as shown by a computation like (1.5). It is also surjective. The lattice $\Omega = \mathbb{Z}\pi + \mathbb{Z}$ has $\mathrm{End}(\Omega) = \mathbb{Z}$ because π is transcendental and (1.6) showed that $\mathrm{End}(\mathbb{Z}\zeta + \mathbb{Z}) \supsetneq \mathbb{Z}$ implied that ζ is algebraic. Else, let $\mathcal{O}$ be an order of an imaginary quadratic field. Then $\mathrm{End}(\mathcal{O}) = \mathcal{O}$. Indeed, if $\alpha\mathcal{O} \subseteq \mathcal{O}$, then α lies in $\mathcal{O}$ because $\mathcal{O}$ contains 1. Conversely, if α is in $\mathcal{O}$ then clearly $\alpha\mathcal{O} \subseteq \mathcal{O}$ because $\mathcal{O}$ is in particular a ring.

Finding all elliptic curves with complex multiplication by $\mathcal{O}$ is now equivalent to finding all lattices Ω (up to homothety) with $\mathrm{End}(\Omega) = \mathcal{O}$. To that end, let us write $\Omega = \mathbb{Z}\tau + \mathbb{Z}$ and $\mathcal{O} = \mathbb{Z}\zeta + \mathbb{Z}$. Since $\mathcal{O}$ is an order, we know that ζ is an algebraic integer. Since $\mathrm{End}(\Omega) = \mathcal{O}$ by assumption and Ω contains 1, we know that $\zeta \cdot 1$ lies in Ω and so $\zeta = k\tau + l$ for some integers k, l. As such $k\Omega = \mathbb{Z}k\tau + \mathbb{Z}k = \mathbb{Z}(\zeta - l) + \mathbb{Z}k = \mathbb{Z}\zeta + \mathbb{Z}l$ is a sublattice of index k. Moreover, because $\mathcal{O}$ is the endomorphism ring of Ω we have $\mathcal{O}\Omega \subseteq \Omega$ and in particular $\mathcal{O}k\Omega \subseteq k\Omega$. Hence $k\Omega \subseteq \mathcal{O}$ is an $\mathcal{O}$-ideal. To summarise, if Ω has $\mathrm{End}(\Omega) = \mathcal{O}$, then Ω is homothetic to an ideal of $\mathcal{O}$.

The converse is not necessarily true. That is, if $\Psi \subseteq \mathcal{O}$ is an ideal, then $\mathrm{End}(\Psi)$ must not necessarily be $\mathcal{O}$. It certainly contains $\mathcal{O}$ because Ψ is an $\mathcal{O}$-ideal, but can be also be a superset. In fact, when $\mathcal{O}$ is not maximal, so not the full ring of integers $\mathcal{O}_D$, then we can *always* find a Ψ so that $\mathrm{End}(\mathcal{O}) = \mathcal{O} \subsetneq \mathrm{End}(\Psi)$. We construct such a Ψ naïvely as follows. We know that $\mathcal{O} = \mathbb{Z} + f\mathbb{Z}w_D$ for some positive integer f, so any (additive) subgroup of $\mathcal{O}$ can be written as $\Psi = k\mathbb{Z} + lf\mathbb{Z}w_D$

for some integers k, l. If $l = k$, then Ψ is homothetic to $\mathcal{O}$ and has the same endomorphism ring, so we assume $l \neq k$. If Ψ is also to be an $\mathcal{O}$-ideal we require $(a + bw_D)(ck + dlfw_D)$ to lie in Ψ for all integers a, b, c, d. If we set $w_D^2 = A + Bw_D$ this gives us

$$(a + bw_D)(ck + dlfw_D) = ack + bdlfA + (bck + adlf + bdlfB)w_D$$

which lies in Ψ if and only if $k \mid lfA$ and $lf \mid k$. An example satisfying this is $k = f^2, l = f$. We verify that

$$\mathrm{End}(\Psi) = \mathrm{End}(f^2\mathbb{Z} + f^2\mathbb{Z}w_D) = \mathrm{End}(\mathbb{Z} + \mathbb{Z}w_D) = \mathcal{O}_D \supsetneq \mathcal{O}$$

as desired. It is worth noting that this example does not work when $f^2 = f$ (that is $f = 1$) and $\Psi = \mathcal{O} = \mathcal{O}_D$ is the maximal order.

We therefore distinguish $\mathcal{O}$-ideals Ψ with $\mathrm{End}(\Psi) = \mathcal{O}$ and call these *proper* $\mathcal{O}$-ideals. We summarise our discussion in the following. Let $\mathcal{O}$ be an order in an imaginary quadratic number field. The homothety classes of lattices $\Omega \subseteq \mathbb{C}$ with $\mathrm{End}(\Omega) = \mathcal{O}$ are represented by the proper ideals of $\mathcal{O}$. That is to say, that up to homothety, there is a bijection of lattices $\Omega \subseteq \mathbb{C}$ with $\mathrm{End}(\Omega) = \mathcal{O}$ and the proper ideals of $\mathcal{O}$.

This can be rephrased in the language of the ideal class group $\mathrm{Cl}(\mathcal{O})$ of $\mathcal{O}$. We briefly recall the definition of the ideal class group in general. Let A be a domain. A *fractional ideal* I of A is a A-submodule of $F = \mathrm{Frac}(A)$ for which there exists an element d so that $dI \subseteq A$. The product of two fractional ideals is a fractional ideal again. Clearly every *principal fractional ideal* $I = \alpha A$ for α in F is fractional. We say that a fractional ideal I is *invertible* if there exists another fractional ideal J so that $IJ = A$. Together with multiplication, the set of invertible fractional ideals form a group. Evidently, principal fractional ideals are invertible and form a subgroup. We define the *ideal class group* $\mathrm{Cl}(A)$ of A to be the quotient group of invertible fractional ideals modulo principal fractional ideals.

The important result for us now is that proper ideals of an order $\mathcal{O}$ in a quadratic number field coincide with invertible fractional ideals of $\mathcal{O}$ [Cox13, Prop. 7.4]. Moreover, if two proper $\mathcal{O}$-ideals $\Psi, \Psi' \subseteq \mathcal{O}$ are homothetic $\Psi' = \lambda\Psi$, then $\lambda = a/b$ with a, b in $\mathcal{O}$. Indeed, if $\Psi = \mathbb{Z}\psi_1 + \mathbb{Z}\psi_2$, then $\lambda\psi_1$ lies in Ψ' and so in $\mathcal{O}$ again. Therefore $\lambda = (\lambda\psi_1)/(\psi_1)$ is a fraction of two elements in $\mathcal{O}$. Rephrasing this, we see that two homothetic $\mathcal{O}$-ideals Ψ, Ψ' satisfy $a\Psi = b\Psi'$ for a, b in $\mathcal{O}$. This sets us up to understand how the set of proper $\mathcal{O}$-ideals modulo homothety is exactly the ideal class group $\mathrm{Cl}(\mathcal{O})$.

Indeed, knowing that the proper $\mathcal{O}$-ideals are the invertible fractional ideals of $\mathcal{O}$, we note that $a\Psi = b\Psi'$ holds true if and only if $a/b\mathcal{O} = \Psi'\Psi^{-1}$ which is a principal ideal. Hence two proper ideals Ψ, Ψ' are homothetic if and only if they are equal modulo principal invertible fractional ideals. Consequently

$$\{\text{proper } \mathcal{O}\text{-ideals}\}/_{\text{homothety}} \cong_{\text{sets}} \mathrm{Cl}(\mathcal{O}).$$

Together with what we know about the bijection of proper $\mathcal{O}$-ideals and elliptic curves with multiplication by $\mathcal{O}$ we can now state

Lemma 1.12. *Let $\mathcal{O}$ be an order in an imaginary quadratic number field. There is a bijection between $\mathrm{Ell}_{\mathbb{C}}(\mathcal{O})$ the isomorphism classes elliptic curves $E/\mathbb{C}$ with complex multiplication by*

$\mathcal{O}$ and the ideal class group $\mathrm{Cl}(\mathcal{O})$*. It is given by*

$$\mathrm{Cl}(\mathcal{O}) \to \mathrm{Ell}_{\mathbb{C}}(\mathcal{O}) \qquad \overline{\Omega(a,b)} \mapsto E(a,b)$$

Recall that $\Omega(a,b)$ is the lattice with Weierstrass invariants a, b and $E(a,b)$ is the elliptic curve cut out by $ZY^2 = 4X^3 - aXZ^2 - bZ^3$.

This delivers the important corollary

Corollary 1.13. *For any given order in an imaginary quadratic field,* $\mathrm{Ell}_{\mathbb{C}}(\mathcal{O})$ *is finite.*

This bijection shows us how to define a group action of $\mathrm{Cl}(\mathcal{O})$ acting on $\mathrm{Ell}_{\mathbb{C}}(\mathcal{O})$. Though, to do this, we introduce slightly different notation. Namely $E_{\mathfrak{a}}$ for the elliptic curve $E(a,b)$ corresponding to the lattice $\mathfrak{a} = \Omega(a,b)$. Of course $E_{\mathfrak{a}} = E_{\mathfrak{b}}$ for any homothetic lattices $\mathfrak{a}, \mathfrak{b}$, so in particular $E_{\mathfrak{a}}$ does not depend on the choice of representative $\mathfrak{a}$ of any (invertible) fractional ideal $\overline{\mathfrak{a}}$ in $\mathrm{Cl}(\mathcal{O})$. We define the group action

$$\mathrm{Cl}(\mathcal{O}) \times \mathrm{Ell}_{\mathbb{C}}(\mathcal{O}) \to \mathrm{Ell}_{\mathbb{C}}(\mathcal{O}) \qquad \overline{\mathfrak{a}}, E_{\mathfrak{b}} \mapsto E_{\mathfrak{a}^{-1}\mathfrak{b}}$$

It is free and transitive, making $\mathrm{Ell}_{\mathbb{C}}(\mathbb{C})$ a *principal homogeneous space* for the action of $\mathrm{Cl}(\mathcal{O})$. Indeed, $\overline{\mathfrak{a}} \cdot E_{\mathfrak{b}} = E_{\mathfrak{a}^{-1}\mathfrak{b}} = E_{\mathfrak{b}}$ can only occur if and only if $\mathfrak{b}, \mathfrak{a}^{-1}\mathfrak{b}$ are homothetic $\mathfrak{b} = \lambda\mathfrak{a}^{-1}\mathfrak{b}$. However, since $\mathfrak{b}$ is invertible, this means $\mathcal{O} = \lambda\mathfrak{a}^{-1}$ and $\mathfrak{a}$ is a principal (invertible fractional) ideal, and so the identity in $\mathrm{Cl}(\mathcal{O})$. In particular, $\overline{\mathfrak{a}} \cdot E_{\mathfrak{b}} = \overline{\mathfrak{c}} \cdot E_{\mathfrak{c}}$ if and only if $\overline{\mathfrak{a}} = \overline{\mathfrak{c}}$. Since $|\mathrm{Cl}(\mathcal{O})| = |\mathrm{Ell}_{\mathbb{C}}(\mathcal{O})|$ is finite, this implies the orbit of every element $E_{\mathfrak{b}}$ must be all of $\mathrm{Ell}_{\mathbb{C}}(\mathcal{O})$ and so the action is transitive.

It is conjectured that this group action can be used to construct quantum-secure cryptographic primitives [DeF17].

1.8 Complex-multiplication points and Binary quadratic forms

We saw in (1.6), that if the elliptic curve corresponding to the lattice $\mathbb{Z}\tau + \mathbb{Z}$ has complex multiplication, then τ in $\mathbb{C}$ is algebraic. It turns that the converse is also true. To illustrate this, we introduce binary quadratic forms.

Let a, b, c be coprime integers $\gcd(a,b,c) = 1$. We call functions of the form $Q\colon \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}; (x,y) \mapsto ax^2 + bxy + y$ *primitive binary quadratic forms*. We call the number $D = b^2 - 4ac$ the *discriminant* of the form Q and denote it by $\mathrm{disc}(Q)$. Moreover, we call a (quadratic binary) form Q *positive definite* if $Q(x,x) > 0$ for all integers x. Through direct calculation one can verify that

$$4aQ(x,y) = (2ax + by)^2 - Dy^2$$

and conclude that a form Q with negative discriminant $D < 0$ is positive definite if and only if $a > 0$.

Let $\gamma = (a,b;c,d)$ in $\mathrm{SL}_2(\mathbb{Z})$, then $(\gamma Q)(x,y) = Q(ax+by, cx+dy)$ defines a group action on the set of primitive binary quadratic forms $\mathcal{B}$, and so an equivalence relation on $\mathcal{B}$. Notably, it preserves the discriminant of the form, $\mathrm{disc}(\gamma Q) = \mathrm{disc}(Q)$, and so we can restrict the $\mathrm{SL}_2(\mathbb{Z})$-action to the set $\mathcal{B}_D$ of forms in $\mathcal{B}$ with discriminant D.

For $D \equiv 2, 3 \pmod 4$ $\mathcal{B}_D$ is empty. Indeed, for any form $b^2 - 4ac \equiv b^2 \pmod 4$, and the only squares modulo 4 are $0, 1$. Conversely, if $D \equiv 0 \pmod 4$, then $Q(x,y) = x^2 - Dy^2/4$ is a form with

discriminant D; else if $D \equiv 1 \pmod 4$, then $Q(x,y) = x^2 + xy - (D-1)y^2/4$ has discriminant D. Thus we call an integer $D \equiv 0, 1 \pmod 4$ a *fundamental* discriminant.

We note that τ in $\mathbb{H}$ satisfying the minimal monic polynomial $x^2 + rx + q$ with rational numbers r, q also satisfies the unique polynomial $ax^2 + bx + c$ for coprime integers a, b, c. Since τ is not real, we know that $D = b^2 - 4ac$ is negative, so $ac > 0$ and we may assume that $a, c > 0$ after multiplying a, b, c by -1 if necessary. So $Q_\tau(x,y) = ax^2 + bxy + cy^2$ is a positive definite primitive binary quadratic form with negative discriminant. It is unique, and so every algebraic number τ in $\mathbb{H}$ can be assigned a unique binary quadratic form Q_τ with $Q_\tau(\tau, 1) = 0$. Conversely, if Q in $\mathcal{B}$ with $Q(\tau, 1) = 0$, then $Q = Q_\tau$.

Moreover, the $\mathrm{SL}_2(Z)$ action on $\mathbb{H}$ is compatible with that on $\mathcal{B}_D$ in the following sense. If $\tau = \gamma\tau'$, then $Q_{\tau'} = \gamma Q_\tau$. Indeed, if $Q_\tau(x,y) = Ax^2 + Bxy + C$ and $\gamma = (a, b; c, d)$ then

$$\begin{aligned}
\gamma Q_\tau(\tau', 1) &= Q_\tau(a\tau' + b, c\tau' + d) \\
&= A(a\tau' + b)^2 + B(a\tau' + b)(c\tau' + d)C(c\tau' + d)^2 \\
&= (c\tau' + d)^{-2}\left(A\tau^2 + B\tau + C\right) \\
&= (c\tau' + d)^{-2} Q_\tau(\tau, 1) \\
&= 0
\end{aligned}$$

hence $\gamma Q_\tau = Q_{\tau'}$. So we may conclude

Lemma 1.14. *There is a bijective correspondence between* $\mathrm{SL}_2(\mathbb{Z})$ *classes of algebraic numbers in* $\mathbb{H}$ *and* $\mathrm{SL}_2(\mathbb{Z})$ *classes in* $\mathcal{B}_D$.

Before we relate binary quadratic forms to the ideal class group we need one more ingredient.

Let $\mathcal{O}$ be an order in an quadratic number field $\mathbb{Q}(\sqrt{D})$. Then $\mathcal{O} = \mathbb{Z} + f\mathbb{Z}w_D$ for some positive integer f and we define the *discriminant* of $\mathcal{O}$ as $\mathrm{disc}(\mathcal{O}) = f^2 d_D$. Recall that d_D is the field discriminant of $\mathbb{Q}(\sqrt{D})$. We note that there is only one order $\mathcal{O} \subseteq \mathbb{Q}(\sqrt{D})$ with discriminant $\mathrm{disc}(\mathcal{O})$. Since $d_D \equiv 0, 1 \pmod 4$ non-square, $\mathrm{disc}(\mathcal{O})$ is also a non-square $\mathrm{disc}(\mathcal{O}) \equiv 0, 1 \pmod 4$.

Conversely, every $D \equiv 0, 1 \pmod 4$ non-square defines an order in an quadratic number field. Indeed, let $D = f^2 d$ with d squarefree. We perform a short case distinction. If $D \equiv 0 \pmod 4$, then $f^2 \equiv 0 \pmod 4$ and we know nothing about $d \pmod 4$. So we perform another case distinction. If $d \equiv 1 \pmod 4$, then $d_d = d$ and $\mathbb{Z} + f\mathbb{Z}w_d$ is an order with discriminant $f^2 d_d = f^2 d = D$; else if $d \not\equiv 1 \pmod 4$, then $d_d = 4d$ and $\mathbb{Z} + (f/2)\mathbb{Z}w_d$ is an order with discriminant $(f/2)^2 d_d = (f^2/4)4d = f^2 d = D$. If $D \equiv 1 \pmod 4$, then things are easier. Since $f^2 \equiv 0, 1 \pmod 4$ as a square, we know $d \equiv 0, 1 \pmod 4$ so that $f^2 d = D \equiv 0, 1 \pmod 4$. However, since d is squarefree, we know $d \not\equiv 0 \pmod 4$. Hence $d \equiv 1 \pmod 4$ and we proceed as before. The order $\mathbb{Z} + f\mathbb{Z}w_d$ has discriminant $f^2 d_d = f^2 d = D$.

We now cite the important result relating binary quadratic forms and the ideal class group

Theorem 1.15 ([Cox13, Th. 7.7, pg. 123])**.** *Let k be a imaginary number field, and $\mathcal{O}$ be an order of discriminant D. Then*

$$\mathcal{B}_D / \mathrm{SL}_2(\mathbb{Z}) \to \mathrm{Cl}(\mathcal{O}) \qquad [ax^2 + bxy + cy^2] \mapsto \mathbb{Z}(-b + \sqrt{D})/2a + \mathbb{Z}$$

is bijection.

In fact, we can turn $\mathcal{B}_D/\mathrm{SL}_2(\mathbb{Z})$ into a group, called the *form class group* with the *Dirichlet composition* as introduced by Cox in the chapter *Composition and the Class Group* of [Cox13]. In *Zetafunktionen und Quadratische Körper*, Zagier gives a similarly explicit isomorphism in the reverse direction using ideal norms [Zag81, Eq. (15), pg. 92].

We collect all of our results in the following. Let τ be an algebraic number in $\mathbb{H}$. This yields a unique positive definite primitive binary quadratic form $Q_\tau(x,y) = ax^2+bxy+y^2$ of discriminant $D = b^2 - 4ac$. This in turn yields an invertible fractional ideal $\mathcal{O} = \mathbb{Z} + \mathbb{Z}(-b+\sqrt{D})/2a$ as in Theorem 1.15. We note that $\tau = (-b+\sqrt{D})/2a$ because $Q_\tau(\tau,1) = 0, \mathrm{Im}(\tau) > 0$ and so we may write the ideal as $\mathcal{O} = \mathbb{Z}\tau + \mathbb{Z}$. We know that invertible fractional ideals are proper ideals and so $\mathrm{End}(\mathcal{O}) = \mathcal{O}$. As such, $E_\mathcal{O}$ is an elliptic curve with complex multiplication by $\mathcal{O}$.

Conversely, for $D \equiv 0,1 \pmod 4$ negative, we call a point ζ_D in $\mathbb{H}$ a *complex multiplication point* or just *CM point*, if it is the root of a primitive binary quadratic form with discriminant D. In other words, if the minimal polynomial with coprime integral coefficients corresponding to ζ_D has discriminant D.

2
The Taylor expansion of modular forms

As described in the introduction, we want to find a different way of expressing the Taylor expansion of a modular form around a point z_0 in the upper-half complex plane $\mathbb{H}$ so that the domain of definition is the entirety of $\mathcal{H}$. Similarly to the development of the q-expansion of a modular form, we fix the issue of the domain of definition of the Taylor expansion by first sending the upper half-plane $\mathbb{H}$ to the unit disc $\mathbb{D}$ in a way that is somehow compatible with the $\mathrm{SL}_2(\mathbb{C})$ action on $\mathbb{H}$. We notice that

$$\rho_{z_0} = \frac{1}{z_0 - \overline{z_0}} \begin{pmatrix} 1 & -z_0 \\ 1 & -\overline{z_0} \end{pmatrix}$$

in $\mathrm{SL}_2(\mathbb{C})$ is a good candidate. Indeed, $\rho_{z_0} z_0 = 0$ and with $z = x + \imath y, z_0 = x_0 + \imath y_0$ we see that

$$\left|\rho_{z_0} z\right| = \left|\frac{z - z_0}{z - \overline{z_0}}\right| = \left|\frac{(x - x_0)^2 + (y - y_0)^2}{(x - x_0)^2 + (y + y_0)^2}\right| \leq 1$$

because $y, y_0 \geq 0$. In other words, ρ_{z_0} homeomorphically sends $\mathbb{H}$ to $\mathbb{D}$ whilst centring z_0 to zero.

However, for compatibility with the notation in [BGHZ04, OR12] we will work with its inverse

$$\sigma_{z_0} \overset{\text{def.}}{=} \rho_{z_0}^{-1} = \frac{1}{z_0 - \overline{z_0}} \begin{pmatrix} -\overline{z_0} & z_0 \\ -1 & 1 \end{pmatrix}.$$

Now we define the slash operator as usual

$$(f|_k \gamma)(z) = \det(\gamma)^{k/2} j(\gamma, z)^{-k} f(\gamma z)$$

and note that holomorphy of f implies holomorphy of $f|_k \sigma_{z_0}$ on $\mathbb{D}$. Hence $f|_k \sigma_{z_0}$ has a Taylor expansion at $z_0 = 0$ of the form

$$(f|_k \sigma_{z_0})(z) = \sum_{n \geq 0} \frac{(f|_k \sigma_{z_0})^{(n)}(0)}{n!} z^n = \sum_{n \geq 0} c_f(z_0, n) z^n. \tag{2.1}$$

We want to translate this Taylor expansion into one that is valid for f on the whole of $\mathbb{H}$. We begin by investigating

$$\begin{aligned}(f|_k \sigma_{z_0})(z) &= \det(\sigma_{z_0})^{k/2} j(\sigma_{z_0}, z)^{-k} f(\sigma_{z_0} z) \\ &= (z_0 - \overline{z_0})^{k/2} (-z + 1)^{-k} f(\sigma_{z_0} z).\end{aligned}$$

By evaluating at $\sigma_{z_0}^{-1} z = \rho_{z_0} z$ we can reverse this into expressing $f(z)$ as a function of $(f|_k \sigma_{z_0})(\sigma_{z_0}^{-1} z)$ for which we have the Taylor expansion (2.1). More precisely, we first compute

$$-\sigma_{z_0}^{-1} z + 1 = -\frac{z - z_0}{z - \overline{z_0}} + 1 = \frac{z - \overline{z_0} - z + z_0}{z - \overline{z_0}} = \frac{z_0 - \overline{z_0}}{z - \overline{z_0}}$$

and obtain

$$\begin{aligned}f(z) &= (z_0 - \overline{z_0})^{-k/2} (-\sigma_{z_0}^{-1} z + 1)^k (f|_k \sigma_{z_0})(\sigma_{z_0}^{-1} z) \\ &= (z_0 - \overline{z_0})^{k/2} (z - \overline{z_0})^{-k} (f|_k \sigma_{z_0})(\sigma_{z_0}^{-1} z)\end{aligned}$$

Substituting the Taylor expansion of $f|_k\sigma_{z_0}$ we find

$$f(z) = (z_0 - \overline{z_0})^{k/2}(z - \overline{z_0})^{-k} \sum_{n\geq 0} c_f(z_0, n)\left(\frac{z - z_0}{z - \overline{z_0}}\right)^n$$

we call this the *Taylor expansion* of f at z_0.

2.1 Trivial Taylor coefficients

It turns out that, that many of the Taylor coefficients are trivially zero.

Let Γ be a subgroup of $\mathrm{SL}_2(\mathbb{R})$ so that $\overline{\Gamma} \leqslant \mathrm{PSL}_2(\mathbb{R})$ is Fuchsian. For example $\Gamma = \mathrm{SL}_2(\mathbb{Z})$ or any congruence subgroup. Choose z_0 in $\mathbb{H} \cup \{\text{cusps}\}$ and set $\Gamma_{z_0} \leqslant \Gamma$ as the stabiliser of z_0. Then $\overline{\Gamma_{z_0}} \leqslant \mathrm{PSL}_2(\mathbb{R})$ is finite cyclic [Kat92, Cor 2.4.2,pg. 38], say of order N with a generator $\overline{\gamma_{z_0}}$.

From the definition of σ_{z_0}, we have that $\sigma_{z_0}0 = z_0$ and $\sigma_{z_0}\infty = \overline{z_0}$. Then $\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}$ fixes both 0 and ∞. Knowing which points a matrix fixes tells us a lot about the entries of the matrix in general. The matrix $(a, b; c, d)$ fixes $z \neq -d/c$ in $\mathbb{C}$ if and only if $cz^2 + (d - a)z - b = 0$. If the matrix is real, then it fixes $\overline{z}$ too. Notably, when $z = 0$ we have $b = 0$. Moreover $(a, b; c, d)$ fixes ∞ if and only if $c = 0$. Consequently, we have

$$\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0} = \begin{pmatrix}\zeta & 0\\ 0 & \omega\end{pmatrix} \quad \text{and} \quad \omega = \zeta^{-1} \quad \text{because} \quad \det\left(\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}\right) = 1.$$

In fact, by computing the whole product directly, we can verify that $\zeta = j(\gamma_{z_0}, \overline{z_0}) = c\overline{z_0} + d$. Let $\gamma_{z_0} = (a, b; c, d)$, then $\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}$ is

$$\frac{1}{z_0 - \overline{z_0}}\begin{pmatrix}1 & -z_0\\ 1 & \overline{z_0}\end{pmatrix}\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}-\overline{z_0} & z_0\\ -1 & 1\end{pmatrix} = \frac{1}{z_0 - \overline{z_0}}\begin{pmatrix}cz_0\overline{z_0} - a\overline{z_0} + dz_0 - b & -cz_0^2 - (d - a)z_0 + b\\ c\overline{z_0}^2 + (d - a)\overline{z_0} - b & -cz_0\overline{z_0} + az_0 - d\overline{z_0} + b\end{pmatrix}$$

and since γ_{z_0} has real entries, it fixes $\overline{z_0}$ too and we have $c\overline{z_0}^2 + (d - a)\overline{z_0} - b = 0$. Moreover, one can use this equality to verify

$$\zeta = \frac{cz_0\overline{z_0} - a\overline{z_0} + dz_0 - b}{z_0 - \overline{z_0}} = c\overline{z_0} + d = j(\gamma_{z_0}, \overline{z_0}).$$

To summarise, we know that $\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0} = (\zeta, 0; 0, \zeta^{-1})$ with $\zeta = j(\gamma_{z_0}, \overline{z_0})$; and the maps $\mathbb{H} \to \mathbb{H}$

$$z \mapsto \sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}z \qquad z \mapsto \overline{\sigma_{z_0}}^{-1}\overline{\gamma_{z_0}\sigma_{z_0}}z \qquad z \mapsto \zeta^2 z$$

are equal.

Importantly, we may now conclude that ζ^2 is a primitive N-th root of unity. Indeed,

$$\overline{\gamma_{z_0}} \text{ has order } N = \left|\overline{\Gamma_{z_0}}\right| \text{ implies that } \left(\overline{\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}}\right)^N = \overline{\sigma_{z_0}}^{-1}\overline{\gamma_{z_0}}^N\overline{\sigma_{z_0}} = \overline{\mathrm{id}} \text{ in } \mathrm{PSL}_2(\mathbb{C})$$

hence $(\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0})^N = \pm\mathrm{id}$ in $\mathrm{SL}_2(\mathbb{C})$ and so *acts* trivially. As such, we know that $z \mapsto \zeta^{2N}z$ must be the identity, which is the case only if $\zeta^{2N} = 1$; so we conclude that ζ^2 is an N-th root of unity. It is primitive, because

$$\text{if } (\zeta^2)^M = 1, \text{ then } \left(\overline{\sigma_{z_0}}^{-1}\overline{\gamma_{z_0}\sigma_{z_0}}\right)^M = \overline{\mathrm{id}}, \text{ equivalently } \overline{\gamma_{z_0}}^M = \overline{\mathrm{id}} \text{ and so } M \geq N.$$

Consequently we may write $\zeta^2 = e(m/N)$ with $\gcd(m, N) = 1$.

Writing the Taylor expansion 2.1 polar coordinates we obtain $f|_k\sigma_{z_0}(re(\theta)) = \sum_{n\geq 0} c_{z_0}(f, n)r^n e(\theta)$. Here we used the usual notation $e(\theta) = \exp(2\pi\imath\theta)$. Using a change of coordinates $\varphi\colon \mathbb{D} \to$

$\mathbb{D}; (\theta, r) \mapsto re(\theta)$ we obtain the 1-periodic function $f|_k\sigma_{z_0} \circ \varphi$. This has a Fourier expansion

$$\left(f|_k\sigma_{z_0} \circ \varphi\right)(\theta + \imath r) = f|_k\sigma_{z_0}(re(\theta)) = \sum_{n\in\mathbb{Z}} a_n e(n\theta) \quad \text{where} \quad a_n \overset{\text{def.}}{=} \int_0^1 (f|_k\sigma_{z_0} \circ \varphi)(\theta + \imath r)e(-n\theta)\,\mathrm{d}\,\theta.$$

Since f is of weight k and γ_{z_0} lies in Γ, we have

$$\begin{aligned}
\zeta^k f|_k\sigma_{z_0}(\zeta^2 z) &= j(\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}, z)^{-k} f|_k\sigma_{z_0}(\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0} z) \\
&= f|_k\sigma_{z_0}|_k\sigma_{z_0}^{-1}\gamma_{z_0}\sigma_{z_0}(z) \\
&= f|_k\gamma_{z_0}\sigma_{z_0}(z) \\
&= f|_k\gamma_{z_0}|_k\sigma_{z_0}(z) \\
&= f|_k\sigma_{z_0}(z)
\end{aligned}$$

and with $\zeta^2 = e(m/N) = e_N(m)$, equivalently $\zeta = e(m/(2N)) = e_{2N}(m)$, we have

$$\begin{aligned}
a_n &= \int_0^1 (f|_k\sigma_{z_0} \circ \varphi)(\theta + \imath r)e(-n\theta)\,\mathrm{d}\,\theta \\
&= \int_0^1 f|_k\sigma_{z_0}(re(\theta))e(-n\theta)\,\mathrm{d}\,\theta \\
&= \int_0^1 \zeta^k f|_k\sigma_{z_0}(\zeta^2 re(\theta))e(-n\theta)\,\mathrm{d}\,\theta \\
&= e_{2N}(mk)\int_0^1 f|_k\sigma_{z_0}(re(\theta + m/N))e(-n\theta)\,\mathrm{d}\,\theta \\
&\overset{\psi=\theta+m/N}{=} e_{2N}(mk)\int_{m/N}^{m/N+1} f|_k\sigma_{z_0}(re(\psi)e(-n\psi + nm/N))\,\mathrm{d}\,\psi \\
&= e_{2N}(mk)e(nm/N)\int_0^1 f|_k\sigma_{z_0}(re(\psi)e(-n\psi))\,\mathrm{d}\,\psi \\
&= e_{2N}(mk + 2nm)a_n.
\end{aligned}$$

Therefore, when

$$e_{2N}(mk + 2nm) \neq 1 \quad \text{which holds if and only if} \quad \frac{k}{2} + n \not\equiv 0 \pmod{N} \quad \text{because} \gcd(m, N) = 1$$

we must have $a_n = 0$. In other words, $a_n = 0$ unless $n \equiv k/2 \pmod{N}$. Since k and N are constants, depending on f and z_0 respectively, we see this imposes a periodicity condition on the a_n.

We summarise this section in the following

Lemma 2.1. *Let f be a modular form of weight k and level Γ. Let z_0 be a point in the upper-half complex plane $\mathbb{H}$ whose stabiliser $\Gamma_{z_0} \leqslant \Gamma$ has order N when viewed inside $\mathrm{PSL}_2(\mathbb{Z})$. The coefficients $c_f(z_0, n)$ in the Taylor expansion $f|_k\sigma_{z_0}(z) = \sum_{n\geq 0} c_f(z_0, n)z^n$ of f are zero when $n \not\equiv k/2 \pmod{N}$.*

2.2 Non-trivial Taylor coefficients

Our goal is now to understand the $c_f(z_0, n)$ from (2.1) better. By definition, we know that

$$c_f(z_0,n) = \frac{1}{n!}\partial^{(n)}\left[(f|_k\sigma_{z_0})(z)\right]_{z=0} = \frac{1}{n!}(z_0-\overline{z_0})^{k/2}\partial^{(n)}\left[(-z+1)^{-k}f(\sigma_{z_0}z)\right]_{z=0}$$

Now instead of manually computing the derivatives, we use the following idea from [IO08, Prop. 16]. Since f is holomorphic in a neighbourhood of z_0 we have the usual Taylor expansion

$$f(z) = \sum_{n\geq 0}\frac{f^{(n)}(z_0)}{n!}(z-z_0)^n$$

and since $(f|_k\sigma_{z_0})(z) = (z_0-\overline{z_0})^{k/2}(-z+1)^{-k}f(\sigma_{z_0}z)$ we have the following equality

$$\sum_{n\geq 0}c_f(z_0,n)z^n = (z_0-\overline{z_0})^{k/2}(-z+1)^{-k}\sum_{n\geq 0}\frac{f^{(n)}(z_0)}{n!}(\sigma_{z_0}z-z_0)^n.$$

After doing some quick algebraic housekeeping

$$\sigma_{z_0}z - z_0 = \frac{-\overline{z_0}z+z_0}{-z+1} - z_0 = \frac{-\overline{z_0}z+z_0+zz_0-z_0}{-z+1} = \frac{z(z_0-\overline{z_0})}{-z+1}$$

we write

$$\sum_{n\geq 0}c_f(z_0,n)z^n = \sum_{n\geq 0}\frac{f^{(n)}(z_0)}{n!}(z_0-\overline{z_0})^{k/2+n}z^n(-z+1)^{n-k}. \tag{2.2}$$

Now of course we would like to compare coefficients to express $c_f(z_0,n)$ as a function of $f^{(n)}(z_0)$. To do this, we must understand $z^n(-z+1)^{n+k}$. Formally, we know that

$$\frac{1}{-z+1} = \sum_{n\geq 0}z^n \quad \text{hence} \quad \left(\frac{1}{-z+1}\right)^j = \left(\sum_{n\geq 0}z^n\right)^j = \sum_{n\geq 0}\binom{j-1+n}{j-1}z^n$$

The last equality is an exercise in combinatorics and is easily proven using the *stars-and-bars* method: for every exponent $z^n = z\cdots z$ we must count the number of ways that we can choose a total of n objects from j brackets. Writing down

$$\star|\star|\star\star||\star\star|\star \quad \text{with} \quad (j-1) \text{ bars and } n \text{ stars}$$

gives us such a configuration; evidently there are $(j-1+n)$ choose $(j-1)$ ways of writing these configurations down.

Applying this to our coefficient comparison equation (2.2) we obtain

$$\sum_{n\geq 0}c_f(z_0,n)z^n = \sum_{n\geq 0}\frac{f^{(n)}(z_0)}{n!}(z_0-\overline{z_0})^{k/2+n}\sum_{l\geq 0}\binom{n+k-1+l}{n+k-1}z^{l+n}$$

and so

$$\begin{aligned} c_f(z_0, n) &= \sum_{s+r=n} \frac{f^{(s)}(z_0)}{s!}(z_0 - \overline{z_0})^{k/2+s} \binom{s+k-1+r}{r+k-1} \\ &= \sum_{s=0}^{n} \frac{f^{(s)}(z_0)}{s!}(z_0 - \overline{z_0})^{k/2+s} \binom{n+k-1}{s+k-1} \\ &= \sum_{s=0}^{n} \binom{n+k-1}{s+k-1} \frac{(z_0 - \overline{z_0})^{k/2+s}}{s!} f^{(s)}(z_0). \end{aligned}$$

The rearrangement in the final equality is to illustrate that up to some easily computed factors, we really have written the Taylor coefficients as a function of the derivatives.

For safe keeping we enshrine this as a numbered equation

$$c_f(n, z_0) = \sum_{s=0}^{n} \binom{n+k-1}{s+k-1} \frac{(z_0 - \overline{z_0})^{k/2+s}}{s!} f^{(s)}(z_0). \tag{2.3}$$

Since computing the derivative of a modular form is in general unwieldy (without already knowing the Fourier coefficients), we now develop a method for computing the derivatives recursively.

2.3 Computing derivatives of modular forms with recursive methods

As discussed in Chapter 5 of [BGHZ04], the usual derivative $D = (\partial_x - \iota\partial_y)/2$ preserves holomorphy but not modularity. Writing a modular form in its q-expansion $f = a_0 + a_1 q + a_2 q^2 + \cdots$ we see that $Df = 2\pi\iota(a_1 + a_2 q + a_3 q^2 + \cdots)$. To keep algebraicity properties of the Fourier coefficients a_i we will work with $\mathcal{D} = D/(2\pi\iota)$. We will also write f' for $\mathcal{D}f$

Nevertheless, $\mathcal{D}$ still does not preserve modularity. This can be remedied in different ways. One way, is to introduce the *Serre derivative* θ_k

$$M_k(\Gamma) \to M_{k+2}(\Gamma) \qquad f \mapsto \mathcal{D}f - \frac{k}{12} E_2 f.$$

It preserves both modularity and holomorphy, but not the weight. Another way to preserve modularity, is via the *Mass raising operator* ∂_k on the space M_k^* of meromorphic functions that transform like modular forms of weight k

$$M_k^*(\Gamma) \to M_{k+2}^* \qquad f \mapsto \mathcal{D}f - \frac{k}{4\pi y} f(z)$$

It clearly does not preserve holomorphy. For both the Serre derivative θ_k and the Maass raising operator ∂_k we will drop the subscripts where convenient.

Two inductive arguments show that we can express ∂ and $\mathcal{D}$ by sums of each other

$$\begin{aligned} \partial^n f = \sum_{s=0}^{n} (-1)^{n-s} \frac{n!}{s!} \binom{k+n-1}{k+s-1} \frac{1}{(4\pi y)^{n-s}} \mathcal{D}^s f \\ \text{and} \quad \mathcal{D}^n f = \sum_{s=0}^{n} \frac{n!}{s!} \binom{k+n-1}{k+s-1} \frac{1}{(4\pi y)^{n-s}} \partial^s f \end{aligned} \tag{2.4}$$

where is defined $\partial^n = \partial_{k+2n}\partial_{k+2(n-1)} \cdots \partial_k$ and $\mathcal{D}^n$ analogously.

In [ZV, Eq. 35; BGHZ04, Ch. 5.2] Zagier and Villegas introduce and motivate the recursively

defined *modified Serre derivative*

$$\theta^{[0]}(f) = f, \quad \theta^{[1]}(f) = \theta(f), \quad \theta^{[n+1]}(f) = \theta(\theta^{[n]}(f)) - n(k+n-1)\frac{E_4}{144}\theta^{[n-1]}(f)$$

where k is the weight of the modular form f. Moreover, we obtain

$$\mathcal{D}^n f = \sum_{s=0}^{n} \frac{n!}{s!}\binom{n+k-1}{s+k-1}\left(\frac{E_2}{12}\right)^{n-s}\theta^{[s]}(f) \tag{2.5}$$

which using the relations above (2.4) can be written as

$$\partial^n f = \sum_{s=0}^{n} \frac{n!}{s!}\binom{n+k-1}{s+k-1}\left(\frac{E_2^*}{12}\right)^{n-s}\theta^{[s]}(f) \tag{2.6}$$

as in [OR12, Eq. 4.3] where $E_2^* = E_2(z) - 3/\pi y$ is the modular Eisenstein series of weight 2. These formulations (2.5), (2.6) illustrate how the derivatives $\mathcal{D}^n, \partial^n$ can be very convenient functions of $\theta^{[n]}(f)$ when evaluated at zeros of E_2, E_2^* respectively.

We can clean up (2.4) for comparison with (2.3). Using $z - \overline{z} = 2\iota y$ and $\mathcal{D}^n f = (2\pi\iota)^{-n} f^{(n)}$ we obtain

$$\partial^n f(z_0) = (2\pi\iota)^{-n} n!(z_0 - \overline{z_0})^{-n} \sum_{s=0}^{n}\binom{n+k-1}{s+k-1}\frac{(z_0-\overline{z_0})^s}{s!}f^{(s)}(z)$$

which, when compared with (2.3), yields

$$c_f(n, z_0) = \frac{(2\pi\iota)^n}{n!}(z_0 - \overline{z_0})^{n+k/2}\partial^n f(z_0) \tag{2.7}$$

$$= \frac{(2\pi\iota)^n}{n!}(z_0 - \overline{z_0})^{n+k/2}\sum_{s=0}^{n}\frac{n!}{s!}\binom{n+k-1}{s+k-1}\left(\frac{E_2^*}{12}\right)^{n-s}\theta^{[s]}(f). \tag{2.8}$$

Consequently we have reduced computing the Fourier coefficients to evaluating the modified Serre derivative.

Before we continue to computing $\theta^{[n]}(f)$ we make some notes about θ itself. We are primarily interested in the j-function $j = {E_4}^3/\Delta$ and so we collect some easy results.

Lemma 2.2. *θ is zero on S_{12}.*

Proof. Since $\theta\colon S_k \to S_{k+2}$ and $S_{14} = 0$, we know that $\theta(f) = 0$ for all f in S_{12}. We note that this is equivalent to proving by manual computation that $\theta(\Delta) = 0$, since $S_{12} = \mathbb{C}\Delta$. □

Lemma 2.3. *The Serre Derivative θ is a derivation. That is, it satisfies the Leibniz rule.*

Proof. We simply compute

$$\begin{aligned}
-\theta(fg) &= \left(\frac{1}{3}R\partial_Q + \frac{1}{2}Q^2\partial_R\right)(fg)\\
&= \frac{1}{3}R\partial_Q(fg) + \frac{1}{2}Q^2\partial_R(fg)\\
&= \frac{1}{3}R\left(\partial_Q(f)g + f\partial_Q(g)\right) + \frac{1}{2}Q^2\left(\partial_R(f)g + f\partial_R(g)\right)\\
&= g\left(\frac{1}{3}R\partial_Q(f) + \frac{1}{2}Q^2\partial_R(f)\right) + f\left(\frac{1}{3}Rf\partial_Q(g) + \frac{1}{2}Q^2 f\partial_R(g)\right)\\
&= g\theta(f) + f\theta(g)
\end{aligned}$$

□

Corollary 2.4. *Let f lie in S_{12}. Then*

$$\theta(fg) = f\theta(g) \quad \textit{and} \quad \theta\left(\frac{g}{f}\right) = \frac{1}{f}\theta(g)$$

for any g in any M_k.

Proof. The first equality holds because θ is a derivation and $\theta(f) = 0$. For the second we compute

$$\begin{aligned}
-\theta\left(\frac{g}{f}\right) &= \left(\frac{1}{3}R\partial_Q + \frac{1}{2}Q^2\partial_R\right)\left(\frac{g}{f}\right) \\
&= \frac{1}{f^2}\left(\frac{1}{3}R\left(\partial_Q(g)f - g\partial_Q(f)\right) + \frac{1}{2}Q^2(\partial_R(g)f - g\partial_R(f))\right) \\
&= \frac{1}{f^2}\left(f\theta(g) - g\theta(f)\right) \\
&= -\frac{1}{f}\theta(g)
\end{aligned}$$

□

2.4 Computing the recursions

In this section we write $R = E_4$ and $Q = R_6$.

It is well-known that M_k is linearly generated by the monomials R^aQ^b where $k = 4a + 6b$. As such, we can formally write

$$Q^aR^b = Q^{k/4-3b/2}R^b = Q^{k/4}(Q^{-3/2}R)^b \quad \text{or} \quad Q^aR^b = Q^aR^{k/6-2a/3} = R^{k/6}(QR^{-2/3})^a$$

and any element f in M_k can be expressed as

$$f = \sum_{4a+6b=k} c_{a,b}Q^aR^b = Q^{k/4}\sum_{b=0}^{k} c_{a,b}(Q^{-3/2}R)^b = R^{k/6}\sum_{a=0}^{k} c_{a,b}(QR^{-2/3})^a.$$

In other words, for every f in M_k there exist univariate polynomials q, r in $\mathbb{C}[t]$ so that

$$f = Q^{k/4}r(Q^{-3/2}R) \quad \text{and} \quad f = R^{k/6}q(QR^{-2/3});$$

and writing T for $QR^{-2/3}$ and S for $Q^{-3/2}R$ we get the more succinct

$$f = Q^{k/4}r(S) \quad \text{and} \quad f = R^{k/6}q(T). \tag{2.9}$$

Each of these equalities holds everywhere except at roots of Q and R respectively. However, since R and Q do not share any roots, we can guarantee the existence of at least one such representation the neighbourhood of any point z_0 in $\mathbb{H}$. We also note that the coefficients $c_{a,b}$ expressing f as a linear combination of Q^aR^b terms are the same of the arising polynomials r, q. Hence if all $c_{a,b}$ are integral, then r, q are in fact elements of $\mathbb{Z}[t]$.

At this point we warn the reader. We Since we have these two different representations we will need to compute many expressions twice, once for each representation

Since our goal is to understand $\theta^{[n]}(f)$ in M_{k+2n} we can try to express the complicated recursion

defining $\theta^{[n]}$ as a simpler recursion in some univariate polynomials. More precisely, if we write

$$\theta^{[n]}(f) = Q^{(k+2n)/4} r_n(S) = R^{(k+2n)/6} q_n(T) \tag{2.10}$$

we might obtain a recursion in r_n and q_n that is easier to compute or understand.

Since

$$\begin{aligned}
\theta^{[n+1]}(f) &= \theta(\theta^{[n]}(f)) - n(n+k-1)\frac{Q}{144}\theta^{[n-1]}(f) \\
&= \theta(Q^{(k+2n)/4} r_n(S)) - n(n+k-1)\frac{1}{144} Q^{(k+2n+2)/4} r_{n-1}(S) \\
&= \theta(R^{(k+2n)/6} q_n(T)) - n(n+k-1)\frac{Q}{144} R^{(k+2n-2)/6} q_{n-1}(T)
\end{aligned}$$

we must first compute

$$\theta(R^{(k+2n)/6} q_n(T)) \quad \text{and} \quad \theta(Q^{(k+2n)/4} r_n(S)).$$

To that end we first compute with the monomial $m_l(t) = t^l$

$$\begin{aligned}
\partial_Q(m_l(T)) &= \partial_Q(Q^l R^{-2l/3}) = l Q^{l-1} R^{-2l/3} = l R^{-2/3} Q^{l-1} R^{-2(l-1)/3} \\
&= R^{-2/3} m_l'(T) \\
\partial_R(m_l(T)) &= \partial_R(Q^l R^{-2l/3}) \\
&= -\frac{2l}{3} Q^l R^{-2l/3-1} \\
&= -\frac{2l}{3} Q R^{-5/3} Q^{l-1} R^{-2(l-1)/3} \\
&= -\frac{2}{3} R^{-1} Q R^{-2/3} m_l'(T) \\
&= -\frac{2}{3} R^{-1} T m_l'(T) \\
\partial_Q(m_l(S)) &= \partial_Q(Q^{-3l/2} R^l) \\
&= -\frac{3l}{2} Q^{-3l/2-1} R^l \\
&= -\frac{3l}{2} R Q^{-5/2} Q^{-3(l-1)/2} R^{l-1} \\
&= -\frac{3}{2} R Q^{-5/2} m_l'(S) \\
&= -\frac{3}{2} Q^{-1} S m_l'(S) \\
\partial_R(m_l(S)) &= \partial_R(Q^{-3l/2} R^l) = l Q^{-3l/2} R^{l-1} = l Q^{-3/2} Q^{-3l/2} R^{l-1} \\
&= Q^{-3/2} m_l'(S)
\end{aligned}$$

Consequently

$$\partial_Q(q_n(T)) = R^{-2/3} q_n'(T) \quad \text{and} \quad \partial_R(q_n(T)) = -\frac{2}{3} R^{-1} T q_n'(T);$$

$$\partial_Q(r_n(S)) = -\frac{3}{2} Q^{-1} S r_n'(S) \quad \text{and} \quad \partial_R(r_n(S)) = Q^{-3/2} r_n'(S).$$

Now we compute

$$\begin{aligned}
&- \theta\left(R^{(k+2n)/6} q_n(T)\right) \\
&= \left(\frac{1}{3} R \partial_Q + \frac{1}{2} Q^2 \partial_R\right)\left(R^{(k+2n)/6} q_n(T)\right) \\
&= \frac{1}{3} R R^{(k+2n)/6} \partial_Q q_n(T) + \frac{1}{2} Q^2 \left[\frac{1}{6}(k+2n) R^{(k+2n-6)/6} q_n(T) + R^{(k+2n)/6} \partial_R(q_n(T))\right] \\
&= \frac{1}{3} R^{(k+2n+2)/6} q_n'(T) + \frac{1}{2} Q^2 \left[\frac{1}{6}(k+2n) R^{(k+2n-6)/6} q_n(T) - \frac{2}{3} R^{(k+2n-6)/6} T q_n'(T)\right] \\
&= R^{(k+2n-6)/6}\left(\frac{1}{3} R^{4/3} q_n'(T) + \frac{1}{2} Q^2 \left[\frac{1}{6}(k+2n) q_n(T) - \frac{2}{3} T q_n'(T)\right]\right) \\
&= R^{(k+2n-6)/6}\left[\frac{1}{12}(k+2n) Q^2 q_n(T) + \frac{1}{3}\left(R^{4/3} - Q^2 T\right) q_n'(T)\right]
\end{aligned}$$

to obtain

$$\begin{aligned}
\theta^{[n+1]}(f) = R^{(k+2n-6)/6}&\left[\frac{1}{3}\left(Q^2 T - R^{4/3}\right) q_n'(T) - \frac{1}{12}(k+2n) Q^2 q_n(T)\right] \\
&- n(n+k-1)\frac{Q}{144} R^{(k+2n-2)/6} q_{n-1}(T)
\end{aligned}$$

Finally substitute $\theta^{[n+1]}(f) = R^{(k+2n+2)/6} p_{n+1}(T)$ to obtain

$$\begin{aligned}
&p_{n+1}(T) \\
&= R^{-4/3}\left[\frac{1}{3}\left(Q^2 T - R^{4/3}\right) q_n'(T) - \frac{1}{12}(k+2n) Q^2 q_n(T)\right] - n(n+k-1)\frac{Q}{144} R^{-2/3} q_{n-1}(T) \\
&= \frac{1}{3}\left(Q^2 R^{-4/3} T - 1\right) q_n'(T) - \frac{1}{12}(k+2n) Q^2 R^{-4/3} q_n(T) - n(n+k-1)\frac{Q}{144} R^{-2/3} q_{n-1}(T) \\
&= \frac{1}{3}\left(T^3 - 1\right) q_n'(T) - \frac{1}{12}(k+2n) T^2 q_n(T) - n(n+k-1)\frac{1}{144} T q_{n-1}(T)
\end{aligned}$$

We now perform exactly the same computations on r_n.

$$\begin{aligned}
&- \theta\left(Q^{(k+2n)/4} r_n(S)\right) \\
&= \left(\frac{1}{3} R \partial_Q + \frac{1}{2} Q^2 \partial_R\right)\left(Q^{(k+2n)/4} r_n(S)\right) \\
&= \frac{1}{3} R\left[\frac{1}{4}(k+2n) Q^{(k+2n-4)/4} r_n(S) - \frac{3}{2} Q^{(k+2n)/4} Q^{-1} S r_n'(S)\right] \\
&\quad + \frac{1}{2} Q^2 Q^{(k+2n)/4} Q^{-3/2} r_n'(S) \\
&= \frac{1}{3} R\left[\frac{1}{4}(k+2n) Q^{(k+2n-4)/4} r_n(S) - \frac{3}{2} Q^{(k+2n-4)/4} S r_n'(S)\right] \\
&\quad + \frac{1}{2} Q^{(k+2n+2)/4} r_n'(S) \\
&= Q^{(k+2n-4)/4}\left(\frac{1}{3} R\left[\frac{1}{4}(k+2n) r_n(S) - \frac{3}{2} S r_n'(S)\right] + \frac{1}{2} Q^{3/2} r_n'(S)\right) \\
&= Q^{(k+2n-4)/4}\left[\frac{1}{12}(k+2n) R r_n(S) + \frac{1}{2}\left(Q^{3/2} - RS\right) r_n'(S)\right]
\end{aligned}$$

Then

$$\theta^{[n+1]}(f) = Q^{(k+2n-4)/4}\left[\frac{1}{2}\left(RS - Q^{3/2}\right)r_n'(S) - \frac{1}{12}(k+2n)Rr_n(S)\right] - n(n+k-1)\frac{1}{144}Q^{(k+2n+2)/4}r_{n-1}(S)$$

Finally substitute $\theta^{[n+1]}(f) = Q^{(k+2n+2)/4}p_{n+1}(S)$ to obtain

$$\begin{aligned}
&r_{n+1}(S)\\
&= Q^{-3/2}\left[\frac{1}{2}\left(RS - Q^{3/2}\right)r_n'(S) - \frac{1}{12}(k+2n)Rr_n(S)\right] - n(n+k-1)\frac{1}{144}r_{n-1}(S)\\
&= \frac{1}{2}\left(Q^{-3/2}RS - 1\right)r_n'(S) - \frac{1}{12}(k+2n)Q^{-3/2}Rr_n(S) - n(n+k-1)\frac{1}{144}r_{n-1}(S)\\
&= \frac{1}{2}\left(S^2 - 1\right)r_n'(S) - \frac{1}{12}(k+2n)Sr_n(S) - n(n+k-1)\frac{1}{144}r_{n-1}(S)
\end{aligned}$$

These recursions can be given integral coefficients, by replacing $Q_n = 12^n q_n$ and $R_n = 12^n r_n$. Indeed

$$\begin{aligned}
Q_{n+1}(T) &= 12^{n+1}q_{n+1}(T)\\
&= 4(T^3-1)(12^n q_n'(T))\\
&\quad - (k+2n)T^2(12^n q_n(T))\\
&\quad - n(n+k-1)T(12^{n-1}q_{n-1}(T))\\
&= 4(T^3-1)Q_n'(T) - (k+2n)T^2Q_n(T) - n(n+k-1)TQ_{n-1}(T)
\end{aligned}$$

likewise

$$\begin{aligned}
R_{n+1}(S) &= 12^{n+1}r_{n+1}(S)\\
&= 6(S^2-1)(12^n r_n'(S))\\
&\quad - (k+2n)S(12^n r_n(S))\\
&\quad - n(n+k-1)(12^{n-1}r_{n-1}(S))\\
&= 6(S^2-1)R_n'(T) - (k+2n)SR_n(S) - n(n+k-1)R_{n-1}(S)
\end{aligned}$$

2.5 O'Sullivan-Risager's method for developing recursions in S_{12}

These recursions are true for all f in M_k for any k. O'Sullivan and Risager make a special note of the case $f = \Delta$ in $S_{12} = \mathbb{C}\Delta$ with the following ideas. We have seen that $\theta(fg) = f\theta(g)$ for f in S_{12} and g in any M_k, and so Δ divides all $\theta^{[n]}(\Delta)$. Indeed, we begin with with $\theta^{[0]}(\Delta) = \Delta$ and

$\theta^{[1]}(\Delta) = \theta(\Delta) = 0$ (both are divisible by Δ) and inductively verify

$$\begin{aligned}\theta^{[n+1]}(\Delta) &= \theta(\theta^{[n]}(\Delta)) - n(n+k-1)\frac{Q}{144}\theta^{[n-1]}(\Delta) \\ &= \theta\left(\Delta\frac{\theta^{[n]}(\Delta)}{\Delta}\right) - n(n+k-1)\frac{Q}{144}\Delta\frac{\theta^{[n-1]}(\Delta)}{\Delta} \\ &= \Delta\theta\left(\frac{\theta^{[n]}(\Delta)}{\Delta}\right) - n(n+k-1)\frac{Q}{144}\Delta\frac{\theta^{[n-1]}(\Delta)}{\Delta} \\ &= \Delta\left[\theta\left(\frac{\theta^{[n]}(\Delta)}{\Delta}\right) - n(n+k-1)\frac{Q}{144}\frac{\theta^{[n-1]}(\Delta)}{\Delta}\right].\end{aligned}$$

Consequently, we can define a new recursively defined operator

$$\psi^{[n]}(\Delta) = \frac{1}{\Delta}\theta^{[n]}(\Delta) \quad \in M_{2n}$$

which is convenient because $\psi^{[0]}(\Delta) = 1, \psi^{[1]}(\Delta) = 0$ are less complicated terms. By definition $\psi^{[n]}$ satisfies the same recursion as $\theta^{[n]}$

$$\begin{aligned}\psi^{[n+1]}(\Delta) &= \frac{1}{\Delta}\theta^{[n+1]}(\Delta) \\ &= \frac{1}{\Delta}\theta(\theta^{[n]}(\Delta)) - n(n+k-1)\frac{Q}{144}\frac{1}{\Delta}\theta^{[n-1]}(\Delta) \\ &= \theta\left(\frac{1}{\Delta}\theta^{[n]}(\Delta)\right) - n(n+k-1)\frac{Q}{144}\frac{1}{\Delta}\theta^{[n-1]}(\Delta) \\ &= \theta\left(\psi^{[n]}(\Delta)\right) - n(n+k-1)\frac{Q}{144}\psi^{[n-1]}(\Delta) \qquad (2.11)\end{aligned}$$

This introduces some subtlety: Since $\psi^{[n]}(\Delta)$ lies in M_{2n} we have on the one hand $\psi^{[n]}(\Delta) = Q^{n/2}r(S) = R^{n/3}q(T)$ from (2.9). In a sense, we have set $k = 0$ in this case. However, on the other hand, the recursion is still valid for $k = 12$ as inherited from the recursion in $\theta^{[n]}$. This means we cannot simply look at the recursions of R_n, Q_n and set $k = 0$ just because the starting function f has weight 0.

Now we can write

$$\psi^{[n]}(\Delta) = Q^{n/2}\tilde{r}_n(S) = R^{n/3}\tilde{q}_n(T)$$

as before, and compute $\tilde{r}_n, \tilde{q}_n$ using the same methods. Firstly

$$-\theta(R^{n/3}\tilde{q}_n(T)) = R^{(n-3)/3}\left[\frac{1}{6}nQ^2\tilde{q}_n(T) + \frac{1}{3}\left(R^{4/3} - Q^2T\right)\tilde{q}_n'(T)\right]$$

so

$$\psi^{[n+1]}(\Delta) = R^{(n-3)/3}\left[\frac{1}{6}nQ^2\tilde{q}_n(T) + \frac{1}{3}\left(Q^2T - R^{4/3}\right)\tilde{q}_n'(T)\right] - n(n+11)\frac{Q}{144}R^{(n-1)/3}\tilde{q}_{n-1}(T)$$

and substituting $\psi^{[n+1]} = R^{(n+1)/3}\tilde{q}_{n+1}(T)$ we obtain

$$\begin{aligned}\tilde{q}_{n+1}(T) &= R^{-4/3}\left[\frac{1}{3}\left(Q^2T - R^{4/3}\right)\tilde{q}_n'(T) - \frac{1}{6}nQ^2\tilde{q}_n(T)\right] - n(n+11)\frac{Q}{144}R^{-2/3}\tilde{q}_{n-1}(T)\\ &= \frac{1}{3}\left(Q^2R^{-4/3}T - 1\right)\tilde{q}_n'(T) - \frac{1}{6}nQ^2R^{-4/3}\tilde{q}_n(T) - n(n+11)\frac{Q}{144}R^{-2/3}\tilde{q}_{n-1}(T)\\ &= \frac{1}{3}\left(T^3 - 1\right)\tilde{q}_{n-1}(T) - \frac{1}{6}nT^2\tilde{q}_n(T) - n(n+11)\frac{1}{144}T\tilde{q}_{n-1}(T)\end{aligned}$$

Likewise

$$-\theta(Q^{n/2}\tilde{r}_n(S)) = Q^{(n-2)/2}\left[\frac{1}{6}nR\tilde{r}_n(S) + \frac{1}{2}(Q^{3/2} - RS)\tilde{r}_n'(S)\right]$$

so

$$\psi^{[n+1]}(\Delta) = Q^{(n-2)/2}\left[\frac{1}{2}(RS - Q^{3/2})\tilde{r}_n'(S) - \frac{1}{6}nR\tilde{r}_n(S)\right] - n(n+11)\frac{1}{144}Q^{(n+1)/4}\tilde{r}_{n-1}(S)$$

and substituting $\psi^{[n+1]} = Q^{(n+1)/2}\tilde{r}_{n+1}(S)$ we garner

$$\begin{aligned}\tilde{r}_{n+1}(S) &= Q^{-3/2}\left[\frac{1}{2}(RS - Q^{3/2})\tilde{r}_n'(S) - \frac{1}{6}nR\tilde{r}_n(S)\right] - n(n+11)\frac{1}{144}\tilde{r}_{n-1}(S)\\ &= \frac{1}{2}(Q^{-3/2}RS - 1)\tilde{r}_n'(S) - \frac{1}{6}nQ^{-3/2}R\tilde{r}_n(S) - n(n+11)\frac{1}{144}\tilde{r}_{n-1}(S)\\ &= \frac{1}{2}(S^2 - 1)\tilde{r}_n'(S) - \frac{1}{6}nS\tilde{r}_n(S) - n(n+11)\frac{1}{144}\tilde{r}_{n-1}(S).\end{aligned}$$

As before, we can also replace $\tilde{Q}_n = 12^n\tilde{q}_n$ and $\tilde{R}_n = 12^n\tilde{r}_n$ to obtain the integral recursions

$$\begin{aligned}\tilde{Q}_{n+1}(T) &= 4\left(T^3 - 1\right)\tilde{Q}_n'(T) - 2nT^2\tilde{Q}_n(T) - n(n+11)T\tilde{Q}_{n-1}(T)\\ \tilde{R}_{n+1}(S) &= 6(S^2 - 1)\tilde{R}_n'(S) - 2nS\tilde{R}_n(S) - n(n+11)\tilde{R}_{n-1}(S).\end{aligned}$$

2.6 Summary of recursions

For completeness, and ease of comparison with other literature we have included both $r_{n+1} = \ldots$ and $r_n = \ldots$ recursions.

(i) General case

$$\begin{aligned}
q_{n+1}(T) &= \frac{1}{3}\left(T^3-1\right)q_n'(T) - \frac{1}{12}(k+2n)T^2 q_n(T) - n(n+k-1)\frac{1}{144}Tq_{n-1}(T)\\
r_{n+1}(S) &= \frac{1}{2}\left(S^2-1\right)r_n'(S) - \frac{1}{12}(k+2n)Sr_n(S) - n(n+k-1)\frac{1}{144}r_{n-1}(S)\\
q_{n}(T) &= \frac{1}{3}\left(T^3-1\right)q_{n-1}'(T) - \frac{1}{12}(k+2n-2)T^2 q_{n-1}(T) - (n-1)(n+k-2)\frac{1}{144}Tq_{n-2}(T)\\
r_{n}(S) &= \frac{1}{2}\left(S^2-1\right)r_{n-1}'(S) - \frac{1}{12}(k+2n-2)Sr_{n-1}(S) - (n-1)(n+k-2)\frac{1}{144}r_{n-2}(S)
\end{aligned}$$

(ii) General integral case ($Q_n = 12^n q_n, R_n = 12^n r_n$)

$$\begin{aligned}
Q_{n+1}(T) &= 4\left(T^3-1\right)Q_n'(T) - (k+2n)T^2 Q_n(T) - n(n+k-1)TQ_{n-1}(T)\\
R_{n+1}(S) &= 6\left(S^2-1\right)R_n'(S) - (k+2n)SR_n(S) - n(n+k-1)R_{n-1}(S)\\
Q_{n}(T) &= 4\left(T^3-1\right)Q_{n-1}'(T) - (k+2n-2)T^2 Q_{n-1}(T) - (n-1)(n+k-2)TQ_{n-2}(T)\\
R_{n}(S) &= 6\left(S^2-1\right)R_{n-1}'(S) - (k+2n-2)SR_{n-1}(S) - (n-1)(n+k-2)R_{n-2}(S)
\end{aligned}$$

(iii) Case $k = 12$

$$\begin{aligned}
Q_{n+1}(T) &= 4\left(T^3-1\right)Q_n'(T) - (12+2n)T^2 Q_n(T) - n(n+11)TQ_{n-1}(T)\\
R_{n+1}(S) &= 6\left(S^2-1\right)R_n'(S) - (12+2n)SR_n(S) - n(n+11)R_{n-1}(S)\\
Q_{n}(T) &= 4\left(T^3-1\right)Q_{n-1}'(T) - (2n+10)T^2 Q_{n-1}(T) - (n-1)(n+10)TQ_{n-2}(T)\\
R_{n}(S) &= 6\left(S^2-1\right)R_{n-1}'(S) - (2n+10)SR_{n-1}(S) - (n-1)(n+10)R_{n-2}(S)
\end{aligned}$$

(iv) O'Sullivan-Risager formulation when $f = \Delta$

$$\begin{aligned}
\tilde{q}_{n+1}(T) &= \frac{1}{3}\left(T^3-1\right)\tilde{q}_n'(T) - \frac{1}{6}nT^2\tilde{q}_n(T) - n(n+11)\frac{1}{144}T\tilde{q}_{n-1}(T)\\
\tilde{r}_{n+1}(S) &= \frac{1}{2}(S^2-1)\tilde{r}_n'(S) - \frac{1}{6}nS\tilde{r}_n(S) - n(n+11)\frac{1}{144}\tilde{r}_{n-1}(S)\\
\tilde{q}_{n}(T) &= \frac{1}{3}\left(T^3-1\right)\tilde{q}_{n-1}'(T) - \frac{1}{6}(n-1)T^2\tilde{q}_{n-1}(T) - (n-1)(n+10)\frac{1}{144}T\tilde{q}_{n-2}(T)\\
\tilde{r}_{n}(S) &= \frac{1}{2}(S^2-1)\tilde{r}_{n-1}'(S) - \frac{1}{6}(n-1)S\tilde{r}_{n-1}(S) - (n-1)(n+10)\frac{1}{144}\tilde{r}_{n-2}(S)
\end{aligned}$$

(v) O'Sullivan-Risager formulation when $f = \Delta$ integral case ($\tilde{Q}_n = 12^n\tilde{q}_n, \tilde{R}_n = 12^n\tilde{r}_n$)

$$\begin{aligned}
\tilde{Q}_{n+1}(T) &= 4\left(T^3-1\right)\tilde{Q}_n'(T) - 2nT^2\tilde{Q}_n(T) - n(n+11)T\tilde{Q}_{n-2}(T)\\
\tilde{R}_{n+1}(S) &= 6(S^2-1)\tilde{R}_n'(S) - 2nS\tilde{R}_n(S) - n(n+11)\tilde{R}_{n-2}(S)\\
\tilde{Q}_{n}(T) &= 4\left(T^3-1\right)\tilde{Q}_{n-1}'(T) - 2(n-1)T^2\tilde{Q}_{n-1}(T) - (n-1)(n+10)T\tilde{Q}_{n-2}(T)\\
\tilde{R}_{n}(S) &= 6(S^2-1)\tilde{R}_{n-1}'(S) - 2(n-1)S\tilde{R}_{n-1}(S) - (n-1)(n+10)\tilde{R}_{n-2}(S)
\end{aligned}$$

2.7 A final expression

We can now piece together our results. Equations (2.7) and (2.10) together yield

$$c_f(n,z_0) = \frac{(2\pi\iota)^n}{n!}(z_0-\overline{z_0})^{n+k/2}\sum_{s=0}^{n}\frac{n!}{s!}\binom{n+k-1}{s+k-1}\left(\frac{E_2^*}{12}\right)^{n-s}Q^{(k+2s)/4}r_s(Q^{-3/2}R)$$
$$= \frac{(2\pi\iota)^n}{n!}(z_0-\overline{z_0})^{n+k/2}\sum_{s=0}^{n}\frac{n!}{s!}\binom{n+k-1}{s+k-1}\left(\frac{E_2^*}{12}\right)^{n-s}R^{(k+2s)/6}q_s(QR^{-2/3})$$

depending on whether we are evaluating at a root of $R = E_4$ or $Q = E_6$ (or neither, and either expression can be used). The recursively defined polynomials r_s, q_s are dependent on f by definition.

These expressions become a lot more manageable when evaluated at a zero of E_2^*.

$$c_f(n,z_0) = \frac{(2\pi\iota)^n}{n!}(z_0-\overline{z_0})^{n+k/2}Q^{(k+2n)/4}r_n(Q^{-3/2}R)$$
$$= \frac{(2\pi\iota)^n}{n!}(z_0-\overline{z_0})^{n+k/2}R^{(k+2n)/6}q_n(QR^{-2/3})$$

Using the table on Page 87 of [BGHZ04], we see that ζ_D is a root of E_2^* for $D=-3,-4$. Here $\zeta_{-3} = \exp(2\pi\iota/3) = \rho$, and $\zeta_{-4} = \iota$. These CM points are unique, since the class number is 1 in both cases; a fact which be proven using the method described by Zagier in the section *Finitness of Class Numbers* [BGHZ04]. In fact, we also see that in the table, that $E_4(\rho)=0$ and $E_6(\iota)=0$. Hence, we obtain

$$c_f(n,\rho) = \frac{(2\pi\iota)^n}{n!}\left(\sqrt{3}\right)^{n+k/2}\left(2\sqrt{-D}\Omega_D^6\right)^{(k+2n)/4}r_n(0) \tag{2.12}$$

$$c_f(n,\iota) = \frac{(2\pi\iota)^n}{n!}2^{n+k/2}\left(\Omega_D^4\right)^{(k+2n)/6}q_n(0). \tag{2.13}$$

We note that these expressions look different to those of O'Sullivan Risager, since they used the alternative method of developing recursions as catalogued in Subsection 2.5. The value Ω_D is called the *Chowla-Selberg* period and defined in Proposition 26 of [BGHZ04].

Finally we remark on expansions at other CM points. Of course the same expressions are valid, yet cumbersome, since the sum does not collapse. O'Sullivan and Risager detail how the method may be altered for arbitrary CM points in Theorem 5.3 of [OR12].

3
Algorithms to determine periodic and non-vanishing behaviour

3.1 General strategy

O'Sullivan-Risager compute the Fourier expansion of Δ at ι and remark that the polynomials p_n, and therefore their evaluation at $p_n(0)$, are periodic modulo 5 [OR12, Prop. 5.1, pg.9]. They go on to prove that this is behaviour is typical in Theorem 6.1: given any modular form f, a CM point ζ and a prime d, the values $p_n(0) \pmod d$ of the $p_n(t)$ corresponding to f expanded at ζ are eventually periodic.

Using similar techniques to O'Sullivan-Risager in Subsection 3.2, we develop the necessary theory to prove in Subsection 3.3 that for $f(t)$ arbitrary and $F_d(t) = f(t)^d$, the sequence of residues $p_n(t) \pmod{d, F_d(t)}$ is eventually periodic. In particular, if $F(0) = 0$, then the residues $p_n(0) \pmod{d, F_d(t)}$ are non-zero only if the values $p_n(0)$ are non-zero, and so the values $p_n(0) \pmod{d, F_d(t)}$ are appropriate for determining non-vanishing behaviour.

Indeed, we observed in (2.12) that the Fourier coefficients are given by $c_f(n, \zeta) = ab^n p_n(0)$ for non-zero constants a, b (depending on f, ζ). So if we know that $p_n(0) \pmod{d, F_d(t)}$ is periodic, then we can verify whether $p_n(0) \not\equiv 0 \pmod{d, F_d(t)}$ for all non-trivial values by simply verification over just one period. In particular then the non-trivial $p_n(0)$ are non-zero, making the non-trivial Fourier coefficients non-zero.

We note that the recursions of interest (Subsection 2.6) all involve the derivative of a previous term and so we must work with the full polynomials $p_n(t)$, and cannot simply write down a recursion for the $p_n(0)$ directly.

In practical terms, this means we must

(i) choose a prime candidate d and a polynomial $f(t)$ with $f(0) = 0$;

(ii) compute *enough* residues $p_0(t), p_1(t), \dots, p_K(t)$ modulo $d, F_d(t) = f(t)^d$;

(iii) use these $K + 1$ polynomials to determine a period p and offset b of the full sequence $p_0(t), p_1(t), \dots, p_K(t), p_{K+1}(t), p_{K+2}(t), \dots$ modulo $d, F_d(t)$;

(iv) and then verify whether all non-trivial values $p_n(0) \pmod{d, F_d(t)}$ are non-zero over one full period.

(v) If all non-trivial values $p_n(0) \pmod{d, F_d(t)}$ are non-zero, then $p_n(0)$ are non-zero in general and we have proven that the Fourier coefficients are eventually non-vanishing. Else, we must choose a new prime candidate d' and start again.

The question of how many polynomials $p_n(t) \pmod{d, F_d(t)}$ we must compute in step (ii) before we can detect periodicity is answered in Subsection 3.4.

3.2 Periodic behaviour of recursively defined sequences

Before developing an algorithm for computing the periodic and non-vanishing behaviour of $p_n(t) \pmod{d, F_d(t)}$ we first introduce some definitions. They are likely not standard across literature.

We say a sequence $a(n), n \geq 0$ in the set X is *recursive* of *depth* r if there exists a function $R\colon \mathbb{N}_0 \times X^r \to X$, such that $a(n) = R(n, a(n-1), \dots, a(n-r))$ for all $n > r$. In this case, we call R an *r-term recursion*. If R does not depend on n, then we say $a(n)$ is recursive of *effective* depth r, and R an *effective* r-term recursion. Finally, if there exists an integer d so that $R(n, \cdot)$ depends only on n modulo d (and the previous r terms $a(n-1), \dots, a(n-r)$), then we say $a(n)$ is recursive of *pseudo-effective* depth r and *height* d. That is, $a(n)$ is recursive of pseudo-effective depth r and height d if

$$R(n, a(n-1), \dots, a(n-r)) = R(n+dk, a(n-1), \dots, a(n-r))$$

for all integral k with $n + dk \geq 0$. In this case we call R a *pseudo-effective r-term recursion of height d*.

An example for a recursively defined sequence with depth 1 is the factorial sequence $a(n) = R(n, a(n-1)) = na(n-1), a(0) = 1$. This sequence does not have effective depth 1. Given the value of an element a_n (e.g. 720) without knowing the index, we cannot compute the next element in the sequence. However, it is of *effective* depth 2. Given the last two terms $a(n-1), a(n-2)$ (e.g. $120, 720$) one can compute $a(n) = S(n, a(n-1), a(n-2)) = (a(n-1)/a(n-2)+1)a(n-1)$. We also note that this example illustrates that knowing that *a particular* recursion describing $a(n)$ is not effective, does not mean that *every* recursion describing $a(n)$ is not effective. In the factorial example, R was not effective, but S was. Another example for a recursively defined sequence with effective depth 2 is the Fibonacci sequence $a(n) = R(n, a(n-1), a(n-2)) = a(n-1)+a(n-2), n > 1$. Given only the previous two terms (e.g. $a(6) = 5, a(7) = 8$), we can compute the next term ($a(8) = 5 + 8 = 13$) without knowing the index (8).

The recursions in Subsection 2.6 are of depth 2, but not effective depth 2: they depend on the last two terms and the current index n. For example, using the O'Sullivan-Risager formulation when expanding around ι, we have the recursion

$$\tilde{R}_n(S) = 6(S^2-1)\tilde{R}'_{n-1}(S) - 2(n-1)S\tilde{R}_{n-1}(S) - (n-1)(n+10)\tilde{R}_{n-2}(S). \tag{3.1}$$

Important for us, is that modulo d these recursions are of pseudo-effective depth 2 and height d.

To make use of this, we now develop some results on periodic behaviour

Lemma 3.1. *Let $a(n)$ be a recursively defined sequence of pseudo-effective depth r and height d. If there exist indices $M < N$ with $M \equiv N \pmod{d}$ so that*

$$a(M) = a(N), a(M+1) = a(N+1), \dots, a(M+r-1) = a(N+r-1) \tag{3.2}$$

Then $a(n)$ is $(N-M)$-periodic with offset M.

Proof. The proof is very short. We simply write

$$a(M+r) = R(M+r, a(M+r-1), \dots, a(M)) = R(N+r, a(N+r-1), \dots, a(N)) = a(N+r).$$

where R is the pseudo-effective recursion describing $a(n)$. The middle equality holds, because $M + r \equiv N + r \pmod{d}$ and R is pseudo-effective of height d. Now we may inductively proceed with $M' = M+1, N' = N+1$ until we arrive at $M' = N$. Then we will have shown that $a(M+k) = a(N+k)$ for all $k = 0, \dots, N-M$. Importantly, $M' = N < N' = 2N - M$ satisfy the requirement of the lemma again. □

We can now also *prove* that the recursion (3.1) is indeed not of effective depth 2 for any recursion R describing $\tilde{R}_n$. We see from Equation 5.2 of [OR12] that the sequence $1, 2t$ repeats twice with $p_4(t) \equiv p_8(t) \equiv 1 \pmod 5$ and $p_5(t) \equiv p_9(t) \equiv 2t \pmod 5$. If $p_n(t) \pmod 5$ really was an effective 2-term recursion, then by the lemma, $p_n(t) \pmod 5$ would have to be 4-periodic. This is patently not the case.

Corollary 3.2. *Let $a(n)$ be a recursively defined sequence with pseudo-effective depth r and height d in a finite set X. Then $a(n)$ is eventually periodic with a period p and offset b so that $p + b \leq d\,|X|^r + 1$.*

Proof. There are only $d\,|X|^r$ many tuples $(n \pmod d, a(n), \dots, a(n+r-1))$ that one can write down. So amongst the first $d|X|^r + 1$ there must be two which are the same, satisfying the requirements of the lemma. We note that if $m < n$ are two indexes for which $(n \pmod d, a(n), \dots, a(n+r-1)), (m \pmod d, a(m), \dots, a(m+r-1))$ are equal, the sequence $a(n)$ must be $(p = n-m)$-periodic with offset $b = m$. Hence $p + b = n - m + m = n \leq d|X|^r + 1$. □

3.3 Reducing modulo $d, F_d(t)$ for periodicity and efficiency

From the general strategy, it seems sensible to pre-compute some large number polynomials $p_n(t)$, and then reduce them modulo $d, F_d(t)$ for different values of d, so that the $p_n(t) \pmod{d, F_d(t)}$ do not need to be computed from scratch every time we choose a different candidate d. However, there are three problems with this approach.

The first problem revolves around the choice for $F_d(t)$. For Lemma 3.1 to hold, we must show that the sequence of residues $p_n(t)$ modulo $d, F_d(t)$ is a recursively defined sequence. The naïve approach of taking the recursion defining $p_n(t)$ simply reducing it modulo $d, F_d(t)$ does not always work because, in general, the derivative does not commute with reduction modulo $d, F_d(t)$. (All of recursions of Subsection 2.6 include a derivative). For example, let $F_d(t) = t^a, d \neq a$. Then

$$\overline{\partial_t(t^a)} \equiv \overline{at^{a-1}} \not\equiv \overline{0} \pmod{5, t^a} \quad \text{whereas} \quad \partial_{\overline{t}}\left(\overline{t^a}\right) \equiv \partial_{\overline{t}}\left(\overline{0}\right) \equiv \overline{0} \pmod{5, t^a}.$$

The snippet of Appendix C shows that this is not just a problem in theory, but also in practice. If we reduce the recursion (3.1) modulo $5, t^5 - t$, then the n-th polynomial produced by the reduced recursion is *not* equal to $p_n(t) \pmod{5, t^5 - t}$. Indeed, the snippet yields

```
p_{23}(t) (mod 5, t^d -t) = t^4 + 3*t^2 + 2 but q_{23} = t^4 + 4*t^2 + 1
```

The $p_n(t)$ in the snippet are the are the $p_n(t)$ computed in the usual way, and the $q_n(t)$ are those produced by the reduced recursion.

We fix this problem, by choosing polynomials $F_d(t)$ so reduction modulo $d, F_d(t)$ *does* commute with the derivative. The prototypical example is $F_d(t) = f(t)^d$, where $f(t)$ is any polynomial. We verify this in the following, first by performing long division and writing $p_n(t) = q(t)F_d(t) + r(t)$. Then $p(t) \equiv r(t) \pmod{d, F_d(t)}$ and

$$p'(t) = q'(t)F_d(t) + dq(t)f'(t)f(t)^{d-1} + r'(t) \quad \text{so} \quad p'(t) \equiv r'(t) \pmod{d, F_d(t)}.$$

This shows that computing the reduction first to obtain $r(t)$ with $\deg(r) < \deg(F_d(t)) = d\deg(f(t))$ and then computing the derivative $r'(t)$ is the same as computing the derivative $p'(t)$ and then reducing.

Moreover, as if $f(0) = 0$, then $p_n(0) \not\equiv 0 \pmod{d, F_d(t)}$ implies $p_n(0) \neq 0$. This is readily seen by long division.

In conclusion, we have shown that if we choose $F_d(t) = f(t)^d$, then $p_n(t) \pmod{d, F_d(t)}$ satisfies the same (integral) recursion as p_n and so have the

Lemma 3.3. *Let $p_n(t)$ be described by one of the integral recursions of Section 2.6, d a prime number and $f(t)$ a polynomial with $f(0) = 0$. Then the sequence $p_n(t) \pmod{d, F_d(t) = f(t)^d}$ is eventually periodic.*

The advantage of using $F(t) = t^d$ is that the reduced polynomial $p(n) \pmod{d, F(t)}$ can be represented using a polynomial of degree less than $\deg(t^d) = d$. This is comparatively cheap to store. Conversely, the advantage of using $F(t) = (t^d - t)^d$, is that, as a function, the reduced polynomial has the same values everywhere, not just at 0. Indeed, in general, if $p(t) \equiv q(t) \pmod{F(t)}$, then $p(a) = q(a)$ if $F(a) = 0$. Since $t^d - t$ is zero on all of $\mathbb{Z}/d\mathbb{Z}$, we see that $\overline{p_n}(a) = p_n(a)$ where $\overline{p_n}$ is the residue of p_n modulo $t^d - t$ (or any power thereof).

The second problem of pre-computing many $p_n(t)$ only to reduce them modulo $d, F_d(t)$ is that we do not know how many polynomials we must pre-compute because we do not know how the period p and offset b grow. The bound $p + b \leq d^{4d+1}$ given in [OR12, Th. 6.1, pg. 14] is not particularly forgiving (we obtain a similarly unforgiving bound in Corollary3.2 with $|X| = \left|\mathbb{Z}[t]/(d, t^d)\right| = d^d$ and $r = 2$). This means, if we expect to find a successful candidate amongst the first n primes, we would need to compute $\approx n \log(n)^{4n\log(n)+1}$ polynomials. This is especially infeasible when we consider the third problem.

Finally, the third problem is that the coefficients of the polynomials defined by a recursion of the sort

$$Q_n(T) = 4\left(T^3 - 1\right) Q'_{n-1}(T) - (k + 2n - 2)T^2 Q_{n-1}(T) - (n-1)(n + k - 2)TQ_{n-2}(T)$$
$$R_n(S) = 6\left(S^2 - 1\right) R'_{n-1}(S) - (k + 2n - 2)SR_{n-1}(S) - (n-1)(n + k - 2)R_{n-2}(S)$$

(from "general integral case" Subsection 2.6) grow exponentially. This is in part because we multiply by 12^n to ensure that they are whole numbers. Although the coefficients grow exponentially, the storage requirements only grow cubically with n (facts we will see shortly).

So *even if we knew* how many polynomials we must pre-compute, we simply would not have enough space to store them. Of course, the computation time also grows with the size of the polynomials.

This is illustrated by the snippet in Appendix B. First we compute the polynomials as-is, with `Recursion(Q**3, 2)` (line 37) and then modulo $5, t^5$ with `Recursion(Q**3, 2, d=5)` (line 64). We plot the cube root of the memory usage against the number of polynomials computed in maroon and the logarithm of the largest coefficient of the polynomials $p_n(t)$ in teal to obtain Figure 3.1

That the dotted maroon line is a straight line, shows that the space required to store n (non-reduced) polynomials does indeed grow cubically. That the teal line is straight shows that the largest coefficient of $p_n(t)$ does indeed grow exponentially.

Moreover, we see that they require a non-trivial amount of memory: storing 8500 unreduced polynomials needed $\approx 47\,\text{GB}$ of RAM on my machine[2].

[2]Since I do not have 47 GB of physical RAM, I had to create large swapfiles for this to work. So I am sure this is

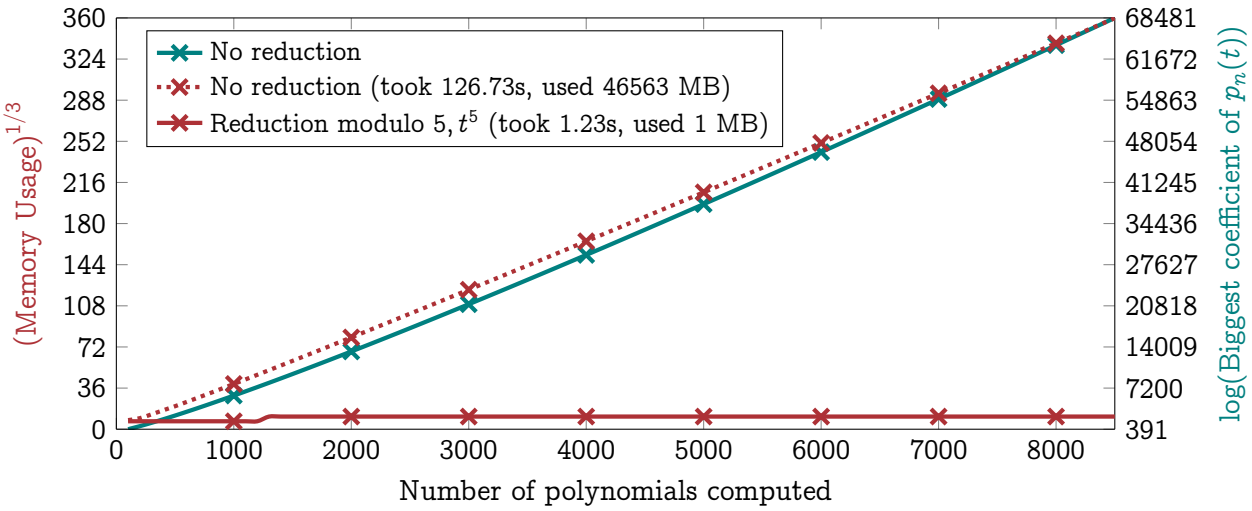


Figure 3.1: Comparison of memory usage when computing the polynomials $p_n(t)$ without reduction and with reduction

We conclude that computing the polynomials $p_n(t) \pmod{d, t^d}$ for each candidate d separately is faster than pre-computing many polynomials $p_n(t)$ and then reducing for each candidate.

3.4 Efficient period detection for the recursively defined sequences

From point (ii) of the general strategy (Subsection 3.1), we need to know how many polynomials we need to compute to determine the period and offset of $p_n(t) \pmod{d, F_d(t)}$ for some prime d. We present a general algorithm here using the condition (3.2) of Lemma 3.1.

not an error of misinterpreting units e.g. kB vs MB.

Algorithm 1 Algorithm to compute period and offset of eventually periodic (zero-indexed) sequence defined by a recursion `R` of pseudo-effective depth `r` and height `d`. The sequence is given as an iterator object `R`, with `next(R)` computing the next item. The minimal period `P` and offset `B` is returned.

```
def period_offset(R, r, d):
  # Compute the first r terms
  S = [next(R) for _ in range(r)]
  P = 0
  while not P:
    # Append the next value in the sequence
    S += [next(R)]
    # Move a window of r elements over the sequence to verify condition (4.2)
    for B in range((len(S)-r) % d, len(S)-r, d):
      if S[B:B+r] == S[len(S)-r:len(S)]:
        # Condition (4.2) is met with M = B, N = len(S) - r
        # Sequence is `N - M` periodic with offset `B`
        P = len(S) - r - B
        break
  # Compute the minimal period `p`
  for p in divisors(P):
    if all([S[B:B+p] == S[B+p*k:B+p*(k+1)] for k in range(1, P//p)]):
      break
  # Compute another `p` terms to ensure `S` contains at least two periods
  S += [next(R) for _ in range(p)]
  # Compute minimal offset
  for b in range(B+1):
    if S[b:b+p] == S[b+p:b+2*p]:
      break
  return p, b

```

3.5 Implementing O'Sullivan-Risager's method

Our implementation will accept functions as polynomials in $Q = E_4$ and $R = E_6$. O'Sullivan-Risager's method, as detailed in Subsection 2.5, generally requires one to divide the function f in S_{12} of interest by Δ. We know the quotient f/Δ is a constant, however sagemath cannot compute this as we are treating Q, R as symbolic variables. For example

```
import sage.all
from sage.rings.integer_ring import ZZ
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing

S = PolynomialRing(ZZ, names=('Q', 'R'))
(Q, R, ) = S._first_ngens(2)

print(S(Q**3/(Q**3-R**2)))
```

raises a `TypeError: fraction must have unit denominator`. Therefore, we can only allow input functions explicitly as a multiple of Δ. For example, this works

```
print(S(4*(Q**3-R**2)/(Q**3-R**2)))
```

3.6 Installation and usage of `TaylorExpansion`

The source is available at github.com/rrueger/TaylorExpansion.

`TaylorExpansion` has been packaged as a regular Python module. It can be installed for usage with Python with

```
pip install https://github.com/rrueger/TaylorExpansion/raw/main/dist/
  sage_taylorexpansion-0.9.0-py3-none-any.whl
```

Sage uses its own package hierarchy to match the version of Python it is shipped with. To install for usage with Sage call

```
sage -pip install https://github.com/rrueger/TaylorExpansion/raw/main/dist/
  sage_taylorexpansion-0.9.0-py3-none-any.whl
```

Sage is not officially available as a python module on PyPi, therefore it is not listed as a dependency in this module and will *not* be automatically installed alongside `TaylorExpansion`. Sage must be installed on the system independently, before `TaylorExpansion` can be installed with `sage -pip` as above.

The usage of the command line program `taylor-product` which is installed when using `pip install ...` is described in the `README` on GitHub.

3.7 Verification of the implementation

We can verify the `TaylorExpansion` methods in the following

```
taylor-expansion --OR 'Q^3 - R^2' 2
```

```
weight = 12, fn = Q^3 - R^2, order = 2, candidate =    5, offset =    0,
   period =   20 (took 42 elements), All non-trivial Fourier coefficients are
    non-zero modulo 5
Here is a repeating period of 20 polynomials
This sequence repeats forever

  p_{0}(t) = 4*t^2 + 1          (mod 5)
  p_{1}(t) = 0                  (mod 5)
  p_{2}(t) = 2*t^2 + 3          (mod 5)
  p_{3}(t) = 2*t^3 + 3*t        (mod 5)
  p_{4}(t) = 4*t^2 + 1          (mod 5)
  p_{5}(t) = 3*t^3 + 2*t        (mod 5)
  p_{6}(t) = 3*t^4 + 4*t^2 + 3  (mod 5)
  p_{7}(t) = 4*t^3 + t          (mod 5)
  p_{8}(t) = 4*t^2 + 1          (mod 5)
  p_{9}(t) = 3*t^3 + 2*t        (mod 5)
  p_{10}(t) = 4*t^4 + 3*t^2 + 3 (mod 5)
  p_{11}(t) = 4*t^3 + 3*t       (mod 5)
  p_{12}(t) = 3*t^4 + 3*t^2 + 1 (mod 5)
  p_{13}(t) = 2*t^3             (mod 5)
  p_{14}(t) = 4*t^4 + 3*t^2 + 3 (mod 5)
  p_{15}(t) = 4*t               (mod 5)
```

```
  p_{16}(t) = t^2 + 1            (mod 5)
  p_{17}(t) = 3*t^3 + t          (mod 5)
  p_{18}(t) = t^4 + 3            (mod 5)
  p_{19}(t) = 4*t                (mod 5)
  p_{20}(t) = 4*t^2 + 1          (mod 5) = p_{0}(t) = 4*t^2 + 1
  p_{21}(t) = 0                  (mod 5) = p_{1}(t) = 0
  ...
Now finding periodic behaviour of p_n(t=0)
The values p_{n}(0) are 4 periodic
That is, the non-trivial values p_{n}(0) are 2 periodic:
  p_{4*n + 0}(0) = 1 (mod 5)
  p_{4*n + 2}(0) = 3 (mod 5)
```

This is the same result as obtained by O'Sullivan-Risager.

3.8 Non-vanishing of the Fourier Coefficients of the j-function

Our main motivation was to compute the Fourier coefficients for the j-function, given by $j = 1728Q^3/\Delta$. Since Δ lies in S_{12}, we know that $\theta(j) = 1728\theta(Q^3)/\Delta$. We perform a computation very similar to (2.11). We know that $\theta^{[0]}(j) = j = 1728Q^3/\Delta$ and $\theta^{[1]}(j) = 1278\theta(Q^3)/\Delta$. Now we inductively verify

$$\begin{aligned}
\theta^{[n]}(j) &= \theta(\theta^{[n-1]}(j)) - n(n+k-1)\frac{Q}{144}\theta^{[n-1]}(j) \\
&= \theta\left(\frac{1728}{\Delta}\theta^{[n-1]}(Q^3)\right) - n(n+k-1)\frac{Q}{144}\frac{1728}{\Delta}\theta^{[n-1]}(Q^3) \\
&= \frac{1728}{\Delta}\theta(\theta^{[n-1]}(Q^3)) - n(n+k-1)\frac{Q}{144}\frac{1728}{\Delta}\theta^{[n-1]}(Q^3) \\
&= \frac{1728}{\Delta}\left(\theta(\theta^{[n-1]}(Q^3)) - n(n+k-1)\frac{Q}{144}\theta^{[n-1]}(Q^3)\right) \\
&= \frac{1728}{\Delta}\theta^{[n]}(Q^3)
\end{aligned}$$

As such, we are really only interested in computing $\theta^{[n]}(Q^3)$. Although Q^3 is in M_{12} it is not a cusp form, so we cannot use the O'Sullivan-Risager method. We prepare use of the module with the imports

```
from TaylorExpansion import compute

import sage.all
from sage.sets.primes import Primes
from sage.rings.integer_ring import ZZ
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing

S = PolynomialRing(ZZ, names=('Q', 'R'))
(Q, R, ) = S._first_ngens(2)
P = Primes().unrank_range(2, 1000)
```

We begin with the expansion around ρ, a point of order 3.

```
TaylorExpansion.compute(S('Q^3'), 3, candidates=P, verbose=True, OS=False)
```

Our first prime candidate is 5. Our program tells us that the polynomials $p_n(t)$ are periodic with period 1 and offset 4, they are $p_5(t) = p_{\{n\geq 5\}}(t) = 0 \pmod 5$. Therefore, by the general strategy (Subsection 3.1), we cannot use this information to conclude any non-vanishing properties of the non-trivial Fourier coefficients.

We continue with the next (prime) candidate 7. The polynomials are again periodic with offset 3 and period 42. Importantly, all non-trivial Fourier coefficients are non-zero! The first 3 polynomials are $p_0 \equiv t^3$, $p_1 \equiv 2t^2$, $p_2 \equiv 4t^4 + 5t$. Then the following 42 are

$p_3 \equiv 6t^3 + 1$	$p_4 \equiv 6t^5 + t^2$	$p_5 \equiv 5t^4 + 2t$	$p_6 \equiv t^6 + 6$	$p_7 \equiv 5t^5 + 2t^2$	$p_8 \equiv 4t^4 + 5t$
$p_9 \equiv 4t^6 + 2t^3 + 1$	$2t^5 + 5t^2$	$t^4 + 2t$	$2t^6 + 2t^3 + 6$	$6t^5 + 6t^2$	$5t$
$p_{15} \equiv 2t^3 + 1$	$3t^5$	$2t$	6	$5t^2$	$5t$
$p_{21} \equiv 6t^3 + 1$	0	$5t^4 + 2t$	$t^3 + 6$	$t^5 + 6t^2$	$2t^4 + 5t$
$p_{27} \equiv 6t^6 + 1$	$2t^5 + 5t^2$	$3t^4 + 2t$	$3t^6 + 5t^3 + 6$	$5t^5 + 2t^2$	$6t^4 + 5t$
$p_{33} \equiv 5t^6 + 5t^3 + 1$	$t^5 + t^2$	$2t$	$5t^3 + 6$	$4t^5$	$5t$
$p_{39} \equiv 1$	$2t^2$	$2t$	$t^3 + 6$	0	$2t^4 + 5t$

Computing the next 2 polynomials $p_{45} \equiv 6t^3 + 1$, $p_{46} = 6t^4 + t^2$ verifies that the sequence indeed repeats (since the indices of comparison $45, 3$ are the same modulo 7, as in Lemma 3.1). A visual inspection immediately shows us that

$$p_{6n}(0) \equiv 6 \pmod 7 \quad \text{and} \quad p_{6n+3}(0) \equiv 1 \pmod 7.$$

We also see that $p_{3n+1}(0) \equiv p_{3n+2}(0) \equiv 0 \pmod 7$. This is consistent with Lemma 2.1. Indeed, $k/2 = 6$ and so when $n \not\equiv k/2 \equiv 0 \pmod 3$ we expect that $p_n(t) = 0$ (this is true without reduction modulo 7).

We can also plot the coefficients of each monomial t^k in $q_n(t) \pmod 7$ to see the periodic and non-vanishing behaviour of $p_n(t)$ and in particular $p_n(0)$. The marked points on the black line indicate the values of $p_n(0) \pmod 7$ at non-trivial Fourier coefficients. We see they alternate between 0 and 6 modulo 7.

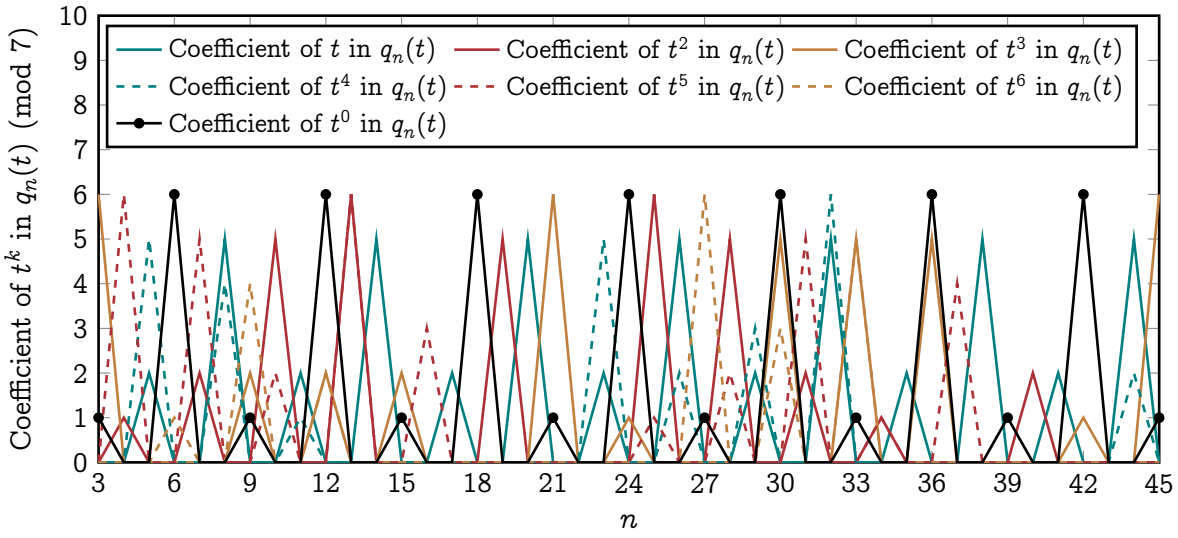


Figure 3.2: Graphing the coefficients of each monomial t^k in $q_n(t)$.

`TaylorExpansion` automates the majority of this work. When called with

```
taylor-expansion --latex 'Q^3' 3
```

our program produces the following (LaTeX source code to produce the) output

> Modulo 7, the polynomials have period 42 and offset 3
> The first 3 polynomials are
>
> $p_0 \equiv t^3$ $p_1 \equiv 2t^2$ $p_2 \equiv 4t^4 + 5t$
>
> The 42 repeating polynomials are
>
> $p_3 \equiv 6t^3 + 1$ $p_4 \equiv 6t^5 + t^2$ $p_5 \equiv 5t^4 + 2t$ $p_6 \equiv t^6 + 6$ $p_7 \equiv 5t^5 + 2t^2$ $p_8 \equiv 4t^4 + 5t$ $p_9 \equiv 4t^6 + 2t^3 + 1$ $p_{10} \equiv 2t^5 + 5t^2$ $p_{11} \equiv t^4 + 2t$ $p_{12} \equiv 2t^6 + 2t^3 + 6$ $p_{13} \equiv 6t^5 + 6t^2$
> $p_{14} \equiv 5t$ $p_{15} \equiv 2t^3 + 1$ $p_{16} \equiv 3t^5$ $p_{17} \equiv 2t$ $p_{18} \equiv 6$ $p_{19} \equiv 5t^2$ $p_{20} \equiv 5t$ $p_{21} \equiv 6t^3 + 1$ $p_{22} \equiv 0$ $p_{23} \equiv 5t^4 + 2t$ $p_{24} \equiv t^3 + 6$
> $p_{25} \equiv t^5 + 6t^2$ $p_{26} \equiv 2t^4 + 5t$ $p_{27} \equiv 6t^6 + 1$ $p_{28} \equiv 2t^5 + 5t^2$ $p_{29} \equiv 3t^4 + 2t$ $p_{30} \equiv 3t^6 + 5t^3 + 6$ $p_{31} \equiv 5t^5 + 2t^2$ $p_{32} \equiv 6t^4 + 5t$ $p_{33} \equiv 5t^6 + 5t^3 + 1$ $p_{34} \equiv t^5 + t^2$ $p_{35} \equiv 2t$
> $p_{36} \equiv 5t^3 + 6$ $p_{37} \equiv 4t^5$ $p_{38} \equiv 5t$ $p_{39} \equiv 1$ $p_{40} \equiv 2t^2$ $p_{41} \equiv 2t$ $p_{42} \equiv t^3 + 6$ $p_{43} \equiv 0$ $p_{44} \equiv 2t^4 + 5t$
>
> Values of $p_n(0) \pmod 7$
>
> $p_{6n+3}(0) \equiv 1 \pmod 7$ $p_{6n+0}(0) \equiv 6 \pmod 7$

We can do the same for expanding around ι. Here, the polynomials repeat with a period of 936 and offset 143 for the candidate $d = 13$. The full list is given in Appendix D. Here we show the periodic, and in particular non-zero, behaviour of the $p_n(0)$.

Values of $p_n(0) \pmod{13}$

$p_{72n+2}(0) \equiv 8 \pmod{13}$ $p_{72n+4}(0) \equiv 12 \pmod{13}$ $p_{72n+6}(0) \equiv 4 \pmod{13}$ $p_{72n+8}(0) \equiv 1 \pmod{13}$ $p_{72n+10}(0) \equiv 9 \pmod{13}$ $p_{72n+12}(0) \equiv 12 \pmod{13}$ $p_{72n+14}(0) \equiv 2 \pmod{13}$ $p_{72n+16}(0) \equiv 3 \pmod{13}$ $p_{72n+18}(0) \equiv 1 \pmod{13}$ $p_{72n+20}(0) \equiv 10 \pmod{13}$
$p_{72n+22}(0) \equiv 12 \pmod{13}$ $p_{72n+24}(0) \equiv 3 \pmod{13}$ $p_{72n+26}(0) \equiv 7 \pmod{13}$ $p_{72n+28}(0) \equiv 4 \pmod{13}$ $p_{72n+30}(0) \equiv 10 \pmod{13}$ $p_{72n+32}(0) \equiv 9 \pmod{13}$ $p_{72n+34}(0) \equiv 3 \pmod{13}$ $p_{72n+36}(0) \equiv 4 \pmod{13}$ $p_{72n+38}(0) \equiv 5 \pmod{13}$ $p_{72n+40}(0) \equiv 1 \pmod{13}$
$p_{72n+42}(0) \equiv 9 \pmod{13}$ $p_{72n+44}(0) \equiv 12 \pmod{13}$ $p_{72n+46}(0) \equiv 4 \pmod{13}$ $p_{72n+48}(0) \equiv 1 \pmod{13}$ $p_{72n+50}(0) \equiv 11 \pmod{13}$ $p_{72n+52}(0) \equiv 10 \pmod{13}$ $p_{72n+54}(0) \equiv 12 \pmod{13}$ $p_{72n+56}(0) \equiv 3 \pmod{13}$ $p_{72n+58}(0) \equiv 1 \pmod{13}$ $p_{72n+60}(0) \equiv 10 \pmod{13}$

$p_{72n+62}(0) \equiv 6 \pmod{13}$ $p_{72n+66}(0) \equiv 3 \pmod{13}$ $p_{72n+70}(0) \equiv 10 \pmod{13}$
$p_{72n+64}(0) \equiv 9 \pmod{13}$ $p_{72n+68}(0) \equiv 4 \pmod{13}$ $p_{72n+0}(0) \equiv 9 \pmod{13}$

We conclude with the result

Lemma 3.4. *The non-trivial Fourier coefficients of the j-function when expanded at ι are non-vanishing.*

3.9 Further results and growth of the period

We can also use `TaylorExpansion` to investigate the growth of the period and offset.

```
from TaylorExpansion import compute

import sage.all
from sage.sets.primes import Primes
from sage.rings.integer_ring import ZZ
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing

S = PolynomialRing(ZZ, names=('Q', 'R'))
(Q, R, ) = S._first_ngens(2)
P = Primes().unrank_range(2, 24)

for p in P:
    for line in compute(Q**3, 2, candidates=[p], verbose=False):
        print(line)
```

Produces the output (post-formatted).

d	Offset	Period	Elts. required	All non-trivial non-zero?
5	4	1	16	No
7	21	1	37	No
11	121	1	145	No
13	2	936	1876	Yes
17	0	1088	2178	No
19	209	1	249	No
23	265	1	313	No
29	18	5684	11388	Yes
31	609	1	673	No
37	26	23976	47980	Yes
41	30	16400	32832	No
43	1365	1	1453	No
47	1681	1	1777	No
53	42	35828	71700	Yes
59	2821	1	2941	No
61	50	109800	219652	No
67	3741	1	3877	No
71	4249	1	4393	No
73	62	94608	189280	Yes
79	5361	1	5521	No
83	5965	1	6133	No
89	78	86152	172384	No

Here, the "Elts. required" column denotes how many elements needed to be computed before the period could be successfully detected with property (3.2). We see two interesting things.

Firstly, in cases where all non-trivial Fourier coefficients are non-zero the period grows very quickly. The log-plot (Figure 3.3) of both the period (in cases where all non-trivial Fourier coefficients are non-zero) appears somewhat straight. It also indicates that the number of elements required to detect periodicity is a constant multiple of the period, showing that the algorithm scales sensibly.

The second interesting point is that often, when there non-trivial Fourier coefficients which *are* $0 \pmod{d}$, it is because the polynomials eventually become 0 (period 1). However, it is not *always* the case.

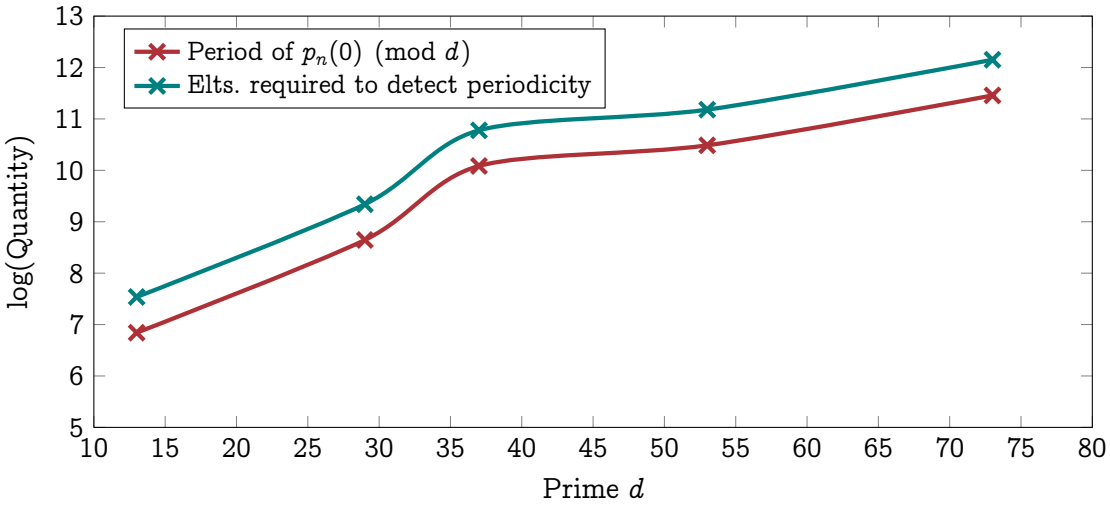


Figure 3.3: A log-plot (in maroon) of the size of the period of $p_n(0) \pmod d$ where d is a prime where all non-trivial Fourier coefficients are non-zero modulo d. Also plotted (in teal) is the number of element $p_n(t)$ that were required to detect the periodicity.

A

A Sage implementation `MFTaylorExpansion`

For completeness the source of the implementation is listed below. It implements a Python module which can be natively imported in both Python and Sage.

```
#!/use/bin/env python3

from argparse import ArgumentParser, ArgumentTypeError

# Sage modules
import sage.all
from sage.rings.integer_ring import ZZ
from sage.rings.rational_field import QQ
from sage.rings.finite_rings.integer_mod_ring import Zmod
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing
from sage.rings.laurent_series_ring import LaurentSeriesRing
from sage.calculus.functional import derivative
from sage.sets.primes import Primes
from sage.arith.misc import divisors
from sage.misc.latex import latex

# Silence warnings about slow implementations in the pre-processing step e.g.
#   verbose 0 (4176: multi_polynomial_ideal.py, groebner_basis)
#   Warning: falling back to very slow toy implementation.
try:
  from sage.misc.verbose import set_verbose
  set_verbose(-1)
except ImportError:
  try:
    from sage.misc.misc import set_verbose
    set_verbose(-1)
  except ImportError:
    pass

(t, ) = PolynomialRing(ZZ, names=('t',))._first_ngens(1)
S = PolynomialRing(ZZ, names=('Q', 'R'))
(Q, R, ) = S._first_ngens(2)


def reduce(poly, n, lift=True):
  # Assumes poly is an element of PolynomialRing(ZZ, names=('t'))
  R = PolynomialRing(ZZ, names=('t'))
  T = R.change_ring(Zmod(n))
  (t, ) = T._first_ngens(1)
  S = T.quotient(t**n)
  if lift:
    return S(poly).lift()
  else:
    return S(poly)

  return poly % n

```

```
48
49 def compute_weight(p, check=True):
50   if p == 0:
51     return 0
52
53   if check:
54     weights = [4*Q_power + 6*R_power for (Q_power, R_power) in p.exponents()]
55     if all([weight == weights[0] for weight in weights]):
56       return weights[0]
57     print(f'{p} is not homogeneous!')
58     exit(1)
59   else:
60     Q_power, R_power = p.exponents()[0]
61     return 4*Q_power + 6*R_power
62
63
64 def init_poly(p, order, OR=False):
65   (t, ) = PolynomialRing(ZZ, names=('t'))._first_ngens(1)
66
67   k = compute_weight(p)
68
69   if OR:
70     if k != 12:
71       print("Error. O'Sullivan-Risager variant only for forms of weight 12")
72       print(f"{f} is of weight {k}")
73       exit(1)
74     return 1
75
76   if p == 0:
77     return 0*t
78
79   # # Sage implementation detail
80   # # In theory, an approach like this would be ideal
81   #
82   # R1 = LaurentSeriesRing(QQ, names=('q')); (q, ) = R1._first_ngens(1)
83   # R2 = LaurentSeriesRing(R1, names=('r')); (r, ) = R2._first_ngens(1)
84   # R3 = PolynomialRing(R3, names=('w')); (w, ) = R3._first_ngens(1)
85   #
86   # if order == 3:
87   #   S = R3.quotient(q**12 * r**(-8) - w)
88   #   # ...
89   # elif order == 2:
90   #   S = R3.quotient(r**12 * q**(-18) - w)
91   #   # ...
92   #
93   # # However, we have the following issue
94   # print(S(q*r).lift())
95   # q*r  # Here we want w!
96   #
97   # # So we are forced to use an approach like this (e.g. for order = 2)
98   # R = LaurentSeriesRing(QQ, names=('q')); (q, ) = R._first_ngens(1)
99   # RR = PolynomialRing(R, names=('r', 'w')); (r, w, ) = RR._first_ngens(2)
100  # S = RR.quotient(q*r-w)
```

```
  # print(S(q*r).lift())
  # w

  if order == 3:
    # Point = \rho
    # R(\rho) != 0, so we can invert R:
    #   Q^a R^b = R^{k/6} (Q R^{-2/3})^a
    # Sage does not support fractional powers for LaurentSeriesRing
    # Think: r = R^{1/12}
    R = LaurentSeriesRing(QQ, names=('r'))
    (r, ) = R._first_ngens(1)
    # Think: q = Q^{1/12}
    # w is a helper ariable used for the quotient
    RR = PolynomialRing(R, names=('q', 'w'))
    (q, w, ) = RR._first_ngens(2)
    # We want w = QR^{-2/3} = q^{12}r^{-8}
    S = RR.quotient(q**12 * r**(-8) - w)

    s = 0*q + 0*r
    for coef, (Q_power, R_power) in zip(p.coefficients(), p.exponents()):
      s += coef * q**(12*Q_power) * r**(12*R_power)
    s *= r**(-2*k)
    s = S(s).lift()

  elif order == 2:
    # Point = i
    # Q(i) != 0, so we can invert Q:
    #   Q^a R^b = Q^{k/4} (Q^{-3/2} R)^b
    # Sage does not support fractional powers for LaurentSeriesRing
    # Think: q = Q^{1/12}
    R = LaurentSeriesRing(QQ, names=('q'))
    (q, ) = R._first_ngens(1)
    # Think: r = R^{1/12}
    # w is a helper variable used for the quotient
    RR = PolynomialRing(R, names=('r', 'w'))
    (r, w, ) = RR._first_ngens(2)
    S = RR.quotient(r**12*q**(-18) - w)

    s = 0*q + 0*r
    for coef, (Q_power, R_power) in zip(p.coefficients(), p.exponents()):
      s += coef * q**(12*Q_power) * r**(12*R_power)
    s *= q**(-3*k)
    s = S(s).lift()

  p_init = 0*t
  # s is a polynomial in RR (it was sent to S and the lifted back to RR)
  # Therefore, although it is expressed only in the variable w, it is
  # formally in r and w (See the definition of RR)
  # So we throw away the first in every pair using (_, exp)
  for coef, (_, exp) in zip(s.coefficients(), s.exponents()):
    p_init += int(coef.subs(r=0, w=0, q=0))*t**exp

  return p_init
```

```
154
155
156 class Recursion:
157   def vprint(self, *args, **kwargs):
158     if self.verbose:
159       print(*args, **kwargs)
160
161   def _reduce(self, poly): return reduce(poly, self.d) if self.d else poly
162
163   def __init__(self, f, order, d=None, verbose=True, OR=False):
164     self.n = 0
165     self.order = order
166     self.d = d
167     self.verbose = verbose
168     self.OR = OR
169     self.k = compute_weight(f)
170
171     if OR and self.k != 12:
172       print("Error. O'Sullivan-Risager variant only for forms of weight 12")
173       print(f"{f} is of weight {self.k}")
174       exit(1)
175
176     if self.OR:
177       try:
178         self.f = S(f/(Q**3 - R**2))
179       except TypeError:
180         print(f"Error. When using the OR-method, {f} must be given"
181             f"as explicit multiple of the discriminant Q^3-R^2")
182         exit(1)
183     else:
184       self.f = f
185
186     self.p0 = self._reduce(init_poly(self.f, order))
187     theta_p0 = -(4*R*derivative(self.f, Q) + 6*Q**2*derivative(self.f, R))
188     self.p1 = self._reduce(init_poly(theta_p0, order))
189
190   def __iter__(self): return self
191
192   def __next__(self):
193     if self.n == 0:
194       self.n += 1
195       return self._reduce(self.p0)
196     elif self.n == 1:
197       self.n += 1
198       return self._reduce(self.p1)
199
200     if self.order == 2 and self.OR:
201       p2 = self._reduce(6*(t**2-1)*derivative(self.p1)
202                 - 2*(self.n-1)*t*self.p1
203                 - (self.n-1)*(self.n+10)*self.p0)
204     elif self.order == 2 and not self.OR:
205       p2 = self._reduce(6*(t**2-1)*derivative(self.p1)
206                 - (self.k+2*self.n-2)*t*self.p1
```

```
                - (self.n-1)*(self.n+10)*self.p0)
    elif self.order == 3 and self.OR:
      p2 = self._reduce(4*(t**3-1)*derivative(self.p1)
                - 2*(self.n-1)*t**2*self.p1
                - (self.n-1)*(self.n+10)*t*self.p0)
    elif self.order == 3 and not self.OR:
      p2 = self._reduce(4*(t**3-1)*derivative(self.p1)
                - (self.k+2*self.n-2)*t**2*self.p1
                - (self.n-1)*(self.n+self.k-2)*t*self.p0)

    self.n += 1
    self.p0, self.p1 = self.p1, p2
    return self._reduce(self.p1)


def minimal_period(seq, period):
  # Given a periodic (from the start) sequence and a period, find the
  # minimal period
  for min_period in divisors(period):
    # Use a generator expression for lazy eval
    if all((seq[:min_period] == seq[min_period*k:min_period*(k+1)]
        for k in range(1, period//min_period)
        )):
      break
  return min_period


def period_offset(R, r, d):
  # Compute the first r terms
  S = [next(R) for _ in range(r)]
  P = 0
  while not P:
    # Append the next value in the sequence
    S += [next(R)]
    # Move a window of r elements over the sequence to verify condition (4.2)
    for B in range((len(S)-r) % d, len(S)-r, d):
      if S[B:B+r] == S[len(S)-r:len(S)]:
        # Condition (4.2) is met with M = B, N = len(S) - r
        # Sequence is `N - M` periodic with offset `B`
        P = len(S) - r - B
        break
  # Compute the minimal period `p`
  for p in divisors(P):
    if all([S[B:B+p] == S[B+p*k:B+p*(k+1)] for k in range(1, P//p)]):
      break
  # Compute another `p` terms to ensure `S` contains at least two periods
  S += [next(R) for _ in range(p)]
  # Compute minimal offset
  for b in range(B+1):
    if S[b:b+p] == S[b+p:b+2*p]:
      break
  return p, b, S

```

```
260
261 def pprint(polys, d, period, offset, rd, LaTeX=False):
262   # Pretty print the polynomials
263
264   poly_strs = [f"p_{{{n}}}(t) = {reduce(poly, d)}"
265         for n, poly in enumerate(polys[:offset+period+rd])]
266   max_len = max([len(poly_str) for poly_str in poly_strs])
267
268   if LaTeX:
269     print("\\noindent")
270     print(f"Modulo {d}, the polynomials have period {period} and offset {offset}")
271     print()
272
273     if offset:
274       print("\\noindent")
275       print(f"The first {offset} polynomials are")
276       print(f"% The first {offset} polynomials are " + "{{{{{")
277       print()
278       print('\\begin{multicols}{2}')
279       for n, poly in enumerate(polys[:offset]):
280         print(f"  \\noindent $p_{{{n}}} \\equiv {latex(poly)}$\n")
281       print('\\end{multicols}')
282       print("% }}}}}")
283
284     print()
285     print("\\noindent")
286     print(f"The {period} repeating polynomials are")
287     print(f"% The {period} repeating polynomials are " + "{{{{{")
288
289     print('\\begin{multicols}{2}')
290     for n, poly in enumerate(polys[offset:offset+period]):
291       print(f"  \\noindent $p_{{{n+offset}}} \\equiv {latex(poly)}$\n")
292     print('\\end{multicols}')
293     print("% }}}}}")
294     print()
295
296   else:
297     if offset:
298       print()
299       print(f"The first {offset} non-periodic polynomials")
300       print()
301       for poly in poly_strs[:offset]:
302         print(f"  {poly.ljust(max_len)} (mod {d})")
303       print()
304
305     print(f"Here is a repeating period of {period} polynomials")
306     print("This sequence repeats forever")
307     print()
308
309     for poly in poly_strs[offset:offset+period]:
310       print(f"  {poly.ljust(max_len)} (mod {d})")
311
312     for n, poly in enumerate(poly_strs[offset+period:offset+period+rd]):
```

```
      print(f"  {poly.ljust(max_len)} (mod {d}) = {poly_strs[n+offset]}")
    print("  ...")


def pprint_p0(polys, d, order, period, offset, LaTeX=False):
  # Pretty print the polynomials evaluated at 0

  if not LaTeX:
    print("Now finding periodic behaviour of p_n(t=0)")

  # Since the polys are `period` periodic (after offset `offset`), the same
  # must be true for evaluating at 0, however p(0) may have a shorter period
  values = [int(poly.subs(t=0)) for poly in polys[offset:offset+period]]
  s_period = minimal_period(values, period)

  if LaTeX:
    print("\\noindent")
    print(f"Values of $p_{{n}}(0) \\pmod{{{d}}}$")
    print(f"% Values of $p_{{n}}(0) \\pmod{{{d}}}$" + "{{{{{")
    print('\\begin{multicols}{3}')
    for n, poly in enumerate(polys[offset:offset+s_period]):
      if poly.subs(t=0) != 0:
        print(f"  \\noindent $p_{{{s_period}n + {(n+offset)%s_period}}}(0)"
            f" \\equiv {poly.subs(t=0)} \\pmod{{{d}}}$")
        print()
    print('\\end{multicols}')
    print("% }}}}}")
    print()

  else:
    print(f"The values p_{{n}}(0) are {s_period} periodic")
    print(f"That is, the non-trivial values p_{{n}}(0) are {s_period//order}
     periodic:")

    for n, poly in enumerate(polys[offset:offset+s_period]):
      if poly.subs(t=0) != 0:
        print(f"  p_{{{s_period}*n + {(n+offset)%s_period}}}(0)"
            f" = {poly.subs(t=0)} (mod {d})")


def compute(f, order, candidates=[5], max_candidates=1, verbose=True, OR=False,
    LaTeX=False):
  def vprint(*args, **kwargs):
    if verbose and not LaTeX:
      print(*args, **kwargs)

  k = compute_weight(f)

  return_str = []
  successful_candidates = 0

  for d in candidates:
    # Initialise recursion
```

```
    R = Recursion(f, order, d=d, verbose=verbose, OR=OR)
    # Get periodicity, offset, and the computed polynomials
    # `comp` includes (at least) two full periods
    period, offset, comp = period_offset(R, 2, d)
    if comp[offset:offset+period] != comp[offset+period:offset+2*period]:
      print("Failed period computation")

    msg = ""
    msg += f"weight = {k}"
    msg += f", fn = {f}"
    msg += f", order = {order}"
    msg += f", candidate = {d:>4}"
    msg += f", offset = {offset:>4}"
    msg += f", period = {period:>4}"
    msg += f" (took {len(comp)} elements)"

    zero = False
    for n, poly in enumerate(comp[offset + abs(k//2 - offset) % order::order]):
      if poly.subs(t=0) == 0:
        nth = n*order + offset + abs(k//2 - offset) % order
        msg += ", Not all non-trivial Fourier coefficients are nonzero"
        msg += f", e.g. the {nth}th poly p_{{{nth}}}(t) = {poly} (mod {d})"
        zero = True
        break

    if not zero:
      # vprint(f"All non-trivial Fourier coefficients are non-zero modulo {d}")
      successful_candidates += 1
      msg += f", All non-trivial Fourier coefficients are non-zero modulo {d}"

    vprint(msg)
    return_str += [msg]

    if verbose and not zero:
      pprint(comp, d, period, offset, 2, LaTeX=LaTeX)
      pprint_p0(comp, d, order, period, offset, LaTeX=LaTeX)

    if successful_candidates == max_candidates:
      break

  return return_str


def qr(polynomial):
  """Verifies that the polynomial given is one of Q, R"""
  S = PolynomialRing(ZZ, names=('Q', 'R'))
  (Q, R, ) = S._first_ngens(2)
  try:
    S(polynomial)
  except TypeError:
    raise ArgumentTypeError(f"'{polynomial}' is not a polynomial in Q, R")
  return S(polynomial)
```

```
417
418 def main():
419   parser = ArgumentParser(description='Prints polynomials for computing the Taylor
      Expansion'
420               ' of a modular form at a given point.'
421               ' The periodicity of the polynomials is computed'
422              )
423   parser.add_argument('--OR', action='store_true',
424             help="Use O'Sullivan-Risager's method."
425                " Only works when function is given as explicit"
426                " multiple of Q^3 - R^2."
427             )
428   parser.add_argument('--latex', action='store_true',
429             help="Pretty print output in LaTeX compatible format"
430             )
431   parser.add_argument('f', type=qr,
432             help="Function to expand."
433                " Expressed as a polynomial in R, Q"
434             )
435   parser.add_argument('order', type=int, choices=[2, 3],
436             help="Order of the point to expand at."
437                " 2: Expands at i. 3: Expands at \\rho"
438             )
439   args = parser.parse_args()
440
441   # Exclude 2 and 3 for now (2, 3 are the only factors of 12)
442   P = Primes().unrank_range(2, 1000)
443   (t, ) = PolynomialRing(ZZ, names=('t',))._first_ngens(1)
444   S = PolynomialRing(ZZ, names=('Q', 'R'))
445   (Q, R, ) = S._first_ngens(2)
446
447   compute(args.f,
448       args.order,
449       candidates=P,
450       OR=args.OR,
451       verbose=True,
452       LaTeX=args.latex)
453
454
455 if __name__ == "__main__":
456   main()
```

B

Memory usage comparison

This snippet produces Figure 3.1.

```
#!/usr/bin/env python3

from TaylorExpansion import Recursion
from os import getpid
from subprocess import run
from math import floor, ceil, sqrt, log
from datetime import datetime

# Sage modules
import sage.all
from sage.rings.integer_ring import ZZ
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing

def _run(command, capture_output=True):
  result = run(command, capture_output=capture_output, text=True)

  try:
    result.check_returncode()
  except Exception as exception:
    if result.stderr:
      print(result.stderr)
    else:
      print(exception)
    exit(1)

  if result.stdout:
    return result.stdout.strip('\n').split('\n')

  return []


def memuse(pid):
  return int(_run(["ps", "-p", f"{pid}", "-o", "vsz="])[0])


upto = 8500
step = 100

pid = getpid()
(Q, R, ) = PolynomialRing(ZZ, names=('Q', 'R'))._first_ngens(2)

start = datetime.now()
memstart = memuse(pid)

R1 = Recursion(Q**3, 2)
S1 = []
M1 = []
C1 = []
```

```
for _ in range(upto//step):
  S1 += [next(R1) for __ in range(step)]
  biggest_coef = max([max([abs(c) for c in poly.coefficients()])
            for poly in S1[-step:]])
  # Need to cast int, since sage treats this like a special type
  C1 += [(len(S1), log(int(biggest_coef)))]
  # Current memory usage of python process in kB
  M1 += [(len(S1), round((memuse(pid)-memstart)**(1/3), 2))]

memfinal1 = (memuse(pid)-memstart)//1000
M1_str = " ".join([f"({n}, {mem})" for n, mem in M1])
C1_str = " ".join([f"({n}, {biggest_coef})" for n, biggest_coef in C1])
diff = (datetime.now() - start)
T1 = round(diff.seconds + diff.microseconds/10**6, 2)

start = datetime.now()
memstart = memuse(pid)

R2 = Recursion(Q**3, 2, d=5)
S2 = []
M2 = []
for _ in range(upto//step):
  S2 += [next(R2) for __ in range(step)]
  # Current memory usage of python process in kB
  M2 += [(len(S2), round((memuse(pid)-memstart)**(1/3), 2))]

memfinal2 = (memuse(pid)-memstart)//1000
M2_str = " ".join([f"({n}, {mem})" for n, mem in M2])
diff = (datetime.now() - start)
T2 = round(diff.seconds + diff.microseconds/10**6, 2)

ymin = min(min([floor(mem) for _, mem in M1]), min([floor(mem) for _, mem in M2]))
ymin = 0 if ymin < 10 else ymin
ymax = max(max([ceil(mem) for _, mem in M1]), max([ceil(mem) for _, mem in M2]))

r_ymin = min([floor(mem) for _, mem in C1])
r_ymax = max([ceil(mem) for _, mem in C1])

print(f'''
\\begin{{figure}}[h]
  \\centering
  \\pgfkeys{{/pgf/number format/.cd,1000 sep={{}}}}
  \\pgfplotsset{{scaled y ticks=false}}
  \\begin{{tikzpicture}}
  \\begin{{axis}}[
    ylabel={{\\teal{{log(Biggest coefficient of $p_n(t)$)}}}},
    width={{0.9\\textwidth}},
    height={{0.3\\textheight}},
    xmin={{0}},
    xmax={{{upto}}},
    xtick={{0,1000,...,{upto}}},
    hide x axis,
    ymin={{{r_ymin}}},
```

```
    ymax={{{r_ymax}}},
    ytick={{{r_ymin},{r_ymin + (r_ymax-r_ymin)//10},...,{r_ymax}}},
    axis y line*={{right}},
    ],
    \\addplot[
    color=teal,
    smooth,
    mark=x,
    mark repeat=10,
    mark phase=10,
    mark options={{solid}},
    mark size=4pt,
    ultra thick,
    ] plot coordinates {{
    {C1_str}
    }}; \\label{{plt1}}
  \\end{{axis}}
  \\begin{{axis}}[
    legend pos={{north west}},
    legend cell align={{left}},
    legend style={{font=\\small}},
    xlabel={{Number of polynomials computed}},
    ylabel={{\\maroon{{${{\\left(\\textup{{Memory Usage}}\\right)}}^{{\\frac
   {{1}}{{3}}}}$}}}},
    width={{0.9\\textwidth}},
    height={{0.3\\textheight}},
    xmin={{0}},
    xmax={{{upto}}},
    xtick={{0,1000,...,{upto}}},
    ymin={{{ymin}}},
    ymax={{{ymax}}},
    ytick={{{ymin},{ymin + (ymax-ymin)//10},...,{ymax}}},
    axis y line*={{left}},
    ],
    \\addlegendimage{{/pgfplots/refstyle=plt1}}\\addlegendentry{{No reduction}}
    \\addplot[
    color=maroon,
    dotted,
    smooth,
    mark=x,
    mark repeat=10,
    mark phase=10,
    mark options={{solid}},
    mark size=4pt,
    ultra thick,
    ] plot coordinates {{
    {M1_str}
    }};
    \\addlegendentry{{No reduction (took {T1}s, used {memfinal1} MB)}}

    \\addplot[
    color=maroon,
    smooth,
```

```
    mark=x,
    mark repeat=10,
    mark phase=10,
    mark options={{solid}},
    mark size=4pt,
    ultra thick,
    ] plot coordinates {{
    {M2_str}
    }};
    \\addlegendentry{{Reduction modulo $5, t^5$ (took {T2}s, used {memfinal2} MB)}}
  \\end{{axis}}
  \\end{{tikzpicture}}
  \\caption{{Comparison of memory usage when computing the polynomials $p_n(t)$
  without reduction and with reduction}}\\label{{fig:memory}}
\\end{{figure}}
''')
```

C

Reduction during computation example

```
#!/usr/bin/env python3

# Sage modules
import sage.all
from sage.rings.integer_ring import ZZ
from sage.rings.finite_rings.integer_mod_ring import Zmod
from sage.rings.polynomial.polynomial_ring_constructor import PolynomialRing
from sage.calculus.functional import derivative


def reduce(poly, n):
  # Assumes poly is an element of PolynomialRing(ZZ, names=('t'))
  R = PolynomialRing(ZZ, names=('t'))
  T = R.change_ring(Zmod(n))
  (t, ) = T._first_ngens(1)
  S = T.quotient(t**n - t)
  return S(poly).lift()


R = PolynomialRing(ZZ, names=('t'))
(t, ) = R._first_ngens(1)

# Cast 0 and 1 as elements of R
p0, p1 = R(0), R(1)
q0, q1 = R(0), R(1)


diff = False
n = 2

while not diff:
  pn = 6*(t**2-1)*derivative(p1) - 2*(n-1)*t*p1 - (n-1)*(n+10)*p0
  p0, p1 = p1, pn

  qn = reduce(6*(t**2-1)*derivative(q1) - 2*(n-1)*t*q1 - (n-1)*(n+10)*q0, 5)
  q0, q1 = q1, qn

  if reduce(pn, 5) != qn:
    print(f"p_{{{n}}}(t) (mod 5, t^d -t) = {reduce(pn, 5)} but q_{{{n}}} = {qn}")
    diff = True
  n += 1
```

D

Polynomials of the Fourier coefficients of the j-function at ι

The following output is produced using

```
taylor-expansion --latex 'Q^3' 2
```

Modulo 13, the polynomials have period 936 and offset 2
The first 2 polynomials are

$p_0 \equiv 1$

$p_1 \equiv t$

The 936 repeating polynomials are

$p_2 \equiv 5t^2 + 8$
$p_3 \equiv 6t^3 + 7t$
$p_4 \equiv t^2 + 12$
$p_5 \equiv 9t^3 + 4t$
$p_6 \equiv 3t^4 + 6t^2 + 4$
$p_7 \equiv 4t^3 + 9t$
$p_8 \equiv 6t^4 + 6t^2 + 1$
$p_9 \equiv 2t^5 + 10t^3 + t$
$p_{10} \equiv t^4 + 3t^2 + 9$
$p_{11} \equiv t^5 + 12t$
$p_{12} \equiv 9t^6 + t^4 + 4t^2 + 12$
$p_{13} \equiv 12t^5 + 10t^3 + 4t$
$p_{14} \equiv 8t^6 + 12t^4 + 4t^2 + 2$
$p_{15} \equiv 7t^7 + 6t$
$p_{16} \equiv 5t^6 + 5t^2 + 3$
$p_{17} \equiv 4t^7 + 2t^5 + 9t^3 + 11t$
$p_{18} \equiv 10t^8 + 7t^6 + 8t^2 + 1$
$p_{19} \equiv 4t^5 + 6t^3 + 3t$
$p_{20} \equiv 7t^8 + 12t^6 + 10t^2 + 10$
$p_{21} \equiv 11t^9 + 5t^7 + t^3 + 9t$
$p_{22} \equiv 11t^8 + 7t^6 + 3t^4 + 6t^2 + 12$
$p_{23} \equiv 12t^9 + 5t^7 + 3t^5 + 4t^3 + 2t$
$p_{24} \equiv 4t^{10} + 4t^8 + 4t^6 + 4t^4 + 7t^2 + 3$
$p_{25} \equiv 6t^9 + 10t^7 + 4t^3 + 6t$
$p_{26} \equiv 5t^{10} + 10t^8 + 10t^6 + 7t^4 + 7$
$p_{27} \equiv 6t^{11} + 8t^9 + 7t^7 + 10t^5 + t^3 + 7t$
$p_{28} \equiv 7t^{10} + 8t^8 + 6t^6 + 10t^4 + 4t^2 + 4$
$p_{29} \equiv 9t^{11} + 5t^9 + 9t^7 + 7t^5 + 4t^3 + 5t$
$p_{30} \equiv 3t^{12} + 7t^{10} + 7t^4 + 12t^2 + 10$
$p_{31} \equiv 5t^{11} + 8t^9 + 6t^7 + 11t^5 + 3t^3 + 6t$
$p_{32} \equiv 6t^{12} + 3t^{10} + 5t^6 + 8t^4 + 8t^2 + 9$
$p_{33} \equiv 11t^{11} + 5t^9 + 7t^7 + 11t^5 + 4t^3 + 12t$
$p_{34} \equiv 9t^{12} + 11t^{10} + 11t^8 + 4t^6 + 8t^4 + 6t^2 + 3$
$p_{35} \equiv 8t^{11} + 9t^9 + 11t^5 + t^3 + 9t$
$p_{36} \equiv 7t^{12} + 9t^{10} + 4t^8 + 8t^6 + 12t^4 + 12t^2 + 4$
$p_{37} \equiv 5t^{11} + 9t^7 + 10t^5 + 3t^3 + 9t$
$p_{38} \equiv t^{10} + t^8 + 12t^6 + 11t^4 + t^2 + 5$
$p_{39} \equiv 9t^{11} + 4t^9 + 3t^7 + 4t^5 + 2t^3 + 2t$
$p_{40} \equiv 5t^{12} + 3t^{10} + 4t^8 + 11t^6 + 9t^4 + 4t^2 + 1$
$p_{41} \equiv 8t^{11} + 12t^9 + t^7 + 10t^5 + 12t^3 + 5t$
$p_{42} \equiv 10t^{12} + 6t^{10} + 2t^8 + 7t^6 + 10t^4 + 7t^2 + 9$
$p_{43} \equiv 2t^{11} + 2t^9 + t^7 + 12t^5 + 7t^3 + 12t$
$p_{44} \equiv 12t^{12} + 5t^{10} + 2t^8 + 9t^6 + t^4 + t^2 + 12$
$p_{45} \equiv 11t^{11} + 8t^9 + 2t^7 + 5t^5 + 4t^3 + 12t$
$p_{46} \equiv 5t^{12} + 5t^{10} + 11t^8 + 3t^6 + 10t^4 + 4$
$p_{47} \equiv 10t^{11} + 4t^7 + 9t^5 + 10t^3 + 9t$
$p_{48} \equiv 10t^{12} + 10t^{10} + 9t^8 + 5t^6 + 8t^4 + 12t^2 + 1$
$p_{49} \equiv 3t^{11} + 4t^9 + 9t^7 + 11t^5 + 2t^3$
$p_{50} \equiv 4t^{12} + 12t^8 + 6t^6 + 3t^4 + 5t^2 + 11$
$p_{51} \equiv 6t^9 + t^7 + 4t^5 + 10t^3 + 8t$
$p_{52} \equiv t^{12} + 4t^{10} + 10t^8 + 7t^6 + 3t^4 + 7t^2 + 10$
$p_{53} \equiv 3t^{11} + 3t^9 + 5t^5 + 6t^3 + 4t$
$p_{54} \equiv t^{12} + 4t^{10} + 4t^8 + 9t^6 + 7t^4 + 10t^2 + 12$
$p_{55} \equiv 5t^9 + t^7 + 5t^5$
$p_{56} \equiv 10t^{12} + 12t^{10} + 2t^8 + 4t^6 + 11t^4 + 9t^2 + 3$
$p_{57} \equiv 7t^{11} + 11t^9 + 12t^7 + 3t^5 + 2t^3 + t$
$p_{58} \equiv 2t^{12} + 9t^{10} + 6t^8 + 6t^6 + 10t^4 + 8t^2 + 1$
$p_{59} \equiv 12t^{11} + 3t^9 + 3t^7 + 11t^5 + 6t^3 + 12t$
$p_{60} \equiv 7t^{12} + 4t^{10} + t^8 + 7t^6 + 9t^2 + 10$
$p_{61} \equiv 10t^{11} + 7t^7 + 10t^3 + 11t$
$p_{62} \equiv 10t^{12} + 11t^{10} + 8t^8 + 6t^6 + 10t^4 + 3t^2 + 6$
$p_{63} \equiv 10t^{11} + t^9 + 3t^7 + 3t^5 + 5t^3 + 10t$
$p_{64} \equiv 6t^{12} + 10t^8 + 9t^6 + 10t^4 + 9t^2 + 9$
$p_{65} \equiv 6t^{11} + 5t^7 + 2t^5 + 10t^3 + 6t$
$p_{66} \equiv 12t^{12} + 7t^{10} + 7t^8 + 8t^6 + 5t^2 + 3$
$p_{67} \equiv 10t^{11} + t^7 + 8t$
$p_{68} \equiv 6t^{12} + 3t^{10} + 10t^6 + 11t^2 + 4$
$p_{69} \equiv 2t^{11} + 2t^9 + 8t^7 + 4t^5 + 12t^3 + 6t$

$p_{70} \equiv 5t^{12} + 3t^{10} + 3t^8 + t^6 + 12t^4 + 2t^2 + 10$
$p_{71} \equiv 10t^{11} + 11t^9 + 10t^7 + 6t^5 + 6t^3 + 4t$
$p_{72} \equiv t^{12} + t^{10} + 8t^8 + 8t^6 + 3t^4 + 6t^2 + 9$
$p_{73} \equiv 2t^{11} + 4t^9 + 9t^7 + 3t^5 + 11t^3 + 9t$
$p_{74} \equiv 2t^{12} + 2t^{10} + 7t^8 + 11t^6 + 7t^4 + 5t^2 + 8$
$p_{75} \equiv 11t^{11} + t^9 + 8t^7 + 10t^5 + 11t^3 + 4t$
$p_{76} \equiv 6t^{12} + 7t^{10} + 12t^8 + 12t^6 + 12$
$p_{77} \equiv 8t^7 + 8t^5 + 3t^3 + 2t$
$p_{78} \equiv 8t^{12} + 5t^{10} + 12t^8 + 9t^6 + 5t^4 + 3t^2 + 4$
$p_{79} \equiv 2t^{11} + 2t^9 + 4t^7 + 9t^5 + 10t^3 + 7t$
$p_{80} \equiv 8t^{12} + 5t^{10} + 3t^8 + 11t^6 + 9t^4 + t^2 + 1$
$p_{81} \equiv 8t^{11} + 4t^9 + 11t^7 + t^5 + t^3 + 11t$
$p_{82} \equiv 9t^{12} + 4t^{10} + 11t^6 + 8t^4 + 3t^2 + 9$
$p_{83} \equiv 7t^{11} + t^9 + 10t^7 + 5t^5 + 10t^3 + 8t$
$p_{84} \equiv 4t^{12} + 4t^{10} + 3t^8 + t^6 + 2t^4 + 11t^2 + 12$
$p_{85} \equiv 3t^9 + 5t^7 + 9t^3 + 12t$
$p_{86} \equiv 3t^{12} + 9t^{10} + 8t^8 + 2t^6 + t^4 + 6t^2 + 2$
$p_{87} \equiv 7t^{11} + 9t^9 + 3t^7 + 2t^5 + 7t^3 + 11t$
$p_{88} \equiv 11t^{12} + 6t^{10} + 8t^8 + 8t^6 + t^4 + 12t^2 + 3$
$p_{89} \equiv 12t^{11} + 10t^9 + 6t^7 + 12t^5 + 8t^3 + 11t$
$p_{90} \equiv 10t^{12} + 10t^{10} + 3t^8 + 11t^4 + 11t^2 + 1$
$p_{91} \equiv 4t^{11} + 4t^9 + 7t^7 + t^5 + 7t^3 + 7t$
$p_{92} \equiv 8t^{12} + 8t^{10} + 7t^8 + 10t^6 + 12t^4 + t^2 + 10$
$p_{93} \equiv 4t^{11} + 9t^9 + 8t^7 + 8t^5 + 3t^3 + 11t$
$p_{94} \equiv 5t^{12} + 8t^8 + 10t^6 + 5t^2 + 12$
$p_{95} \equiv 5t^{11} + 5t^9 + 6t^7 + 6t^5 + 3t$
$p_{96} \equiv 8t^{12} + 9t^{10} + 6t^8 + t^6 + 2t^4 + 9t^2 + 3$
$p_{97} \equiv 3t^{11} + 9t^9 + 8t^5 + 5t^3 + 2t$
$p_{98} \equiv 12t^{12} + 12t^{10} + 7t^8 + 11t^6 + 7t^4 + 9t^2 + 7$
$p_{99} \equiv 11t^{11} + 3t^9 + 8t^7 + 12t^5 + 12t^3 + 4t$
$p_{100} \equiv 11t^{12} + 11t^{10} + 4t^8 + 5t^6 + 3t^4 + 5t^2 + 4$
$p_{101} \equiv 2t^{11} + 3t^9 + 10t^7 + 8t^5 + 5t^3 + 10t$
$p_{102} \equiv 7t^{12} + 3t^{10} + 8t^8 + 9t^6 + 9t^4 + 4t^2 + 10$
$p_{103} \equiv 12t^9 + 9t^7 + 3t^5 + 7t^3$
$p_{104} \equiv 5t^{12} + 12t^{10} + 6t^8 + 6t^6 + 4t^4 + 5t^2 + 9$
$p_{105} \equiv 8t^{11} + 3t^9 + 12t^7 + t^5 + 8t^3 + t$
$p_{106} \equiv 5t^{12} + 7t^{10} + 11t^8 + 12t^6 + 6t^4 + 9t^2 + 3$
$p_{107} \equiv 10t^9 + 11t^7 + 6t^5 + 2t^3$
$p_{108} \equiv 11t^{12} + t^{10} + 3t^8 + 3t^6 + 10t^4 + 2t^2 + 4$
$p_{109} \equiv 2t^{11} + 9t^9 + 11t^7 + t^5 + t^3$
$p_{110} \equiv t^{12} + 11t^{10} + t^8 + 8t^6 + 11t^4 + 4t^2 + 5$
$p_{111} \equiv 3t^{11} + 6t^9 + 5t^7 + 2t^3 + t$
$p_{112} \equiv 7t^{12} + t^{10} + 7t^8 + 4t^6 + 2t^4 + 12t^2 + 1$
$p_{113} \equiv 8t^{11} + 2t^7 + 4t^5 + 2t^3 + t$
$p_{114} \equiv 3t^{12} + 7t^{10} + 12t^8 + 2t^6 + 3t^4 + 3t^2 + 9$
$p_{115} \equiv 3t^{11} + 6t^9 + 7t^3 + 12t$
$p_{116} \equiv 7t^{12} + 9t^{10} + 9t^8 + 10t^6 + 7t^4 + 8t^2 + 12$
$p_{117} \equiv 2t^{11} + 5t^9 + 10t^7 + 11t^5 + 9t^3 + 8t$
$p_{118} \equiv 4t^{12} + 4t^8 + 12t^6 + 10t^4 + 11t^2 + 4$
$p_{119} \equiv 11t^9 + 4t^7 + 4t^5 + 7t^3 + 2t$
$p_{120} \equiv 2t^{10} + 4t^8 + 5t^6 + 11t^4 + 10t^2 + 1$
$p_{121} \equiv 6t^{11} + 6t^9 + 3t^7 + 4t^5 + 6t^3 + 12t$
$p_{122} \equiv 3t^{12} + 9t^8 + 4t^6 + t^4 + 8t^2 + 11$
$p_{123} \equiv 6t^{11} + t^9 + 8t^7 + 6t^5 + t^3 + 2t$
$p_{124} \equiv 11t^{12} + 11t^{10} + 4t^8 + 7t^6 + 11t^4 + t^2 + 10$
$p_{125} \equiv 9t^{11} + 4t^9 + t^7 + 10t^5 + 12t^3 + 9t$
$p_{126} \equiv 9t^{12} + 9t^{10} + 9t^8 + 6t^6 + t^4 + 6t^2 + 12$
$p_{127} \equiv 4t^{11} + 7t^9 + 9t^7 + 6t^5 + 9t^3 + 2t$
$p_{128} \equiv t^{12} + 2t^{10} + 6t^8 + t^6 + 4t^4 + 8t^2 + 3$
$p_{129} \equiv 12t^9 + 6t^7 + 6t^5 + 7t^3 + 7t$
$p_{130} \equiv 10t^{12} + 2t^{10} + 7t^8 + 8t^6 + 7t^4 + 4t^2 + 1$
$p_{131} \equiv 2t^9 + 12t^7 + 4t^5 + t^3 + 5t$
$p_{132} \equiv 10t^{12} + 4t^{10} + t^8 + 10t^6 + 8t^4 + 11t^2 + 10$
$p_{133} \equiv 2t^{11} + 9t^7 + 3t^5 + 11t^3 + 7t$
$p_{134} \equiv t^{12} + 12t^{10} + 5t^8 + 5t^6 + 3t^4 + 10t^2 + 6$
$p_{135} \equiv 2t^{11} + 5t^9 + 2t^7 + 3t^5 + 7t^3 + 3t$
$p_{136} \equiv 8t^{12} + 4t^{10} + 7t^8 + 8t^6 + 6t^4 + t^2 + 9$
$p_{137} \equiv t^{11} + 3t^9 + 11t^7 + 4t^5 + t^3 + 12t$
$p_{138} \equiv 7t^{12} + 8t^{10} + 3t^8 + 2t^6 + 6t^2 + 3$
$p_{139} \equiv 3t^{11} + 8t^9 + 7t^7 + 9t^5 + 12t^3 + 9t$
$p_{140} \equiv 5t^{12} + 10t^{10} + 9t^8 + 9t^6 + 2t^4 + 9t^2 + 4$
$p_{141} \equiv 5t^{11} + 9t^9 + 6t^7 + 6t^5 + 8t^3 + 6t$
$p_{142} \equiv 3t^{12} + 4t^{10} + 10t^8 + 3t^6 + t^4 + 6t^2 + 10$
$p_{143} \equiv 8t^{11} + 9t^9 + 9t^7 + 5t^5 + 3t^3 + 5t$
$p_{144} \equiv 3t^{12} + 6t^{10} + 5t^8 + 11t^6 + 11t^4 + 7t^2 + 9$
$p_{145} \equiv 3t^{11} + t^9 + 11t^7 + 5t^5 + 11t^3 + 3t$
$p_{146} \equiv 7t^{12} + 9t^{10} + 11t^8 + 11t^6 + 2t^4 + 6t^2 + 8$
$p_{147} \equiv 8t^{11} + 8t^9 + t^7 + 4t^5 + 9t$
$p_{148} \equiv 10t^{10} + 9t^8 + t^6 + 7t^4 + 8t^2 + 12$
$p_{149} \equiv 8t^9 + 9t^7 + 9t^5 + 12t^3 + 12t$
$p_{150} \equiv 8t^8 + 3t^6 + 10t^4 + 4$
$p_{151} \equiv 7t^7 + 12t^5 + 3t^3 + 9t$
$p_{152} \equiv 2t^6 + t^4 + 8t^2 + 1$
$p_{153} \equiv 11t^5 + 7t^3 + 9t$
$p_{154} \equiv 12t^4 + t^2 + 9$
$p_{155} \equiv 11t^3$
$p_{156} \equiv 7t^2 + 12$
$p_{157} \equiv 6t$
$p_{158} \equiv 11t^2 + 2$
$p_{159} \equiv 8t^3 + 5t$
$p_{160} \equiv 10t^2 + 3$
$p_{161} \equiv 12t^3 + t$
$p_{162} \equiv 4t^4 + 8t^2 + 1$
$p_{163} \equiv t^3 + 12t$

$p_{164} \equiv 8t^4 + 8t^2 + 10$
$p_{165} \equiv 7t^5 + 9t^3 + 10t$
$p_{166} \equiv 10t^4 + 4t^2 + 12$
$p_{167} \equiv 10t^5 + 3t$
$p_{168} \equiv 12t^6 + 10t^4 + t^2 + 3$
$p_{169} \equiv 3t^5 + 9t^3 + t$
$p_{170} \equiv 2t^6 + 3t^4 + t^2 + 7$
$p_{171} \equiv 5t^7 + 8t$
$p_{172} \equiv 11t^6 + 11t^2 + 4$
$p_{173} \equiv t^7 + 7t^5 + 12t^3 + 6t$
$p_{174} \equiv 9t^8 + 5t^6 + 2t^2 + 10$
$p_{175} \equiv t^5 + 8t^3 + 4t$
$p_{176} \equiv 5t^8 + 3t^6 + 9t^2 + 9$
$p_{177} \equiv 6t^9 + 11t^7 + 10t^3 + 12t$
$p_{178} \equiv 6t^8 + 5t^6 + 4t^4 + 8t^2 + 3$
$p_{179} \equiv 3t^9 + 11t^7 + 4t^5 + t^3 + 7t$
$p_{180} \equiv t^{10} + t^8 + t^6 + t^4 + 5t^2 + 4$
$p_{181} \equiv 8t^9 + 9t^7 + t^3 + 8t$
$p_{182} \equiv 11t^{10} + 9t^8 + 9t^6 + 5t^4 + 5$
$p_{183} \equiv 8t^{11} + 2t^9 + 5t^7 + 9t^5 + 10t^3 + 5t$
$p_{184} \equiv 5t^{10} + 2t^8 + 8t^6 + 9t^4 + t^2 + 1$
$p_{185} \equiv 12t^{11} + 11t^9 + 12t^7 + 5t^5 + t^3 + 11t$
$p_{186} \equiv 4t^{12} + 5t^{10} + 5t^4 + 3t^2 + 9$
$p_{187} \equiv 11t^{11} + 2t^9 + 8t^7 + 6t^5 + 4t^3 + 8t$
$p_{188} \equiv 8t^{12} + 4t^{10} + 11t^6 + 2t^4 + 2t^2 + 12$
$p_{189} \equiv 6t^{11} + 11t^9 + 5t^7 + 6t^5 + t^3 + 3t$
$p_{190} \equiv 12t^{12} + 6t^{10} + 6t^8 + t^6 + 2t^4 + 8t^2 + 4$
$p_{191} \equiv 2t^{11} + 12t^9 + 6t^5 + 10t^3 + 12t$
$p_{192} \equiv 5t^{12} + 12t^{10} + t^8 + 2t^6 + 3t^4 + 3t^2 + 1$
$p_{193} \equiv 11t^{11} + 12t^7 + 9t^5 + 4t^3 + 12t$
$p_{194} \equiv 10t^{10} + 10t^8 + 3t^6 + 6t^4 + 10t^2 + 11$
$p_{195} \equiv 12t^{11} + t^9 + 4t^7 + t^5 + 7t^3 + 7t$
$p_{196} \equiv 11t^{12} + 4t^{10} + t^8 + 6t^6 + 12t^4 + t^2 + 10$
$p_{197} \equiv 2t^{11} + 3t^9 + 10t^7 + 9t^5 + 3t^3 + 11t$
$p_{198} \equiv 9t^{12} + 8t^{10} + 7t^8 + 5t^6 + 9t^4 + 5t^2 + 12$
$p_{199} \equiv 7t^{11} + 7t^9 + 10t^7 + 3t^5 + 5t^3 + 3t$
$p_{200} \equiv 3t^{12} + 11t^{10} + 7t^8 + 12t^6 + 10t^4 + 10t^2 + 3$
$p_{201} \equiv 6t^{11} + 2t^9 + 7t^7 + 11t^5 + t^3 + 3t$
$p_{202} \equiv 11t^{12} + 11t^{10} + 6t^8 + 4t^6 + 9t^4 + 1$
$p_{203} \equiv 9t^{11} + t^7 + 12t^5 + 9t^3 + 12t$
$p_{204} \equiv 9t^{12} + 9t^{10} + 12t^8 + 11t^6 + 2t^4 + 3t^2 + 10$
$p_{205} \equiv 4t^{11} + t^9 + 12t^7 + 6t^5 + 7t^3$
$p_{206} \equiv t^{12} + 3t^8 + 8t^6 + 4t^4 + 11t^2 + 6$
$p_{207} \equiv 8t^9 + 10t^7 + t^5 + 9t^3 + 2t$
$p_{208} \equiv 10t^{12} + t^{10} + 9t^8 + 5t^6 + 4t^4 + 5t^2 + 9$
$p_{209} \equiv 4t^{11} + 4t^9 + 11t^5 + 8t^3 + t$
$p_{210} \equiv 10t^{12} + t^{10} + t^8 + 12t^6 + 5t^4 + 9t^2 + 3$
$p_{211} \equiv 11t^9 + 10t^7 + 11t^5$
$p_{212} \equiv 9t^{12} + 3t^{10} + 7t^8 + t^6 + 6t^4 + 12t^2 + 4$
$p_{213} \equiv 5t^{11} + 6t^9 + 3t^7 + 4t^5 + 7t^3 + 10t$
$p_{214} \equiv 7t^{12} + 12t^{10} + 8t^8 + 8t^6 + 9t^4 + 2t^2 + 10$
$p_{215} \equiv 3t^{11} + 4t^9 + 4t^7 + 6t^5 + 8t^3 + 3t$
$p_{216} \equiv 5t^{12} + t^{10} + 10t^8 + 5t^6 + 12t^2 + 9$
$p_{217} \equiv 9t^{11} + 5t^7 + 9t^3 + 6t$
$p_{218} \equiv 9t^{12} + 6t^{10} + 2t^8 + 8t^6 + 9t^4 + 4t^2 + 8$
$p_{219} \equiv 9t^{11} + 10t^9 + 4t^7 + 4t^5 + 11t^3 + 9t$
$p_{220} \equiv 8t^{12} + 9t^8 + 12t^6 + 9t^4 + 12t^2 + 12$
$p_{221} \equiv 8t^{11} + 11t^7 + 7t^5 + 9t^3 + 8t$
$p_{222} \equiv 3t^{12} + 5t^{10} + 5t^8 + 2t^6 + 11t^2 + 4$
$p_{223} \equiv 9t^{11} + 10t^7 + 2t$
$p_{224} \equiv 8t^{12} + 4t^{10} + 9t^6 + 6t^2 + 1$
$p_{225} \equiv 7t^{11} + 7t^9 + 2t^7 + t^5 + 3t^3 + 8t$
$p_{226} \equiv 11t^{12} + 4t^{10} + 4t^8 + 10t^6 + 3t^4 + 7t^2 + 9$
$p_{227} \equiv 9t^{11} + 6t^9 + 9t^7 + 8t^5 + 8t^3 + t$
$p_{228} \equiv 10t^{12} + 10t^{10} + 2t^8 + 2t^6 + 4t^4 + 8t^2 + 12$
$p_{229} \equiv 7t^{11} + t^9 + 12t^7 + 4t^5 + 6t^3 + 12t$
$p_{230} \equiv 7t^{12} + 7t^{10} + 5t^8 + 6t^6 + 5t^4 + 11t^2 + 2$
$p_{231} \equiv 6t^{11} + 10t^9 + 2t^7 + 9t^5 + 6t^3 + t$
$p_{232} \equiv 8t^{12} + 5t^{10} + 3t^8 + 3t^6 + 3$
$p_{233} \equiv 2t^7 + 2t^5 + 4t^3 + 7t$
$p_{234} \equiv 2t^{12} + 11t^{10} + 3t^8 + 12t^6 + 11t^4 + 4t^2 + 1$
$p_{235} \equiv 7t^{11} + 7t^9 + t^7 + 12t^5 + 9t^3 + 5t$
$p_{236} \equiv 2t^{12} + 11t^{10} + 4t^8 + 6t^6 + 12t^4 + 10t^2 + 10$
$p_{237} \equiv 2t^{11} + t^9 + 6t^7 + 10t^5 + 10t^3 + 6t$
$p_{238} \equiv 12t^{12} + t^{10} + 6t^6 + 2t^4 + 4t^2 + 12$
$p_{239} \equiv 5t^{11} + 10t^9 + 9t^7 + 11t^5 + 9t^3 + 2t$
$p_{240} \equiv t^{12} + t^{10} + 4t^8 + 10t^6 + 7t^4 + 6t^2 + 3$
$p_{241} \equiv 4t^9 + 11t^7 + 12t^3 + 3t$
$p_{242} \equiv 4t^{12} + 12t^{10} + 2t^8 + 7t^6 + 10t^4 + 8t^2 + 7$
$p_{243} \equiv 5t^{11} + 12t^9 + 4t^7 + 7t^5 + 5t^3 + 6t$
$p_{244} \equiv 6t^{12} + 8t^{10} + 2t^8 + 2t^6 + 10t^4 + 3t^2 + 4$
$p_{245} \equiv 3t^{11} + 9t^9 + 8t^7 + 3t^5 + 2t^3 + 6t$
$p_{246} \equiv 9t^{12} + 9t^{10} + 4t^8 + 6t^4 + 6t^2 + 10$
$p_{247} \equiv t^{11} + t^9 + 5t^7 + 10t^5 + 5t^3 + 5t$
$p_{248} \equiv 2t^{12} + 2t^{10} + 5t^8 + 9t^6 + 3t^4 + 10t^2 + 9$
$p_{249} \equiv t^{11} + 12t^9 + 2t^7 + 2t^5 + 4t^3 + 6t$
$p_{250} \equiv 11t^{12} + 2t^8 + 9t^6 + 11t^2 + 3$
$p_{251} \equiv 11t^{11} + 11t^9 + 8t^7 + 8t^5 + 4t$
$p_{252} \equiv 2t^{12} + 12t^{10} + 8t^8 + 10t^6 + 7t^4 + 12t^2 + 4$
$p_{253} \equiv 4t^{11} + 12t^9 + 2t^5 + 11t^3 + 7t$
$p_{254} \equiv 3t^{12} + 3t^{10} + 5t^8 + 6t^6 + 5t^4 + 12t^2 + 5$
$p_{255} \equiv 6t^{11} + 4t^9 + 2t^7 + 3t^5 + 3t^3 + t$
$p_{256} \equiv 6t^{12} + 6t^{10} + t^8 + 11t^6 + 4t^4 + 11t^2 + 1$
$p_{257} \equiv 7t^{11} + 4t^9 + 9t^7 + 2t^5 + 11t^3 + 9t$

$p_{258} \equiv 5t^{12} + 4t^{10} + 2t^8 + 12t^6 + 12t^4 + t^2 + 9$
$p_{259} \equiv 3t^9 + 12t^7 + 4t^5 + 5t^3$
$p_{260} \equiv 11t^{12} + 3t^{10} + 8t^8 + 8t^6 + t^4 + 11t^2 + 12$
$p_{261} \equiv 2t^{11} + 4t^9 + 3t^7 + 10t^5 + 2t^3 + 10t$
$p_{262} \equiv 11t^{12} + 5t^{10} + 6t^8 + 3t^6 + 8t^4 + 12t^2 + 4$
$p_{263} \equiv 9t^9 + 6t^7 + 8t^5 + 7t^3$
$p_{264} \equiv 6t^{12} + 10t^{10} + 4t^8 + 4t^6 + 9t^4 + 7t^2 + 1$
$p_{265} \equiv 7t^{11} + 12t^9 + 6t^7 + 10t^5 + 10t^3$
$p_{266} \equiv 10t^{12} + 6t^{10} + 10t^8 + 2t^6 + 6t^4 + t^2 + 11$
$p_{267} \equiv 4t^{11} + 8t^9 + 11t^7 + 7t^3 + 10t$
$p_{268} \equiv 5t^{12} + 10t^{10} + 5t^8 + t^6 + 7t^4 + 3t^2 + 10$
$p_{269} \equiv 2t^{11} + 7t^7 + t^5 + 7t^3 + 10t$
$p_{270} \equiv 4t^{12} + 5t^{10} + 3t^8 + 7t^6 + 4t^4 + 4t^2 + 12$
$p_{271} \equiv 4t^{11} + 8t^9 + 5t^3 + 3t$
$p_{272} \equiv 5t^{12} + 12t^{10} + 12t^8 + 9t^6 + 5t^4 + 2t^2 + 3$
$p_{273} \equiv 7t^{11} + 11t^9 + 9t^7 + 6t^5 + 12t^3 + 2t$
$p_{274} \equiv t^{12} + t^8 + 3t^6 + 9t^4 + 6t^2 + 1$
$p_{275} \equiv 6t^9 + t^7 + t^5 + 5t^3 + 7t$
$p_{276} \equiv 7t^{10} + t^8 + 11t^6 + 6t^4 + 9t^2 + 10$
$p_{277} \equiv 8t^{11} + 8t^9 + 4t^7 + t^5 + 8t^3 + 3t$
$p_{278} \equiv 4t^{12} + 12t^8 + t^6 + 10t^4 + 2t^2 + 6$
$p_{279} \equiv 8t^{11} + 10t^9 + 2t^7 + 8t^5 + 10t^3 + 7t$
$p_{280} \equiv 6t^{12} + 6t^{10} + t^8 + 5t^6 + 6t^4 + 10t^2 + 9$
$p_{281} \equiv 12t^{11} + t^9 + 10t^7 + 9t^5 + 3t^3 + 12t$
$p_{282} \equiv 12t^{12} + 12t^{10} + 12t^8 + 8t^6 + 10t^4 + 8t^2 + 3$
$p_{283} \equiv t^{11} + 5t^9 + 12t^7 + 8t^5 + 12t^3 + 7t$
$p_{284} \equiv 10t^{12} + 7t^{10} + 8t^8 + 10t^6 + t^4 + 2t^2 + 4$
$p_{285} \equiv 3t^9 + 8t^7 + 8t^5 + 5t^3 + 5t$
$p_{286} \equiv 9t^{12} + 7t^{10} + 5t^8 + 2t^6 + 5t^4 + t^2 + 10$
$p_{287} \equiv 7t^9 + 3t^7 + t^5 + 10t^3 + 11t$
$p_{288} \equiv 9t^{12} + t^{10} + 10t^8 + 9t^6 + 2t^4 + 6t^2 + 9$
$p_{289} \equiv 7t^{11} + 12t^7 + 4t^5 + 6t^3 + 5t$
$p_{290} \equiv 10t^{12} + 3t^{10} + 11t^8 + 11t^6 + 4t^4 + 9t^2 + 8$
$p_{291} \equiv 7t^{11} + 11t^9 + 7t^7 + 4t^5 + 5t^3 + 4t$
$p_{292} \equiv 2t^{12} + t^{10} + 5t^8 + 2t^6 + 8t^4 + 10t^2 + 12$
$p_{293} \equiv 10t^{11} + 4t^9 + 6t^7 + t^5 + 10t^3 + 3t$
$p_{294} \equiv 5t^{12} + 2t^{10} + 4t^8 + 7t^6 + 8t^2 + 4$
$p_{295} \equiv 4t^{11} + 2t^9 + 5t^7 + 12t^5 + 3t^3 + 12t$
$p_{296} \equiv 11t^{12} + 9t^{10} + 12t^8 + 12t^6 + 7t^4 + 12t^2 + 1$
$p_{297} \equiv 11t^{11} + 12t^9 + 8t^7 + 8t^5 + 2t^3 + 8t$
$p_{298} \equiv 4t^{12} + t^{10} + 9t^8 + 4t^6 + 10t^4 + 8t^2 + 9$
$p_{299} \equiv 2t^{11} + 12t^9 + 12t^7 + 11t^5 + 4t^3 + 11t$
$p_{300} \equiv 4t^{12} + 8t^{10} + 11t^8 + 6t^6 + 6t^4 + 5t^2 + 12$
$p_{301} \equiv 4t^{11} + 10t^9 + 6t^7 + 11t^5 + 6t^3 + 4t$
$p_{302} \equiv 5t^{12} + 12t^{10} + 6t^8 + 6t^6 + 7t^4 + 8t^2 + 2$
$p_{303} \equiv 2t^{11} + 2t^9 + 10t^7 + t^5 + 12t$
$p_{304} \equiv 9t^{10} + 12t^8 + 10t^6 + 5t^4 + 2t^2 + 3$
$p_{305} \equiv 2t^9 + 12t^7 + 12t^5 + 3t^3 + 3t$
$p_{306} \equiv 2t^8 + 4t^6 + 9t^4 + 1$
$p_{307} \equiv 5t^7 + 3t^5 + 4t^3 + 12t$
$p_{308} \equiv 7t^6 + 10t^4 + 2t^2 + 10$
$p_{309} \equiv 6t^5 + 5t^3 + 12t$
$p_{310} \equiv 3t^4 + 10t^2 + 12$
$p_{311} \equiv 6t^3$
$p_{312} \equiv 5t^2 + 3$
$p_{313} \equiv 8t$
$p_{314} \equiv 6t^2 + 7$
$p_{315} \equiv 2t^3 + 11t$
$p_{316} \equiv 9t^2 + 4$
$p_{317} \equiv 3t^3 + 10t$
$p_{318} \equiv t^4 + 2t^2 + 10$
$p_{319} \equiv 10t^3 + 3t$
$p_{320} \equiv 2t^4 + 2t^2 + 9$
$p_{321} \equiv 5t^5 + 12t^3 + 9t$
$p_{322} \equiv 9t^4 + t^2 + 3$
$p_{323} \equiv 9t^5 + 4t$
$p_{324} \equiv 3t^6 + 9t^4 + 10t^2 + 4$
$p_{325} \equiv 4t^5 + 12t^3 + 10t$
$p_{326} \equiv 7t^6 + 4t^4 + 10t^2 + 5$
$p_{327} \equiv 11t^7 + 2t$
$p_{328} \equiv 6t^6 + 6t^2 + 1$
$p_{329} \equiv 10t^7 + 5t^5 + 3t^3 + 8t$
$p_{330} \equiv 12t^8 + 11t^6 + 7t^2 + 9$
$p_{331} \equiv 10t^5 + 2t^3 + t$
$p_{332} \equiv 11t^8 + 4t^6 + 12t^2 + 12$
$p_{333} \equiv 8t^9 + 6t^7 + 9t^3 + 3t$
$p_{334} \equiv 8t^8 + 11t^6 + t^4 + 2t^2 + 4$
$p_{335} \equiv 4t^9 + 6t^7 + t^5 + 10t^3 + 5t$
$p_{336} \equiv 10t^{10} + 10t^8 + 10t^6 + 10t^4 + 11t^2 + 1$
$p_{337} \equiv 2t^9 + 12t^7 + 10t^3 + 2t$
$p_{338} \equiv 6t^{10} + 12t^8 + 12t^6 + 11t^4 + 11$
$p_{339} \equiv 2t^{11} + 7t^9 + 11t^7 + 12t^5 + 9t^3 + 11t$
$p_{340} \equiv 11t^{10} + 7t^8 + 2t^6 + 12t^4 + 10t^2 + 10$
$p_{341} \equiv 3t^{11} + 6t^9 + 3t^7 + 11t^5 + 10t^3 + 6t$
$p_{342} \equiv t^{12} + 11t^{10} + 11t^4 + 4t^2 + 12$
$p_{343} \equiv 6t^{11} + 7t^9 + 2t^7 + 8t^5 + t^3 + 2t$
$p_{344} \equiv 2t^{12} + t^{10} + 6t^6 + 7t^4 + 7t^2 + 3$
$p_{345} \equiv 8t^{11} + 6t^9 + 11t^7 + 8t^5 + 10t^3 + 4t$
$p_{346} \equiv 3t^{12} + 8t^{10} + 8t^8 + 10t^6 + 7t^4 + 2t^2 + 1$
$p_{347} \equiv 7t^{11} + 3t^9 + 8t^5 + 9t^3 + 3t$
$p_{348} \equiv 11t^{12} + 3t^{10} + 10t^8 + 7t^6 + 4t^4 + 4t^2 + 10$
$p_{349} \equiv 6t^{11} + 3t^7 + 12t^5 + t^3 + 3t$
$p_{350} \equiv 9t^{10} + 9t^8 + 4t^6 + 8t^4 + 9t^2 + 6$
$p_{351} \equiv 3t^{11} + 10t^9 + t^7 + 10t^5 + 5t^3 + 5t$

$p_{352} \equiv 6t^{12} + t^{10} + 10t^8 + 8t^6 + 3t^4 + 10t^2 + 9$
$p_{353} \equiv 7t^{11} + 4t^9 + 9t^7 + 12t^5 + 4t^3 + 6t$
$p_{354} \equiv 12t^{12} + 2t^{10} + 5t^8 + 11t^6 + 12t^4 + 11t^2 + 3$
$p_{355} \equiv 5t^{11} + 5t^9 + 9t^7 + 4t^5 + 11t^3 + 4t$
$p_{356} \equiv 4t^{12} + 6t^{10} + 5t^8 + 3t^6 + 9t^4 + 9t^2 + 4$
$p_{357} \equiv 8t^{11} + 7t^9 + 5t^7 + 6t^5 + 10t^3 + 4t$
$p_{358} \equiv 6t^{12} + 6t^{10} + 8t^8 + t^6 + 12t^4 + 10$
$p_{359} \equiv 12t^{11} + 10t^7 + 3t^5 + 12t^3 + 3t$
$p_{360} \equiv 12t^{12} + 12t^{10} + 3t^8 + 6t^6 + 7t^4 + 4t^2 + 9$
$p_{361} \equiv t^{11} + 10t^9 + 3t^7 + 8t^5 + 5t^3$
$p_{362} \equiv 10t^{12} + 4t^8 + 2t^6 + t^4 + 6t^2 + 8$
$p_{363} \equiv 2t^9 + 9t^7 + 10t^5 + 12t^3 + 7t$
$p_{364} \equiv 9t^{12} + 10t^{10} + 12t^8 + 11t^6 + t^4 + 11t^2 + 12$
$p_{365} \equiv t^{11} + t^9 + 6t^5 + 2t^3 + 10t$
$p_{366} \equiv 9t^{12} + 10t^{10} + 10t^8 + 3t^6 + 11t^4 + 12t^2 + 4$
$p_{367} \equiv 6t^9 + 9t^7 + 6t^5$
$p_{368} \equiv 12t^{12} + 4t^{10} + 5t^8 + 10t^6 + 8t^4 + 3t^2 + 1$
$p_{369} \equiv 11t^{11} + 8t^9 + 4t^7 + t^5 + 5t^3 + 9t$
$p_{370} \equiv 5t^{12} + 3t^{10} + 2t^8 + 2t^6 + 12t^4 + 7t^2 + 9$
$p_{371} \equiv 4t^{11} + t^9 + t^7 + 8t^5 + 2t^3 + 4t$
$p_{372} \equiv 11t^{12} + 10t^{10} + 9t^8 + 11t^6 + 3t^2 + 12$
$p_{373} \equiv 12t^{11} + 11t^7 + 12t^3 + 8t$
$p_{374} \equiv 12t^{12} + 8t^{10} + 7t^8 + 2t^6 + 12t^4 + t^2 + 2$
$p_{375} \equiv 12t^{11} + 9t^9 + t^7 + t^5 + 6t^3 + 12t$
$p_{376} \equiv 2t^{12} + 12t^8 + 3t^6 + 12t^4 + 3t^2 + 3$
$p_{377} \equiv 2t^{11} + 6t^7 + 5t^5 + 12t^3 + 2t$
$p_{378} \equiv 4t^{12} + 11t^{10} + 11t^8 + 7t^6 + 6t^2 + 1$
$p_{379} \equiv 12t^{11} + 9t^7 + 7t$
$p_{380} \equiv 2t^{12} + t^{10} + 12t^6 + 8t^2 + 10$
$p_{381} \equiv 5t^{11} + 5t^9 + 7t^7 + 10t^5 + 4t^3 + 2t$
$p_{382} \equiv 6t^{12} + t^{10} + t^8 + 9t^6 + 4t^4 + 5t^2 + 12$
$p_{383} \equiv 12t^{11} + 8t^9 + 12t^7 + 2t^5 + 2t^3 + 10t$
$p_{384} \equiv 9t^{12} + 9t^{10} + 7t^8 + 7t^6 + t^4 + 2t^2 + 3$
$p_{385} \equiv 5t^{11} + 10t^9 + 3t^7 + t^5 + 8t^3 + 3t$
$p_{386} \equiv 5t^{12} + 5t^{10} + 11t^8 + 8t^6 + 11t^4 + 6t^2 + 7$
$p_{387} \equiv 8t^{11} + 9t^9 + 7t^7 + 12t^5 + 8t^3 + 10t$
$p_{388} \equiv 2t^{12} + 11t^{10} + 4t^8 + 4t^6 + 4$
$p_{389} \equiv 7t^7 + 7t^5 + t^3 + 5t$
$p_{390} \equiv 7t^{12} + 6t^{10} + 4t^8 + 3t^6 + 6t^4 + t^2 + 10$
$p_{391} \equiv 5t^{11} + 5t^9 + 10t^7 + 3t^5 + 12t^3 + 11t$
$p_{392} \equiv 7t^{12} + 6t^{10} + t^8 + 8t^6 + 3t^4 + 9t^2 + 9$
$p_{393} \equiv 7t^{11} + 10t^9 + 8t^7 + 9t^5 + 9t^3 + 8t$
$p_{394} \equiv 3t^{12} + 10t^{10} + 8t^6 + 7t^4 + t^2 + 3$
$p_{395} \equiv 11t^{11} + 9t^9 + 12t^7 + 6t^5 + 12t^3 + 7t$
$p_{396} \equiv 10t^{12} + 10t^{10} + t^8 + 9t^6 + 5t^4 + 8t^2 + 4$
$p_{397} \equiv t^9 + 6t^7 + 3t^3 + 4t$
$p_{398} \equiv t^{12} + 3t^{10} + 7t^8 + 5t^6 + 9t^4 + 2t^2 + 5$
$p_{399} \equiv 11t^{11} + 3t^9 + t^7 + 5t^5 + 11t^3 + 8t$
$p_{400} \equiv 8t^{12} + 2t^{10} + 7t^8 + 7t^6 + 9t^4 + 4t^2 + 1$
$p_{401} \equiv 4t^{11} + 12t^9 + 2t^7 + 4t^5 + 7t^3 + 8t$
$p_{402} \equiv 12t^{12} + 12t^{10} + t^8 + 8t^4 + 8t^2 + 9$
$p_{403} \equiv 10t^{11} + 10t^9 + 11t^7 + 9t^5 + 11t^3 + 11t$
$p_{404} \equiv 7t^{12} + 7t^{10} + 11t^8 + 12t^6 + 4t^4 + 9t^2 + 12$
$p_{405} \equiv 10t^{11} + 3t^9 + 7t^7 + 7t^5 + t^3 + 8t$
$p_{406} \equiv 6t^{12} + 7t^8 + 12t^6 + 6t^2 + 4$
$p_{407} \equiv 6t^{11} + 6t^9 + 2t^7 + 2t^5 + t$
$p_{408} \equiv 7t^{12} + 3t^{10} + 2t^8 + 9t^6 + 5t^4 + 3t^2 + 1$
$p_{409} \equiv t^{11} + 3t^9 + 7t^5 + 6t^3 + 5t$
$p_{410} \equiv 4t^{12} + 4t^{10} + 11t^8 + 8t^6 + 11t^4 + 3t^2 + 11$
$p_{411} \equiv 8t^{11} + t^9 + 7t^7 + 4t^5 + 4t^3 + 10t$
$p_{412} \equiv 8t^{12} + 8t^{10} + 10t^8 + 6t^6 + t^4 + 6t^2 + 10$
$p_{413} \equiv 5t^{11} + t^9 + 12t^7 + 7t^5 + 6t^3 + 12t$
$p_{414} \equiv 11t^{12} + t^{10} + 7t^8 + 3t^6 + 3t^4 + 10t^2 + 12$
$p_{415} \equiv 4t^9 + 3t^7 + t^5 + 11t^3$
$p_{416} \equiv 6t^{12} + 4t^{10} + 2t^8 + 2t^6 + 10t^4 + 6t^2 + 3$
$p_{417} \equiv 7t^{11} + t^9 + 4t^7 + 9t^5 + 7t^3 + 9t$
$p_{418} \equiv 6t^{12} + 11t^{10} + 8t^8 + 4t^6 + 2t^4 + 3t^2 + 1$
$p_{419} \equiv 12t^9 + 8t^7 + 2t^5 + 5t^3$
$p_{420} \equiv 8t^{12} + 9t^{10} + t^8 + t^6 + 12t^4 + 5t^2 + 10$
$p_{421} \equiv 5t^{11} + 3t^9 + 8t^7 + 9t^5 + 9t^3$
$p_{422} \equiv 9t^{12} + 8t^{10} + 9t^8 + 7t^6 + 8t^4 + 10t^2 + 6$
$p_{423} \equiv t^{11} + 2t^9 + 6t^7 + 5t^3 + 9t$
$p_{424} \equiv 11t^{12} + 9t^{10} + 11t^8 + 10t^6 + 5t^4 + 4t^2 + 9$
$p_{425} \equiv 7t^{11} + 5t^7 + 10t^5 + 5t^3 + 9t$
$p_{426} \equiv t^{12} + 11t^{10} + 4t^8 + 5t^6 + t^4 + t^2 + 3$
$p_{427} \equiv t^{11} + 2t^9 + 11t^3 + 4t$
$p_{428} \equiv 11t^{12} + 3t^{10} + 3t^8 + 12t^6 + 11t^4 + 7t^2 + 4$
$p_{429} \equiv 5t^{11} + 6t^9 + 12t^7 + 8t^5 + 3t^3 + 7t$
$p_{430} \equiv 10t^{12} + 10t^8 + 4t^6 + 12t^4 + 8t^2 + 10$
$p_{431} \equiv 8t^9 + 10t^7 + 10t^5 + 11t^3 + 5t$
$p_{432} \equiv 5t^{10} + 10t^8 + 6t^6 + 8t^4 + 12t^2 + 9$
$p_{433} \equiv 2t^{11} + 2t^9 + t^7 + 10t^5 + 2t^3 + 4t$
$p_{434} \equiv t^{12} + 3t^8 + 10t^6 + 9t^4 + 7t^2 + 8$
$p_{435} \equiv 2t^{11} + 9t^9 + 7t^7 + 2t^5 + 9t^3 + 5t$
$p_{436} \equiv 8t^{12} + 8t^{10} + 10t^8 + 11t^6 + 8t^4 + 9t^2 + 12$
$p_{437} \equiv 3t^{11} + 10t^9 + 9t^7 + 12t^5 + 4t^3 + 3t$
$p_{438} \equiv 3t^{12} + 3t^{10} + 3t^8 + 2t^6 + 9t^4 + 2t^2 + 4$
$p_{439} \equiv 10t^{11} + 11t^9 + 3t^7 + 2t^5 + 3t^3 + 5t$
$p_{440} \equiv 9t^{12} + 5t^{10} + 2t^8 + 9t^6 + 10t^4 + 7t^2 + 1$
$p_{441} \equiv 4t^9 + 2t^7 + 2t^5 + 11t^3 + 11t$
$p_{442} \equiv 12t^{12} + 5t^{10} + 11t^8 + 7t^6 + 11t^4 + 10t^2 + 9$
$p_{443} \equiv 5t^9 + 4t^7 + 10t^5 + 9t^3 + 6t$
$p_{444} \equiv 12t^{12} + 10t^{10} + 9t^8 + 12t^6 + 7t^4 + 8t^2 + 12$
$p_{445} \equiv 5t^{11} + 3t^7 + t^5 + 8t^3 + 11t$

$p_{446} \equiv 9t^{12} + 4t^{10} + 6t^8 + 6t^6 + t^4 + 12t^2 + 2$
$p_{447} \equiv 5t^{11} + 6t^9 + 5t^7 + t^5 + 11t^3 + t$
$p_{448} \equiv 7t^{12} + 10t^{10} + 11t^8 + 7t^6 + 2t^4 + 9t^2 + 3$
$p_{449} \equiv 9t^{11} + t^9 + 8t^7 + 10t^5 + 9t^3 + 4t$
$p_{450} \equiv 11t^{12} + 7t^{10} + t^8 + 5t^6 + 2t^2 + 1$
$p_{451} \equiv t^{11} + 7t^9 + 11t^7 + 3t^5 + 4t^3 + 3t$
$p_{452} \equiv 6t^{12} + 12t^{10} + 3t^8 + 3t^6 + 5t^4 + 3t^2 + 10$
$p_{453} \equiv 6t^{11} + 3t^9 + 2t^7 + 2t^5 + 7t^3 + 2t$
$p_{454} \equiv t^{12} + 10t^{10} + 12t^8 + t^6 + 9t^4 + 2t^2 + 12$
$p_{455} \equiv 7t^{11} + 3t^9 + 3t^7 + 6t^5 + t^3 + 6t$
$p_{456} \equiv t^{12} + 2t^{10} + 6t^8 + 8t^6 + 8t^4 + 11t^2 + 3$
$p_{457} \equiv t^{11} + 9t^9 + 8t^7 + 6t^5 + 8t^3 + t$
$p_{458} \equiv 11t^{12} + 3t^{10} + 8t^8 + 8t^6 + 5t^4 + 2t^2 + 7$
$p_{459} \equiv 7t^{11} + 7t^9 + 9t^7 + 10t^5 + 3t$
$p_{460} \equiv 12t^{10} + 3t^8 + 9t^6 + 11t^4 + 7t^2 + 4$
$p_{461} \equiv 7t^9 + 3t^7 + 3t^5 + 4t^3 + 4t$
$p_{462} \equiv 7t^8 + t^6 + 12t^4 + 10$
$p_{463} \equiv 11t^7 + 4t^5 + t^3 + 3t$
$p_{464} \equiv 5t^6 + 9t^4 + 7t^2 + 9$
$p_{465} \equiv 8t^5 + 11t^3 + 3t$
$p_{466} \equiv 4t^4 + 9t^2 + 3$
$p_{467} \equiv 8t^3$
$p_{468} \equiv 11t^2 + 4$
$p_{469} \equiv 2t$
$p_{470} \equiv 8t^2 + 5$
$p_{471} \equiv 7t^3 + 6t$
$p_{472} \equiv 12t^2 + 1$
$p_{473} \equiv 4t^3 + 9t$
$p_{474} \equiv 10t^4 + 7t^2 + 9$
$p_{475} \equiv 9t^3 + 4t$
$p_{476} \equiv 7t^4 + 7t^2 + 12$
$p_{477} \equiv 11t^5 + 3t^3 + 12t$
$p_{478} \equiv 12t^4 + 10t^2 + 4$
$p_{479} \equiv 12t^5 + t$
$p_{480} \equiv 4t^6 + 12t^4 + 9t^2 + 1$
$p_{481} \equiv t^5 + 3t^3 + 9t$
$p_{482} \equiv 5t^6 + t^4 + 9t^2 + 11$
$p_{483} \equiv 6t^7 + 7t$
$p_{484} \equiv 8t^6 + 8t^2 + 10$
$p_{485} \equiv 9t^7 + 11t^5 + 4t^3 + 2t$
$p_{486} \equiv 3t^8 + 6t^6 + 5t^2 + 12$
$p_{487} \equiv 9t^5 + 7t^3 + 10t$
$p_{488} \equiv 6t^8 + t^6 + 3t^2 + 3$
$p_{489} \equiv 2t^9 + 8t^7 + 12t^3 + 4t$
$p_{490} \equiv 2t^8 + 6t^6 + 10t^4 + 7t^2 + 1$
$p_{491} \equiv t^9 + 8t^7 + 10t^5 + 9t^3 + 11t$
$p_{492} \equiv 9t^{10} + 9t^8 + 9t^6 + 9t^4 + 6t^2 + 10$
$p_{493} \equiv 7t^9 + 3t^7 + 9t^3 + 7t$
$p_{494} \equiv 8t^{10} + 3t^8 + 3t^6 + 6t^4 + 6$
$p_{495} \equiv 7t^{11} + 5t^9 + 6t^7 + 3t^5 + 12t^3 + 6t$
$p_{496} \equiv 6t^{10} + 5t^8 + 7t^6 + 3t^4 + 9t^2 + 9$
$p_{497} \equiv 4t^{11} + 8t^9 + 4t^7 + 6t^5 + 9t^3 + 8t$
$p_{498} \equiv 10t^{12} + 6t^{10} + 6t^4 + t^2 + 3$
$p_{499} \equiv 8t^{11} + 5t^9 + 7t^7 + 2t^5 + 10t^3 + 7t$
$p_{500} \equiv 7t^{12} + 10t^{10} + 8t^6 + 5t^4 + 5t^2 + 4$
$p_{501} \equiv 2t^{11} + 8t^9 + 6t^7 + 2t^5 + 9t^3 + t$
$p_{502} \equiv 4t^{12} + 2t^{10} + 2t^8 + 9t^6 + 5t^4 + 7t^2 + 10$
$p_{503} \equiv 5t^{11} + 4t^9 + 2t^5 + 12t^3 + 4t$
$p_{504} \equiv 6t^{12} + 4t^{10} + 9t^8 + 5t^6 + t^4 + t^2 + 9$
$p_{505} \equiv 8t^{11} + 4t^7 + 3t^5 + 10t^3 + 4t$
$p_{506} \equiv 12t^{10} + 12t^8 + t^6 + 2t^4 + 12t^2 + 8$
$p_{507} \equiv 4t^{11} + 9t^9 + 10t^7 + 9t^5 + 11t^3 + 11t$
$p_{508} \equiv 8t^{12} + 10t^{10} + 9t^8 + 2t^6 + 4t^4 + 9t^2 + 12$
$p_{509} \equiv 5t^{11} + t^9 + 12t^7 + 3t^5 + t^3 + 8t$
$p_{510} \equiv 3t^{12} + 7t^{10} + 11t^8 + 6t^6 + 3t^4 + 6t^2 + 4$
$p_{511} \equiv 11t^{11} + 11t^9 + 12t^7 + t^5 + 6t^3 + t$
$p_{512} \equiv t^{12} + 8t^{10} + 11t^8 + 4t^6 + 12t^4 + 12t^2 + 1$
$p_{513} \equiv 2t^{11} + 5t^9 + 11t^7 + 8t^5 + 9t^3 + t$
$p_{514} \equiv 8t^{12} + 8t^{10} + 2t^8 + 10t^6 + 3t^4 + 9$
$p_{515} \equiv 3t^{11} + 9t^7 + 4t^5 + 3t^3 + 4t$
$p_{516} \equiv 3t^{12} + 3t^{10} + 4t^8 + 8t^6 + 5t^4 + t^2 + 12$
$p_{517} \equiv 10t^{11} + 9t^9 + 4t^7 + 2t^5 + 11t^3$
$p_{518} \equiv 9t^{12} + t^8 + 7t^6 + 10t^4 + 8t^2 + 2$
$p_{519} \equiv 7t^9 + 12t^7 + 9t^5 + 3t^3 + 5t$
$p_{520} \equiv 12t^{12} + 9t^{10} + 3t^8 + 6t^6 + 10t^4 + 6t^2 + 3$
$p_{521} \equiv 10t^{11} + 10t^9 + 8t^5 + 7t^3 + 9t$
$p_{522} \equiv 12t^{12} + 9t^{10} + 9t^8 + 4t^6 + 6t^4 + 3t^2 + 1$
$p_{523} \equiv 8t^9 + 12t^7 + 8t^5$
$p_{524} \equiv 3t^{12} + t^{10} + 11t^8 + 9t^6 + 2t^4 + 4t^2 + 10$
$p_{525} \equiv 6t^{11} + 2t^9 + t^7 + 10t^5 + 11t^3 + 12t$
$p_{526} \equiv 11t^{12} + 4t^{10} + 7t^8 + 7t^6 + 3t^4 + 5t^2 + 12$
$p_{527} \equiv t^{11} + 10t^9 + 10t^7 + 2t^5 + 7t^3 + t$
$p_{528} \equiv 6t^{12} + 9t^{10} + 12t^8 + 6t^6 + 4t^2 + 3$
$p_{529} \equiv 3t^{11} + 6t^7 + 3t^3 + 2t$
$p_{530} \equiv 3t^{12} + 2t^{10} + 5t^8 + 7t^6 + 3t^4 + 10t^2 + 7$
$p_{531} \equiv 3t^{11} + 12t^9 + 10t^7 + 10t^5 + 8t^3 + 3t$
$p_{532} \equiv 7t^{12} + 3t^8 + 4t^6 + 3t^4 + 4t^2 + 4$
$p_{533} \equiv 7t^{11} + 8t^7 + 11t^5 + 3t^3 + 7t$
$p_{534} \equiv t^{12} + 6t^{10} + 6t^8 + 5t^6 + 8t^2 + 10$
$p_{535} \equiv 3t^{11} + 12t^7 + 5t$
$p_{536} \equiv 7t^{12} + 10t^{10} + 3t^6 + 2t^2 + 9$
$p_{537} \equiv 11t^{11} + 11t^9 + 5t^7 + 9t^5 + t^3 + 7t$
$p_{538} \equiv 8t^{12} + 10t^{10} + 10t^8 + 12t^6 + t^4 + 11t^2 + 3$
$p_{539} \equiv 3t^{11} + 2t^9 + 3t^7 + 7t^5 + 7t^3 + 9t$

$p_{540} \equiv 12t^{12} + 12t^{10} + 5t^8 + 5t^6 + 10t^4 + 7t^2 + 4$
$p_{541} \equiv 11t^{11} + 9t^9 + 4t^7 + 10t^5 + 2t^3 + 4t$
$p_{542} \equiv 11t^{12} + 11t^{10} + 6t^8 + 2t^6 + 6t^4 + 8t^2 + 5$
$p_{543} \equiv 2t^{11} + 12t^9 + 5t^7 + 3t^5 + 2t^3 + 9t$
$p_{544} \equiv 7t^{12} + 6t^{10} + t^8 + t^6 + 1$
$p_{545} \equiv 5t^7 + 5t^5 + 10t^3 + 11t$
$p_{546} \equiv 5t^{12} + 8t^{10} + t^8 + 4t^6 + 8t^4 + 10t^2 + 9$
$p_{547} \equiv 11t^{11} + 11t^9 + 9t^7 + 4t^5 + 3t^3 + 6t$
$p_{548} \equiv 5t^{12} + 8t^{10} + 10t^8 + 2t^6 + 4t^4 + 12t^2 + 12$
$p_{549} \equiv 5t^{11} + 9t^9 + 2t^7 + 12t^5 + 12t^3 + 2t$
$p_{550} \equiv 4t^{12} + 9t^{10} + 2t^6 + 5t^4 + 10t^2 + 4$
$p_{551} \equiv 6t^{11} + 12t^9 + 3t^7 + 8t^5 + 3t^3 + 5t$
$p_{552} \equiv 9t^{12} + 9t^{10} + 10t^8 + 12t^6 + 11t^4 + 2t^2 + 1$
$p_{553} \equiv 10t^9 + 8t^7 + 4t^3 + t$
$p_{554} \equiv 10t^{12} + 4t^{10} + 5t^8 + 11t^6 + 12t^4 + 7t^2 + 11$
$p_{555} \equiv 6t^{11} + 4t^9 + 10t^7 + 11t^5 + 6t^3 + 2t$
$p_{556} \equiv 2t^{12} + 7t^{10} + 5t^8 + 5t^6 + 12t^4 + t^2 + 10$
$p_{557} \equiv t^{11} + 3t^9 + 7t^7 + t^5 + 5t^3 + 2t$
$p_{558} \equiv 3t^{12} + 3t^{10} + 10t^8 + 2t^4 + 2t^2 + 12$
$p_{559} \equiv 9t^{11} + 9t^9 + 6t^7 + 12t^5 + 6t^3 + 6t$
$p_{560} \equiv 5t^{12} + 5t^{10} + 6t^8 + 3t^6 + t^4 + 12t^2 + 3$
$p_{561} \equiv 9t^{11} + 4t^9 + 5t^7 + 5t^5 + 10t^3 + 2t$
$p_{562} \equiv 8t^{12} + 5t^8 + 3t^6 + 8t^2 + 1$
$p_{563} \equiv 8t^{11} + 8t^9 + 7t^7 + 7t^5 + 10t$
$p_{564} \equiv 5t^{12} + 4t^{10} + 7t^8 + 12t^6 + 11t^4 + 4t^2 + 10$
$p_{565} \equiv 10t^{11} + 4t^9 + 5t^5 + 8t^3 + 11t$
$p_{566} \equiv t^{12} + t^{10} + 6t^8 + 2t^6 + 6t^4 + 4t^2 + 6$
$p_{567} \equiv 2t^{11} + 10t^9 + 5t^7 + t^5 + t^3 + 9t$
$p_{568} \equiv 2t^{12} + 2t^{10} + 9t^8 + 8t^6 + 10t^4 + 8t^2 + 9$
$p_{569} \equiv 11t^{11} + 10t^9 + 3t^7 + 5t^5 + 8t^3 + 3t$
$p_{570} \equiv 6t^{12} + 10t^{10} + 5t^8 + 4t^6 + 4t^4 + 9t^2 + 3$
$p_{571} \equiv t^9 + 4t^7 + 10t^5 + 6t^3$
$p_{572} \equiv 8t^{12} + t^{10} + 7t^8 + 7t^6 + 9t^4 + 8t^2 + 4$
$p_{573} \equiv 5t^{11} + 10t^9 + t^7 + 12t^5 + 5t^3 + 12t$
$p_{574} \equiv 8t^{12} + 6t^{10} + 2t^8 + t^6 + 7t^4 + 4t^2 + 10$
$p_{575} \equiv 3t^9 + 2t^7 + 7t^5 + 11t^3$
$p_{576} \equiv 2t^{12} + 12t^{10} + 10t^8 + 10t^6 + 3t^4 + 11t^2 + 9$
$p_{577} \equiv 11t^{11} + 4t^9 + 2t^7 + 12t^5 + 12t^3$
$p_{578} \equiv 12t^{12} + 2t^{10} + 12t^8 + 5t^6 + 2t^4 + 9t^2 + 8$
$p_{579} \equiv 10t^{11} + 7t^9 + 8t^7 + 11t^3 + 12t$
$p_{580} \equiv 6t^{12} + 12t^{10} + 6t^8 + 9t^6 + 11t^4 + t^2 + 12$
$p_{581} \equiv 5t^{11} + 11t^7 + 9t^5 + 11t^3 + 12t$
$p_{582} \equiv 10t^{12} + 6t^{10} + t^8 + 11t^6 + 10t^4 + 10t^2 + 4$
$p_{583} \equiv 10t^{11} + 7t^9 + 6t^3 + t$
$p_{584} \equiv 6t^{12} + 4t^{10} + 4t^8 + 3t^6 + 6t^4 + 5t^2 + 1$
$p_{585} \equiv 11t^{11} + 8t^9 + 3t^7 + 2t^5 + 4t^3 + 5t$
$p_{586} \equiv 9t^{12} + 9t^8 + t^6 + 3t^4 + 2t^2 + 9$
$p_{587} \equiv 2t^9 + 9t^7 + 9t^5 + 6t^3 + 11t$
$p_{588} \equiv 11t^{10} + 9t^8 + 8t^6 + 2t^4 + 3t^2 + 12$
$p_{589} \equiv 7t^{11} + 7t^9 + 10t^7 + 9t^5 + 7t^3 + t$
$p_{590} \equiv 10t^{12} + 4t^8 + 9t^6 + 12t^4 + 5t^2 + 2$
$p_{591} \equiv 7t^{11} + 12t^9 + 5t^7 + 7t^5 + 12t^3 + 11t$
$p_{592} \equiv 2t^{12} + 2t^{10} + 9t^8 + 6t^6 + 2t^4 + 12t^2 + 3$
$p_{593} \equiv 4t^{11} + 9t^9 + 12t^7 + 3t^5 + t^3 + 4t$
$p_{594} \equiv 4t^{12} + 4t^{10} + 4t^8 + 7t^6 + 12t^4 + 7t^2 + 1$
$p_{595} \equiv 9t^{11} + 6t^9 + 4t^7 + 7t^5 + 4t^3 + 11t$
$p_{596} \equiv 12t^{12} + 11t^{10} + 7t^8 + 12t^6 + 9t^4 + 5t^2 + 10$
$p_{597} \equiv t^9 + 7t^7 + 7t^5 + 6t^3 + 6t$
$p_{598} \equiv 3t^{12} + 11t^{10} + 6t^8 + 5t^6 + 6t^4 + 9t^2 + 12$
$p_{599} \equiv 11t^9 + t^7 + 9t^5 + 12t^3 + 8t$
$p_{600} \equiv 3t^{12} + 9t^{10} + 12t^8 + 3t^6 + 5t^4 + 2t^2 + 3$
$p_{601} \equiv 11t^{11} + 4t^7 + 10t^5 + 2t^3 + 6t$
$p_{602} \equiv 12t^{12} + t^{10} + 8t^8 + 8t^6 + 10t^4 + 3t^2 + 7$
$p_{603} \equiv 11t^{11} + 8t^9 + 11t^7 + 10t^5 + 6t^3 + 10t$
$p_{604} \equiv 5t^{12} + 9t^{10} + 6t^8 + 5t^6 + 7t^4 + 12t^2 + 4$
$p_{605} \equiv 12t^{11} + 10t^9 + 2t^7 + 9t^5 + 12t^3 + t$
$p_{606} \equiv 6t^{12} + 5t^{10} + 10t^8 + 11t^6 + 7t^2 + 10$
$p_{607} \equiv 10t^{11} + 5t^9 + 6t^7 + 4t^5 + t^3 + 4t$
$p_{608} \equiv 8t^{12} + 3t^{10} + 4t^8 + 4t^6 + 11t^4 + 4t^2 + 9$
$p_{609} \equiv 8t^{11} + 4t^9 + 7t^7 + 7t^5 + 5t^3 + 7t$
$p_{610} \equiv 10t^{12} + 9t^{10} + 3t^8 + 10t^6 + 12t^4 + 7t^2 + 3$
$p_{611} \equiv 5t^{11} + 4t^9 + 4t^7 + 8t^5 + 10t^3 + 8t$
$p_{612} \equiv 10t^{12} + 7t^{10} + 8t^8 + 2t^6 + 2t^4 + 6t^2 + 4$
$p_{613} \equiv 10t^{11} + 12t^9 + 2t^7 + 8t^5 + 2t^3 + 10t$
$p_{614} \equiv 6t^{12} + 4t^{10} + 2t^8 + 2t^6 + 11t^4 + 7t^2 + 5$
$p_{615} \equiv 5t^{11} + 5t^9 + 12t^7 + 9t^5 + 4t$
$p_{616} \equiv 3t^{10} + 4t^8 + 12t^6 + 6t^4 + 5t^2 + 1$
$p_{617} \equiv 5t^9 + 4t^7 + 4t^5 + t^3 + t$
$p_{618} \equiv 5t^8 + 10t^6 + 3t^4 + 9$
$p_{619} \equiv 6t^7 + t^5 + 10t^3 + 4t$
$p_{620} \equiv 11t^6 + 12t^4 + 5t^2 + 12$
$p_{621} \equiv 2t^5 + 6t^3 + 4t$
$p_{622} \equiv t^4 + 12t^2 + 4$
$p_{623} \equiv 2t^3$
$p_{624} \equiv 6t^2 + 1$
$p_{625} \equiv 7t$
$p_{626} \equiv 2t^2 + 11$
$p_{627} \equiv 5t^3 + 8t$
$p_{628} \equiv 3t^2 + 10$
$p_{629} \equiv t^3 + 12t$
$p_{630} \equiv 9t^4 + 5t^2 + 12$
$p_{631} \equiv 12t^3 + t$
$p_{632} \equiv 5t^4 + 5t^2 + 3$
$p_{633} \equiv 6t^5 + 4t^3 + 3t$

$p_{634} \equiv 3t^4 + 9t^2 + 1$
$p_{635} \equiv 3t^5 + 10t$
$p_{636} \equiv t^6 + 3t^4 + 12t^2 + 10$
$p_{637} \equiv 10t^5 + 4t^3 + 12t$
$p_{638} \equiv 11t^6 + 10t^4 + 12t^2 + 6$
$p_{639} \equiv 8t^7 + 5t$
$p_{640} \equiv 2t^6 + 2t^2 + 9$
$p_{641} \equiv 12t^7 + 6t^5 + t^3 + 7t$
$p_{642} \equiv 4t^8 + 8t^6 + 11t^2 + 3$
$p_{643} \equiv 12t^5 + 5t^3 + 9t$
$p_{644} \equiv 8t^8 + 10t^6 + 4t^2 + 4$
$p_{645} \equiv 7t^9 + 2t^7 + 3t^3 + t$
$p_{646} \equiv 7t^8 + 8t^6 + 9t^4 + 5t^2 + 10$
$p_{647} \equiv 10t^9 + 2t^7 + 9t^5 + 12t^3 + 6t$
$p_{648} \equiv 12t^{10} + 12t^8 + 12t^6 + 12t^4 + 8t^2 + 9$
$p_{649} \equiv 5t^9 + 4t^7 + 12t^3 + 5t$
$p_{650} \equiv 2t^{10} + 4t^8 + 4t^6 + 8t^4 + 8$
$p_{651} \equiv 5t^{11} + 11t^9 + 8t^7 + 4t^5 + 3t^3 + 8t$
$p_{652} \equiv 8t^{10} + 11t^8 + 5t^6 + 4t^4 + 12t^2 + 12$
$p_{653} \equiv t^{11} + 2t^9 + t^7 + 8t^5 + 12t^3 + 2t$
$p_{654} \equiv 9t^{12} + 8t^{10} + 8t^4 + 10t^2 + 4$
$p_{655} \equiv 2t^{11} + 11t^9 + 5t^7 + 7t^5 + 9t^3 + 5t$
$p_{656} \equiv 5t^{12} + 9t^{10} + 2t^6 + 11t^4 + 11t^2 + 1$
$p_{657} \equiv 7t^{11} + 2t^9 + 8t^7 + 7t^5 + 12t^3 + 10t$
$p_{658} \equiv t^{12} + 7t^{10} + 7t^8 + 12t^6 + 11t^4 + 5t^2 + 9$
$p_{659} \equiv 11t^{11} + t^9 + 7t^5 + 3t^3 + t$
$p_{660} \equiv 8t^{12} + t^{10} + 12t^8 + 11t^6 + 10t^4 + 10t^2 + 12$
$p_{661} \equiv 2t^{11} + t^7 + 4t^5 + 9t^3 + t$
$p_{662} \equiv 3t^{10} + 3t^8 + 10t^6 + 7t^4 + 3t^2 + 2$
$p_{663} \equiv t^{11} + 12t^9 + 9t^7 + 12t^5 + 6t^3 + 6t$
$p_{664} \equiv 2t^{12} + 9t^{10} + 12t^8 + 7t^6 + t^4 + 12t^2 + 3$
$p_{665} \equiv 11t^{11} + 10t^9 + 3t^7 + 4t^5 + 10t^3 + 2t$
$p_{666} \equiv 4t^{12} + 5t^{10} + 6t^8 + 8t^6 + 4t^4 + 8t^2 + 1$
$p_{667} \equiv 6t^{11} + 6t^9 + 3t^7 + 10t^5 + 8t^3 + 10t$
$p_{668} \equiv 10t^{12} + 2t^{10} + 6t^8 + t^6 + 3t^4 + 3t^2 + 10$
$p_{669} \equiv 7t^{11} + 11t^9 + 6t^7 + 2t^5 + 12t^3 + 10t$
$p_{670} \equiv 2t^{12} + 2t^{10} + 7t^8 + 9t^6 + 4t^4 + 12$
$p_{671} \equiv 4t^{11} + 12t^7 + t^5 + 4t^3 + t$
$p_{672} \equiv 4t^{12} + 4t^{10} + t^8 + 2t^6 + 11t^4 + 10t^2 + 3$
$p_{673} \equiv 9t^{11} + 12t^9 + t^7 + 7t^5 + 6t^3$
$p_{674} \equiv 12t^{12} + 10t^8 + 5t^6 + 9t^4 + 2t^2 + 7$
$p_{675} \equiv 5t^9 + 3t^7 + 12t^5 + 4t^3 + 11t$
$p_{676} \equiv 3t^{12} + 12t^{10} + 4t^8 + 8t^6 + 9t^4 + 8t^2 + 4$
$p_{677} \equiv 9t^{11} + 9t^9 + 2t^5 + 5t^3 + 12t$
$p_{678} \equiv 3t^{12} + 12t^{10} + 12t^8 + t^6 + 8t^4 + 4t^2 + 10$
$p_{679} \equiv 2t^9 + 3t^7 + 2t^5$
$p_{680} \equiv 4t^{12} + 10t^{10} + 6t^8 + 12t^6 + 7t^4 + t^2 + 9$
$p_{681} \equiv 8t^{11} + 7t^9 + 10t^7 + 9t^5 + 6t^3 + 3t$
$p_{682} \equiv 6t^{12} + t^{10} + 5t^8 + 5t^6 + 4t^4 + 11t^2 + 3$
$p_{683} \equiv 10t^{11} + 9t^9 + 9t^7 + 7t^5 + 5t^3 + 10t$
$p_{684} \equiv 8t^{12} + 12t^{10} + 3t^8 + 8t^6 + t^2 + 4$
$p_{685} \equiv 4t^{11} + 8t^7 + 4t^3 + 7t$
$p_{686} \equiv 4t^{12} + 7t^{10} + 11t^8 + 5t^6 + 4t^4 + 9t^2 + 5$
$p_{687} \equiv 4t^{11} + 3t^9 + 9t^7 + 9t^5 + 2t^3 + 4t$
$p_{688} \equiv 5t^{12} + 4t^8 + t^6 + 4t^4 + t^2 + 1$
$p_{689} \equiv 5t^{11} + 2t^7 + 6t^5 + 4t^3 + 5t$
$p_{690} \equiv 10t^{12} + 8t^{10} + 8t^8 + 11t^6 + 2t^2 + 9$
$p_{691} \equiv 4t^{11} + 3t^7 + 11t$
$p_{692} \equiv 5t^{12} + 9t^{10} + 4t^6 + 7t^2 + 12$
$p_{693} \equiv 6t^{11} + 6t^9 + 11t^7 + 12t^5 + 10t^3 + 5t$
$p_{694} \equiv 2t^{12} + 9t^{10} + 9t^8 + 3t^6 + 10t^4 + 6t^2 + 4$
$p_{695} \equiv 4t^{11} + 7t^9 + 4t^7 + 5t^5 + 5t^3 + 12t$
$p_{696} \equiv 3t^{12} + 3t^{10} + 11t^8 + 11t^6 + 9t^4 + 5t^2 + 1$
$p_{697} \equiv 6t^{11} + 12t^9 + t^7 + 9t^5 + 7t^3 + t$
$p_{698} \equiv 6t^{12} + 6t^{10} + 8t^8 + 7t^6 + 8t^4 + 2t^2 + 11$
$p_{699} \equiv 7t^{11} + 3t^9 + 11t^7 + 4t^5 + 7t^3 + 12t$
$p_{700} \equiv 5t^{12} + 8t^{10} + 10t^8 + 10t^6 + 10$
$p_{701} \equiv 11t^7 + 11t^5 + 9t^3 + 6t$
$p_{702} \equiv 11t^{12} + 2t^{10} + 10t^8 + t^6 + 2t^4 + 9t^2 + 12$
$p_{703} \equiv 6t^{11} + 6t^9 + 12t^7 + t^5 + 4t^3 + 8t$
$p_{704} \equiv 11t^{12} + 2t^{10} + 9t^8 + 7t^6 + t^4 + 3t^2 + 3$
$p_{705} \equiv 11t^{11} + 12t^9 + 7t^7 + 3t^5 + 3t^3 + 7t$
$p_{706} \equiv t^{12} + 12t^{10} + 7t^6 + 11t^4 + 9t^2 + 1$
$p_{707} \equiv 8t^{11} + 3t^9 + 4t^7 + 2t^5 + 4t^3 + 11t$
$p_{708} \equiv 12t^{12} + 12t^{10} + 9t^8 + 3t^6 + 6t^4 + 7t^2 + 10$
$p_{709} \equiv 9t^9 + 2t^7 + t^3 + 10t$
$p_{710} \equiv 9t^{12} + t^{10} + 11t^8 + 6t^6 + 3t^4 + 5t^2 + 6$
$p_{711} \equiv 8t^{11} + t^9 + 9t^7 + 6t^5 + 8t^3 + 7t$
$p_{712} \equiv 7t^{12} + 5t^{10} + 11t^8 + 11t^6 + 3t^4 + 10t^2 + 9$
$p_{713} \equiv 10t^{11} + 4t^9 + 5t^7 + 10t^5 + 11t^3 + 7t$
$p_{714} \equiv 4t^{12} + 4t^{10} + 9t^8 + 7t^4 + 7t^2 + 3$
$p_{715} \equiv 12t^{11} + 12t^9 + 8t^7 + 3t^5 + 8t^3 + 8t$
$p_{716} \equiv 11t^{12} + 11t^{10} + 8t^8 + 4t^6 + 10t^4 + 3t^2 + 4$
$p_{717} \equiv 12t^{11} + t^9 + 11t^7 + 11t^5 + 9t^3 + 7t$
$p_{718} \equiv 2t^{12} + 11t^8 + 4t^6 + 2t^2 + 10$
$p_{719} \equiv 2t^{11} + 2t^9 + 5t^7 + 5t^5 + 9t$
$p_{720} \equiv 11t^{12} + t^{10} + 5t^8 + 3t^6 + 6t^4 + t^2 + 9$
$p_{721} \equiv 9t^{11} + t^9 + 11t^5 + 2t^3 + 6t$
$p_{722} \equiv 10t^{12} + 10t^{10} + 8t^8 + 7t^6 + 8t^4 + t^2 + 8$
$p_{723} \equiv 7t^{11} + 9t^9 + 11t^7 + 10t^5 + 10t^3 + 12t$
$p_{724} \equiv 7t^{12} + 7t^{10} + 12t^8 + 2t^6 + 9t^4 + 2t^2 + 12$
$p_{725} \equiv 6t^{11} + 9t^9 + 4t^7 + 11t^5 + 2t^3 + 4t$
$p_{726} \equiv 8t^{12} + 9t^{10} + 11t^8 + t^6 + t^4 + 12t^2 + 4$
$p_{727} \equiv 10t^9 + t^7 + 9t^5 + 8t^3$

$p_{728} \equiv 2t^{12} + 10t^{10} + 5t^8 + 5t^6 + 12t^4 + 2t^2 + 1$
$p_{729} \equiv 11t^{11} + 9t^9 + 10t^7 + 3t^5 + 11t^3 + 3t$
$p_{730} \equiv 2t^{12} + 8t^{10} + 7t^8 + 10t^6 + 5t^4 + t^2 + 9$
$p_{731} \equiv 4t^9 + 7t^7 + 5t^5 + 6t^3$
$p_{732} \equiv 7t^{12} + 3t^{10} + 9t^8 + 9t^6 + 4t^4 + 6t^2 + 12$
$p_{733} \equiv 6t^{11} + t^9 + 7t^7 + 3t^5 + 3t^3$
$p_{734} \equiv 3t^{12} + 7t^{10} + 3t^8 + 11t^6 + 7t^4 + 12t^2 + 2$
$p_{735} \equiv 9t^{11} + 5t^9 + 2t^7 + 6t^3 + 3t$
$p_{736} \equiv 8t^{12} + 3t^{10} + 8t^8 + 12t^6 + 6t^4 + 10t^2 + 3$
$p_{737} \equiv 11t^{11} + 6t^7 + 12t^5 + 6t^3 + 3t$
$p_{738} \equiv 9t^{12} + 8t^{10} + 10t^8 + 6t^6 + 9t^4 + 9t^2 + 1$
$p_{739} \equiv 9t^{11} + 5t^9 + 8t^3 + 10t$
$p_{740} \equiv 8t^{12} + t^{10} + t^8 + 4t^6 + 8t^4 + 11t^2 + 10$
$p_{741} \equiv 6t^{11} + 2t^9 + 4t^7 + 7t^5 + t^3 + 11t$
$p_{742} \equiv 12t^{12} + 12t^8 + 10t^6 + 4t^4 + 7t^2 + 12$
$p_{743} \equiv 7t^9 + 12t^7 + 12t^5 + 8t^3 + 6t$
$p_{744} \equiv 6t^{10} + 12t^8 + 2t^6 + 7t^4 + 4t^2 + 3$
$p_{745} \equiv 5t^{11} + 5t^9 + 9t^7 + 12t^5 + 5t^3 + 10t$
$p_{746} \equiv 9t^{12} + t^8 + 12t^6 + 3t^4 + 11t^2 + 7$
$p_{747} \equiv 5t^{11} + 3t^9 + 11t^7 + 5t^5 + 3t^3 + 6t$
$p_{748} \equiv 7t^{12} + 7t^{10} + 12t^8 + 8t^6 + 7t^4 + 3t^2 + 4$
$p_{749} \equiv t^{11} + 12t^9 + 3t^7 + 4t^5 + 10t^3 + t$
$p_{750} \equiv t^{12} + t^{10} + t^8 + 5t^6 + 3t^4 + 5t^2 + 10$
$p_{751} \equiv 12t^{11} + 8t^9 + t^7 + 5t^5 + t^3 + 6t$
$p_{752} \equiv 3t^{12} + 6t^{10} + 5t^8 + 3t^6 + 12t^4 + 11t^2 + 9$
$p_{753} \equiv 10t^9 + 5t^7 + 5t^5 + 8t^3 + 8t$
$p_{754} \equiv 4t^{12} + 6t^{10} + 8t^8 + 11t^6 + 8t^4 + 12t^2 + 3$
$p_{755} \equiv 6t^9 + 10t^7 + 12t^5 + 3t^3 + 2t$
$p_{756} \equiv 4t^{12} + 12t^{10} + 3t^8 + 4t^6 + 11t^4 + 7t^2 + 4$
$p_{757} \equiv 6t^{11} + t^7 + 9t^5 + 7t^3 + 8t$
$p_{758} \equiv 3t^{12} + 10t^{10} + 2t^8 + 2t^6 + 9t^4 + 4t^2 + 5$
$p_{759} \equiv 6t^{11} + 2t^9 + 6t^7 + 9t^5 + 8t^3 + 9t$
$p_{760} \equiv 11t^{12} + 12t^{10} + 8t^8 + 11t^6 + 5t^4 + 3t^2 + 1$
$p_{761} \equiv 3t^{11} + 9t^9 + 7t^7 + 12t^5 + 3t^3 + 10t$
$p_{762} \equiv 8t^{12} + 11t^{10} + 9t^8 + 6t^6 + 5t^2 + 9$
$p_{763} \equiv 9t^{11} + 11t^9 + 8t^7 + t^5 + 10t^3 + t$
$p_{764} \equiv 2t^{12} + 4t^{10} + t^8 + t^6 + 6t^4 + t^2 + 12$
$p_{765} \equiv 2t^{11} + t^9 + 5t^7 + 5t^5 + 11t^3 + 5t$
$p_{766} \equiv 9t^{12} + 12t^{10} + 4t^8 + 9t^6 + 3t^4 + 5t^2 + 4$
$p_{767} \equiv 11t^{11} + t^9 + t^7 + 2t^5 + 9t^3 + 2t$
$p_{768} \equiv 9t^{12} + 5t^{10} + 2t^8 + 7t^6 + 7t^4 + 8t^2 + 1$
$p_{769} \equiv 9t^{11} + 3t^9 + 7t^7 + 2t^5 + 7t^3 + 9t$
$p_{770} \equiv 8t^{12} + t^{10} + 7t^8 + 7t^6 + 6t^4 + 5t^2 + 11$
$p_{771} \equiv 11t^{11} + 11t^9 + 3t^7 + 12t^5 + t$
$p_{772} \equiv 4t^{10} + t^8 + 3t^6 + 8t^4 + 11t^2 + 10$
$p_{773} \equiv 11t^9 + t^7 + t^5 + 10t^3 + 10t$
$p_{774} \equiv 11t^8 + 9t^6 + 4t^4 + 12$
$p_{775} \equiv 8t^7 + 10t^5 + 9t^3 + t$
$p_{776} \equiv 6t^6 + 3t^4 + 11t^2 + 3$
$p_{777} \equiv 7t^5 + 8t^3 + t$
$p_{778} \equiv 10t^4 + 3t^2 + 1$
$p_{779} \equiv 7t^3$
$p_{780} \equiv 8t^2 + 10$
$p_{781} \equiv 5t$
$p_{782} \equiv 7t^2 + 6$
$p_{783} \equiv 11t^3 + 2t$
$p_{784} \equiv 4t^2 + 9$
$p_{785} \equiv 10t^3 + 3t$
$p_{786} \equiv 12t^4 + 11t^2 + 3$
$p_{787} \equiv 3t^3 + 10t$
$p_{788} \equiv 11t^4 + 11t^2 + 4$
$p_{789} \equiv 8t^5 + t^3 + 4t$
$p_{790} \equiv 4t^4 + 12t^2 + 10$
$p_{791} \equiv 4t^5 + 9t$
$p_{792} \equiv 10t^6 + 4t^4 + 3t^2 + 9$
$p_{793} \equiv 9t^5 + t^3 + 3t$
$p_{794} \equiv 6t^6 + 9t^4 + 3t^2 + 8$
$p_{795} \equiv 2t^7 + 11t$
$p_{796} \equiv 7t^6 + 7t^2 + 12$
$p_{797} \equiv 3t^7 + 8t^5 + 10t^3 + 5t$
$p_{798} \equiv t^8 + 2t^6 + 6t^2 + 4$
$p_{799} \equiv 3t^5 + 11t^3 + 12t$
$p_{800} \equiv 2t^8 + 9t^6 + t^2 + 1$
$p_{801} \equiv 5t^9 + 7t^7 + 4t^3 + 10t$
$p_{802} \equiv 5t^8 + 2t^6 + 12t^4 + 11t^2 + 9$
$p_{803} \equiv 9t^9 + 7t^7 + 12t^5 + 3t^3 + 8t$
$p_{804} \equiv 3t^{10} + 3t^8 + 3t^6 + 3t^4 + 2t^2 + 12$
$p_{805} \equiv 11t^9 + t^7 + 3t^3 + 11t$
$p_{806} \equiv 7t^{10} + t^8 + t^6 + 2t^4 + 2$
$p_{807} \equiv 11t^{11} + 6t^9 + 2t^7 + t^5 + 4t^3 + 2t$
$p_{808} \equiv 2t^{10} + 6t^8 + 11t^6 + t^4 + 3t^2 + 3$
$p_{809} \equiv 10t^{11} + 7t^9 + 10t^7 + 2t^5 + 3t^3 + 7t$
$p_{810} \equiv 12t^{12} + 2t^{10} + 2t^4 + 9t^2 + 1$
$p_{811} \equiv 7t^{11} + 6t^9 + 11t^7 + 5t^5 + 12t^3 + 11t$
$p_{812} \equiv 11t^{12} + 12t^{10} + 7t^6 + 6t^4 + 6t^2 + 10$
$p_{813} \equiv 5t^{11} + 7t^9 + 2t^7 + 5t^5 + 3t^3 + 9t$
$p_{814} \equiv 10t^{12} + 5t^{10} + 5t^8 + 3t^6 + 6t^4 + 11t^2 + 12$
$p_{815} \equiv 6t^{11} + 10t^9 + 5t^5 + 4t^3 + 10t$
$p_{816} \equiv 2t^{12} + 10t^{10} + 3t^8 + 6t^6 + 9t^4 + 9t^2 + 3$
$p_{817} \equiv 7t^{11} + 10t^7 + t^5 + 12t^3 + 10t$
$p_{818} \equiv 4t^{10} + 4t^8 + 9t^6 + 5t^4 + 4t^2 + 7$
$p_{819} \equiv 10t^{11} + 3t^9 + 12t^7 + 3t^5 + 8t^3 + 8t$
$p_{820} \equiv 7t^{12} + 12t^{10} + 3t^8 + 5t^6 + 10t^4 + 3t^2 + 4$
$p_{821} \equiv 6t^{11} + 9t^9 + 4t^7 + t^5 + 9t^3 + 7t$

$p_{822} \equiv t^{12} + 11t^{10} + 8t^8 + 2t^6 + t^4 + 2t^2 + 10$
$p_{823} \equiv 8t^{11} + 8t^9 + 4t^7 + 9t^5 + 2t^3 + 9t$
$p_{824} \equiv 9t^{12} + 7t^{10} + 8t^8 + 10t^6 + 4t^4 + 4t^2 + 9$
$p_{825} \equiv 5t^{11} + 6t^9 + 8t^7 + 7t^5 + 3t^3 + 9t$
$p_{826} \equiv 7t^{12} + 7t^{10} + 5t^8 + 12t^6 + t^4 + 3$
$p_{827} \equiv t^{11} + 3t^7 + 10t^5 + t^3 + 10t$
$p_{828} \equiv t^{12} + t^{10} + 10t^8 + 7t^6 + 6t^4 + 9t^2 + 4$
$p_{829} \equiv 12t^{11} + 3t^9 + 10t^7 + 5t^5 + 8t^3$
$p_{830} \equiv 3t^{12} + 9t^8 + 11t^6 + 12t^4 + 7t^2 + 5$
$p_{831} \equiv 11t^9 + 4t^7 + 3t^5 + t^3 + 6t$
$p_{832} \equiv 4t^{12} + 3t^{10} + t^8 + 2t^6 + 12t^4 + 2t^2 + 1$
$p_{833} \equiv 12t^{11} + 12t^9 + 7t^5 + 11t^3 + 3t$
$p_{834} \equiv 4t^{12} + 3t^{10} + 3t^8 + 10t^6 + 2t^4 + t^2 + 9$
$p_{835} \equiv 7t^9 + 4t^7 + 7t^5$
$p_{836} \equiv t^{12} + 9t^{10} + 8t^8 + 3t^6 + 5t^4 + 10t^2 + 12$
$p_{837} \equiv 2t^{11} + 5t^9 + 9t^7 + 12t^5 + 8t^3 + 4t$
$p_{838} \equiv 8t^{12} + 10t^{10} + 11t^8 + 11t^6 + t^4 + 6t^2 + 4$
$p_{839} \equiv 9t^{11} + 12t^9 + 12t^7 + 5t^5 + 11t^3 + 9t$
$p_{840} \equiv 2t^{12} + 3t^{10} + 4t^8 + 2t^6 + 10t^2 + 1$
$p_{841} \equiv t^{11} + 2t^7 + t^3 + 5t$
$p_{842} \equiv t^{12} + 5t^{10} + 6t^8 + 11t^6 + t^4 + 12t^2 + 11$
$p_{843} \equiv t^{11} + 4t^9 + 12t^7 + 12t^5 + 7t^3 + t$
$p_{844} \equiv 11t^{12} + t^8 + 10t^6 + t^4 + 10t^2 + 10$
$p_{845} \equiv 11t^{11} + 7t^7 + 8t^5 + t^3 + 11t$
$p_{846} \equiv 9t^{12} + 2t^{10} + 2t^8 + 6t^6 + 7t^2 + 12$
$p_{847} \equiv t^{11} + 4t^7 + 6t$
$p_{848} \equiv 11t^{12} + 12t^{10} + t^6 + 5t^2 + 3$
$p_{849} \equiv 8t^{11} + 8t^9 + 6t^7 + 3t^5 + 9t^3 + 11t$
$p_{850} \equiv 7t^{12} + 12t^{10} + 12t^8 + 4t^6 + 9t^4 + 8t^2 + 1$
$p_{851} \equiv t^{11} + 5t^9 + t^7 + 11t^5 + 11t^3 + 3t$
$p_{852} \equiv 4t^{12} + 4t^{10} + 6t^8 + 6t^6 + 12t^4 + 11t^2 + 10$
$p_{853} \equiv 8t^{11} + 3t^9 + 10t^7 + 12t^5 + 5t^3 + 10t$
$p_{854} \equiv 8t^{12} + 8t^{10} + 2t^8 + 5t^6 + 2t^4 + 7t^2 + 6$
$p_{855} \equiv 5t^{11} + 4t^9 + 6t^7 + t^5 + 5t^3 + 3t$
$p_{856} \equiv 11t^{12} + 2t^{10} + 9t^8 + 9t^6 + 9$
$p_{857} \equiv 6t^7 + 6t^5 + 12t^3 + 8t$
$p_{858} \equiv 6t^{12} + 7t^{10} + 9t^8 + 10t^6 + 7t^4 + 12t^2 + 3$
$p_{859} \equiv 8t^{11} + 8t^9 + 3t^7 + 10t^5 + t^3 + 2t$
$p_{860} \equiv 6t^{12} + 7t^{10} + 12t^8 + 5t^6 + 10t^4 + 4t^2 + 4$
$p_{861} \equiv 6t^{11} + 3t^9 + 5t^7 + 4t^5 + 4t^3 + 5t$
$p_{862} \equiv 10t^{12} + 3t^{10} + 5t^6 + 6t^4 + 12t^2 + 10$
$p_{863} \equiv 2t^{11} + 4t^9 + t^7 + 7t^5 + t^3 + 6t$
$p_{864} \equiv 3t^{12} + 3t^{10} + 12t^8 + 4t^6 + 8t^4 + 5t^2 + 9$
$p_{865} \equiv 12t^9 + 7t^7 + 10t^3 + 9t$
$p_{866} \equiv 12t^{12} + 10t^{10} + 6t^8 + 8t^6 + 4t^4 + 11t^2 + 8$
$p_{867} \equiv 2t^{11} + 10t^9 + 12t^7 + 8t^5 + 2t^3 + 5t$
$p_{868} \equiv 5t^{12} + 11t^{10} + 6t^8 + 6t^6 + 4t^4 + 9t^2 + 12$
$p_{869} \equiv 9t^{11} + t^9 + 11t^7 + 9t^5 + 6t^3 + 5t$
$p_{870} \equiv t^{12} + t^{10} + 12t^8 + 5t^4 + 5t^2 + 4$
$p_{871} \equiv 3t^{11} + 3t^9 + 2t^7 + 4t^5 + 2t^3 + 2t$
$p_{872} \equiv 6t^{12} + 6t^{10} + 2t^8 + t^6 + 9t^4 + 4t^2 + 1$
$p_{873} \equiv 3t^{11} + 10t^9 + 6t^7 + 6t^5 + 12t^3 + 5t$
$p_{874} \equiv 7t^{12} + 6t^8 + t^6 + 7t^2 + 9$
$p_{875} \equiv 7t^{11} + 7t^9 + 11t^7 + 11t^5 + 12t$
$p_{876} \equiv 6t^{12} + 10t^{10} + 11t^8 + 4t^6 + 8t^4 + 10t^2 + 12$
$p_{877} \equiv 12t^{11} + 10t^9 + 6t^5 + 7t^3 + 8t$
$p_{878} \equiv 9t^{12} + 9t^{10} + 2t^8 + 5t^6 + 2t^4 + 10t^2 + 2$
$p_{879} \equiv 5t^{11} + 12t^9 + 6t^7 + 9t^5 + 9t^3 + 3t$
$p_{880} \equiv 5t^{12} + 5t^{10} + 3t^8 + 7t^6 + 12t^4 + 7t^2 + 3$
$p_{881} \equiv 8t^{11} + 12t^9 + t^7 + 6t^5 + 7t^3 + t$
$p_{882} \equiv 2t^{12} + 12t^{10} + 6t^8 + 10t^6 + 10t^4 + 3t^2 + 1$
$p_{883} \equiv 9t^9 + 10t^7 + 12t^5 + 2t^3$
$p_{884} \equiv 7t^{12} + 9t^{10} + 11t^8 + 11t^6 + 3t^4 + 7t^2 + 10$
$p_{885} \equiv 6t^{11} + 12t^9 + 9t^7 + 4t^5 + 6t^3 + 4t$
$p_{886} \equiv 7t^{12} + 2t^{10} + 5t^8 + 9t^6 + 11t^4 + 10t^2 + 12$
$p_{887} \equiv t^9 + 5t^7 + 11t^5 + 8t^3$
$p_{888} \equiv 5t^{12} + 4t^{10} + 12t^8 + 12t^6 + t^4 + 8t^2 + 3$
$p_{889} \equiv 8t^{11} + 10t^9 + 5t^7 + 4t^5 + 4t^3$
$p_{890} \equiv 4t^{12} + 5t^{10} + 4t^8 + 6t^6 + 5t^4 + 3t^2 + 7$
$p_{891} \equiv 12t^{11} + 11t^9 + 7t^7 + 8t^3 + 4t$
$p_{892} \equiv 2t^{12} + 4t^{10} + 2t^8 + 3t^6 + 8t^4 + 9t^2 + 4$
$p_{893} \equiv 6t^{11} + 8t^7 + 3t^5 + 8t^3 + 4t$
$p_{894} \equiv 12t^{12} + 2t^{10} + 9t^8 + 8t^6 + 12t^4 + 12t^2 + 10$
$p_{895} \equiv 12t^{11} + 11t^9 + 2t^3 + 9t$
$p_{896} \equiv 2t^{12} + 10t^{10} + 10t^8 + t^6 + 2t^4 + 6t^2 + 9$
$p_{897} \equiv 8t^{11} + 7t^9 + t^7 + 5t^5 + 10t^3 + 6t$
$p_{898} \equiv 3t^{12} + 3t^8 + 9t^6 + t^4 + 5t^2 + 3$
$p_{899} \equiv 5t^9 + 3t^7 + 3t^5 + 2t^3 + 8t$
$p_{900} \equiv 8t^{10} + 3t^8 + 7t^6 + 5t^4 + t^2 + 4$
$p_{901} \equiv 11t^{11} + 11t^9 + 12t^7 + 3t^5 + 11t^3 + 9t$
$p_{902} \equiv 12t^{12} + 10t^8 + 3t^6 + 4t^4 + 6t^2 + 5$
$p_{903} \equiv 11t^{11} + 4t^9 + 6t^7 + 11t^5 + 4t^3 + 8t$
$p_{904} \equiv 5t^{12} + 5t^{10} + 3t^8 + 2t^6 + 5t^4 + 4t^2 + 1$
$p_{905} \equiv 10t^{11} + 3t^9 + 4t^7 + t^5 + 9t^3 + 10t$
$p_{906} \equiv 10t^{12} + 10t^{10} + 10t^8 + 11t^6 + 4t^4 + 11t^2 + 9$
$p_{907} \equiv 3t^{11} + 2t^9 + 10t^7 + 11t^5 + 10t^3 + 8t$
$p_{908} \equiv 4t^{12} + 8t^{10} + 11t^8 + 4t^6 + 3t^4 + 6t^2 + 12$
$p_{909} \equiv 9t^9 + 11t^7 + 11t^5 + 2t^3 + 2t$
$p_{910} \equiv t^{12} + 8t^{10} + 2t^8 + 6t^6 + 2t^4 + 3t^2 + 4$
$p_{911} \equiv 8t^9 + 9t^7 + 3t^5 + 4t^3 + 7t$
$p_{912} \equiv t^{12} + 3t^{10} + 4t^8 + t^6 + 6t^4 + 5t^2 + 1$
$p_{913} \equiv 8t^{11} + 10t^7 + 12t^5 + 5t^3 + 2t$
$p_{914} \equiv 4t^{12} + 9t^{10} + 7t^8 + 7t^6 + 12t^4 + t^2 + 11$
$p_{915} \equiv 8t^{11} + 7t^9 + 8t^7 + 12t^5 + 2t^3 + 12t$

$p_{916} \equiv 6t^{12} + 3t^{10} + 2t^8 + 6t^6 + 11t^4 + 4t^2 + 10$
$p_{917} \equiv 4t^{11} + 12t^9 + 5t^7 + 3t^5 + 4t^3 + 9t$
$p_{918} \equiv 2t^{12} + 6t^{10} + 12t^8 + 8t^6 + 11t^2 + 12$
$p_{919} \equiv 12t^{11} + 6t^9 + 2t^7 + 10t^5 + 9t^3 + 10t$
$p_{920} \equiv 7t^{12} + t^{10} + 10t^8 + 10t^6 + 8t^4 + 10t^2 + 3$
$p_{921} \equiv 7t^{11} + 10t^9 + 11t^7 + 11t^5 + 6t^3 + 11t$
$p_{922} \equiv 12t^{12} + 3t^{10} + t^8 + 12t^6 + 4t^4 + 11t^2 + 1$
$p_{923} \equiv 6t^{11} + 10t^9 + 10t^7 + 7t^5 + 12t^3 + 7t$
$p_{924} \equiv 12t^{12} + 11t^{10} + 7t^8 + 5t^6 + 5t^4 + 2t^2 + 10$
$p_{925} \equiv 12t^{11} + 4t^9 + 5t^7 + 7t^5 + 5t^3 + 12t$
$p_{926} \equiv 2t^{12} + 10t^{10} + 5t^8 + 5t^6 + 8t^4 + 11t^2 + 6$
$p_{927} \equiv 6t^{11} + 6t^9 + 4t^7 + 3t^5 + 10t$
$p_{928} \equiv t^{10} + 10t^8 + 4t^6 + 2t^4 + 6t^2 + 9$
$p_{929} \equiv 6t^9 + 10t^7 + 10t^5 + 9t^3 + 9t$
$p_{930} \equiv 6t^8 + 12t^6 + t^4 + 3$
$p_{931} \equiv 2t^7 + 9t^5 + 12t^3 + 10t$
$p_{932} \equiv 8t^6 + 4t^4 + 6t^2 + 4$
$p_{933} \equiv 5t^5 + 2t^3 + 10t$
$p_{934} \equiv 9t^4 + 4t^2 + 10$
$p_{935} \equiv 5t^3$
$p_{936} \equiv 2t^2 + 9$
$p_{937} \equiv 11t$

References

[OR12] Cormac O'Sullivan and Morten S. Risager, *Non-vanishing of Taylor coefficients and Poincaré series* (2012). ↑1, 16, 21, 29, 30, 32, 33

[BGHZ04] Jan Hendrik Bruinier, Gerard van der Geer, Günter Harder, and Don Zaiger, *The 1-2-3 of Modular Forms*, 2004. ↑16, 20, 29

[IO08] Özlem Imamoḡlu and Cormac O'Sullivan, *Parabolic, hyperbolic and elliptic Poincaré series* (2008). ↑19

[Cox13] David A. Cox, *Primes of the form $x^2 + ny^2$: Fermat, class field theory, and complex multiplication*, Vol. 34, John Wiley & Sons, 2013. ↑9, 12, 14, 15

[Mil06] James S. Milne, *Elliptic Curves*, BookSurge Publishers, 2006. ↑2, 3, 4

[Mil08] ———, *Abelian Varieties (v2.00)*, 2008. Available at jmilne.org/math. ↑2

[Mil17] ———, *Algebraic Geometry (v6.02)*, 2017. Available at jmilne.org/math. ↑2, 3

[Sil09] Joseph H. Silverman, *The arithmetic of elliptic curves*, Vol. 106, Springer, 2009. ↑2, 6, 8, 11

[ST15] Joseph H. Silverman and John T. Tate, *Rational points on elliptic curves*, Springer, 2015. ↑2

[Gal18] Steven D. Galbraith, *Mathematics of Public Key Cryptography. Version 2.0*, Cambridge University Press, 2018. Available at math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html. ↑2

[Sut21] Andrew Sutherland, *Elliptic Curves*, 2021. Available at math.mit.edu/classes/18.783/2017/lectures. ↑6, 10

[Vak23] Ravi Vakil, *The Rising Sea: Foundations of Algebraic Geometry*, 2023. Available at math.stanford.edu/~vakil/216blog/FOAGjul3123public.pdf. ↑2

[KK07] Max Koecher and Aloys Krieg, *Elliptische Funktionen und Modulformen*, 2nd ed., Springer, 2007. ↑5, 6, 8

[Zag81] Don B. Zagier, *Zetafunktionen und Quadratische Körper: Eine Einführung in die höhere Zahlentheorie*, Springer, 1981. ↑15

[Kat92] Svetlana Katok, *Fuchsian Groups*, The University of Chicago Press, 1992. ↑17

[DeF17] Luca DeFeo, *Mathematics of Isogeny Based Cryptography*, arXiv, 2017. Available at arxiv.org/abs/1711.04062. ↑13

[ZV] Don Zagier and Fernando Rodriguez Villegas, *Square roots of central values of Hecke L-series*. Available at people.mpim-bonn.mpg.de/zagier/files/mpim/92-68/fulltext.pdf. ↑20