Computing isolated coefficients of the j-function

Fredrik Johansson

COMPUTING ISOLATED COEFFICIENTS OF THE j-FUNCTION

FREDRIK JOHANSSON

Abstract. We consider the problem of efficiently computing isolated coefficients c_n in the Fourier series of the elliptic modular function $j(\tau)$. We show that a hybrid numerical-modular method with complexity $n^{1+o(1)}$ is efficient in practice. As an application, we locate the first few values of c_n that are prime, the first occurring at $n = 457871$.

1. Introduction

The coefficients c_n appearing in the Fourier series of the elliptic modular function $j(\tau) = \sum_{n=-1}^{\infty} c_n e^{2\pi i n \tau}$ have many remarkable arithmetical properties. This integer sequence (A000521 in Sloane's OEIS [7]) begins

$$c_{-1} = 1, \quad c_0 = 744, \quad c_1 = 196884, \quad c_2 = 21493760, \quad \ldots$$

and is perhaps most notorious for its connection with the monster group (a correspondence known as *monstrous moonshine* [4]).

The numerical growth rate of these coefficients is known: c_n has around $\beta\sqrt{n}$ digits where $\beta = 4\pi/\log(10) \approx 5.46$. More precisely, explicit lower and upper bounds for $n \geq 1$ are given by [3, Theorem 1.1]

$$c_n = \frac{e^{4\pi\sqrt{n}}}{\sqrt{2}n^{3/4}}\left(1 - \frac{3}{32\pi\sqrt{n}} + \varepsilon_n\right), \quad |\varepsilon_n| \leq \frac{0.055}{n}.$$

Our concern in this work will be the exact calculation of c_n for large n. Baier and Köhler [2] survey several strategies, concluding that a formula by Zagier and Kaneko is the "most efficient method". However, their analysis only considers recursive calculation which does not yield the best possible complexity. Using fast power series arithmetic, applicable to many of the formulas discussed by Baier and Köhler, it is possible to compute $c_{-1}, \ldots, c_n$ simultaneously in $n^{1.5+o(1)}$ time (bit operations), or in $n^{1+o(1)}$ time modulo a fixed number M. These complexities are essentially optimal since they are quasilinear in the size of the output.

A more challenging problem is to compute an isolated value c_n quickly. It is presently unknown whether an algorithm with quasioptimal $n^{0.5+o(1)}$ time complexity exists, but it is possible to achieve $n^{1+o(1)}$ time and $n^{0.5+o(1)}$ space complexity using numerical evaluation of an infinite series for c_n derived by Petersson [13] and Rademacher [14], improving on the $n^{1.5+o(1)}$ time and $n^{1+o(1)}$ space needed with the power series method. Unfortunately, Rademacher, Baier and Köhler and several authors have dismissed this method as impractical for computations. We analyze this method below and find that it is useful, but that current error bounds are too pessimistic to make it practical if we insist on rigorous results.

As a solution to this problem, we propose a rigorous hybrid algorithm that uses the Petersson-Rademacher series to obtain the high bits of c_n together with a power series calculation or a suitable congruence to obtain the low bits. This

method requires $n^{1+o(1)}$ time and space, or $n^{0.5+o(1)}$ space for n of special form, and is highly efficient in practice.

This work was prompted by a MathOverflow post by David Feldman which asked whether the sequence c_n contains prime numbers [5]. We can answer this in the affirmative: an exhaustive search of $n \le 2 \cdot 10^7$ finds seven prime values of c_n in this range (Table 2), the first occurring at $n = 457871$.

2. The power series method

A natural way to compute c_n is to expand $j(\tau) = \sum_{k=-1}^{\infty} c_k q^k$ as a truncated formal power series in q and read off the last coefficient. This method obviously gives all the coefficients up to c_n if we so prefer.

There are many ways to express $j(\tau)$ in terms of simple functions suitable for power series computations. The following formula is economical and works in any characteristic: define the Eisenstein series $E_4 = 1 + 240 \sum_{n=1}^{\infty} \sigma_3(n) q^n$ where $\sigma_x(n) = \sum_{d|n} d^x$ is the divisor function, and denote by $\phi = \sum_{n \in \mathbb{Z}} (-1)^n q^{n(3n-1)/2}$ the Dedekind eta function without leading factor $q^{1/24}$. Then

$$j(\tau) = \frac{1}{q} \left(\frac{E_4}{\phi^8} \right)^3. \tag{1}$$

Noting that $\phi^4 = \phi^3 \cdot \phi$ can be generated cheaply as a sparse product, the cost of evaluating (1) is essentially 2 multiplications, 2 squarings and 1 inversion of power series, plus the construction of E_4 which can be done in similar time to a multiplication using the sieve of Eratosthenes (alternatively, E_4 can be constructed from Jacobi theta functions using further multiplications). For a small speedup, we note that the final multiplication $(E_4/\phi^8)^3 = (E_4/\phi^8)^2 \cdot (E_4/\phi^8)$ can be replaced by an linear summation if we only want the last coefficient.

The computations can be done in $n^{1.5+o(1)}$ time using FFT multiplication together with Newton inversion, since we have power series of length n with coefficients $n^{0.5+o(1)}$ bits in size. The memory requirement is $n^{1.5+o(1)}$ bits if we work directly over $\mathbb{Z}$. Computing modulo a b-bit integer M, the complexity for determining c_n mod M reduces to $(nb)^{1+o(1)}$. The memory requirement for computing the single integer c_n can thus be reduced to $n^{1+o(1)}$ if we compute separately modulo small pairwise coprime integers m_k with $M = m_1, \ldots, m_k$, $c_n < M$, and then reconstruct c_n using the Chinese remainder theorem.

3. The Petersson-Rademacher series

Given any $n \ge 1$, the Petersson-Rademacher series for c_n is the convergent series (letting $N \to \infty$)

$$c_n = \frac{2\pi}{\sqrt{n}} \sum_{k=1}^{N} \frac{S(n,-1,k)}{k} I_1\left(\frac{4\pi\sqrt{n}}{k} \right) + R_N(n) \tag{2}$$

in which $S(a,b;k)$ denotes a Kloosterman sum, $I_\nu(x)$ denotes a modified Bessel function of the first kind, and the remainder term can be shown to satisfy

$$|R_N(n)| \le \frac{72\pi}{\sqrt{n}} N^{3/4} I_1\left(\frac{4\pi\sqrt{n}}{N} \right). \tag{3}$$

For a proof of the truncation bound as well as a generalization to the corresponding coefficients for the function $j(\tau)^m$, see Brisebarre and Philibert [3].

The Kloosterman sum is the exponential sum

$$S(a,b;k) = \sum_{\gcd(x,k)=1} e^{2\pi i(ax+by)/k}$$

where the index x ranges over $0 \le x < k$ and y is any solution of $xy \equiv 1 \bmod k$.

The Petersson-Rademacher series for c_n is analogous to the Hardy-Ramanujan-Rademacher formula for the integer partition function $p(n)$, which allows computing $p(n)$ in quasioptimal time $n^{0.5+o(1)}$ [8, 9]. The idea is that adding $\Theta(n^{0.5})$ terms of the series gives an approximation y with $|y - p(n)| < 0.5$, yielding the correct result when rounded to the nearest integer. Although there are $\Theta(n^{0.5})$ terms and the result has $\Theta(n^{0.5})$ bits, the overall complexity is $n^{0.5+o(1)}$ rather than $n^{1+o(1)}$ when the algorithm is implemented carefully since the bit sizes of the terms fall off as a hyperbola after the first term.

By a similar analysis of the Petersson-Rademacher series for c_n, one can show the following:

Theorem 3.1. *The integer c_n can be computed using $n^{1+o(1)}$ bit operations and $n^{0.5+o(1)}$ bits of space.*

The reason why we do not get an $n^{0.5+o(1)}$ algorithm is that computing the Kloosterman sum $S(a,b;k)$ ostensibly requires adding $O(k)$ terms. The computation of $S(a,b;k)$ can be reduced to the computation of the shorter sums $S(a_i,b_i;q_i)$ for each prime power $q_i = p_i^{e_i}$ in the factorization of k, and closed formulas are known when $e_i \ge 2$ [18], but for prime modulus $q_i = p$ no algorithm better than $O(p)$ summation is currently known (the existence of such an algorithm would immediately lead to a better complexity bound for computing c_n). The corresponding exponential sums in the series for $p(n)$ admit a complete factorization into simple trigonometric expressions and can therefore be computed rapidly.

3.1. **Analysis of the error bound.** Although the Petersson-Rademacher series regrettably does not yield an $n^{0.5+o(1)}$ algorithm with current technology, the problem with the method is not the asymptotic $n^{1+o(1)}$ complexity (which is quite serviceable), but the hidden constant factors.

To make calculations explicit, we may combine (3) with the following bounds for the Bessel function $I_1(x)$, accurate when $x \to \infty$ and $x \to 0$ respectively (obtained from the asymptotic expansion and the Taylor series at $x = 0$).

Lemma 3.2. For $x > 0$, $I_1(x) < e^x/\sqrt{2\pi x}$.

Lemma 3.3. For $0 < x < 0.1$, $I_1(x) < 0.501x$.

A direct calculation using Lemma 3.3 gives, for instance:

Theorem 3.4. *If $N \ge \max(C_0, C_1\sqrt{n})$ where $C_0 = 6.7 \cdot 10^{13}$ and $C_1 = 40\pi$, then $|R_N(n)| \le 0.499$.*

This shows that $\Theta(n^{0.5})$ terms are sufficient to determine c_n (as needed in the proof of Theorem 3.1), noting that bounding the truncation error by 0.499 gives some wiggle room for floating-point error in the approximation of the sum.

The constant C_0 makes the method virtually useless, as we would have to perform some $C_0^2 \approx 10^{26}$ operations to compute any c_n. The constants C_0 and C_1 in this theorem are not optimal, but C_0 cannot be brought below $16 \cdot (12\pi)^8 \approx 6.53 \cdot 10^{13}$ using the bound (3) (without a corresponding pessimistic increase of C_1). Rademacher, working with a somewhat worse error bound than (3), similarly concluded:

> "Unfortunately, the convergence of [...] is rather slow, so that we should need quite a number of terms in order to get an error which is safely less than 1/2."

How slow is the convergence really? We can run some experiments to get an idea. Figure 1 compares the actual rate of convergence of the Petersson-Rademacher series with the bound (3). It turns out that the bound is quite pessimistic, and if we cut off the summation heuristically, the computation becomes practical. However, simply stopping when the partial sum seems to be very close to an integer is dangerous. The erratic nature of the terms (clearly visible in the figure, and observed in similar contexts by other authors, e.g. [11]) means that the sum can stabilize within ε of an integer for rather small ε and remain there through many consecutive terms before suddenly making a large jump. We could probably find a good empirical fit for the true error, but we prefer a rigorous analysis.

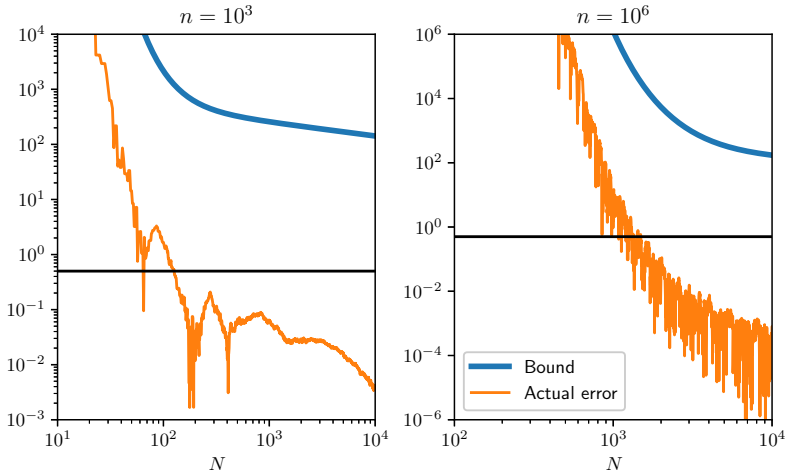

FIGURE 1. The actual error $R_N(n)$ in the approximation of c_n after summing N terms of the Petersson-Rademacher series (2), compared with the bound (3). The horizontal line locates 1/2.

This brings us to the question of how (3) can be improved. The ingredients in the derivation of this bound are [3, Section 5.1]:

- The Weil bound $|S(a, b; k)| \le \sqrt{\sigma_0(k)}\sqrt{\gcd(a, b, k)}\sqrt{k}$.
- The bound $\sigma_0(k) \le 9k^{1/4}$ for the number of divisors of k.
- Bounding the sum of terms in (2) by a sum of absolute values of the terms.

The bound $\sigma_0(k) \le 9k^{1/4}$ is not optimal. However, this bound is quite reasonable over the relevant non-asymptotic range of k. If we could estimate $\sigma_0(k)$ by its *average* value $\log(k)$, and show that the deviations from the average give a negligible contribution, we would get a constant C_0 around $3 \cdot 10^7$, which is much better but still rather impractical for computations.

It therefore seems that a useful error bound will require a much more involved analysis of cancellation in sums of Kloosterman sums. This is a well-studied problem [15], but it appears that all published results stronger than the Weil bound are asymptotic without explicit constants. We leave it as an open problem to prove sharp bounds for $R_N(n)$.

4. Hybrid algorithm

To get around the issues discussed in the previous section without sacrificing rigor, we make the following modification: instead of computing c_n directly, we assume that c_n is known modulo some integer M. If we denote by r the unique residue of c_n with $0 \le r < M$, we have $c_n = M[(c_n - r)/M] + r$. We can now compute $(c_n - r)/M$ using (2), stopping the summation when the bound for $|R_N(n)|$ is just less than $M/2$ instead of $1/2$. If we choose M large enough, the truncation error is in the exponentially-large domain of the Bessel function (Lemma 3.2) where the available error bound works well.

For a general index n, we can compute the residue of c_n in $\mathbb{Z}/M\mathbb{Z}$ using the power series (1). We can either compute a single power series over $\mathbb{Z}/M\mathbb{Z}$ or choose a composite modulus, say $M = p_1 p_2 \cdots p_k$, and compute the value for each $\mathbb{Z}/p_i\mathbb{Z}$ in order to minimize memory usage. If M is bounded, the memory usage will be $n^{1+o(1)}$, and the running time will be $n^{1+o(1)}$. This is not as good as the Petersson-Rademacher series would be if we had an optimal error bound, but it is better than computing c_n using power series alone.

The optimal choice of M will depend on the implementation-dependent relative speeds of the power series arithmetic and the numerical calculations for the Petersson-Rademacher series. In our experiments, we have obtained the best performance with M slightly larger than 2^{10} for computing isolated values of c_n. We get relatively uniform performance with M anywhere between 2^{10} and 2^{20} while the running time increases sharply if M is smaller than 2^9. If we want to compute a range of values of c_n, it is more efficient to choose a larger M, say word-size $M \approx 2^{64}$ (or bigger, if sufficient memory is available), precomputing the residues $c_{-1}, \ldots, c_n$ once before using the Petersson-Rademacher series to determine the high bits for each coefficient of interest.

4.1. Special indices.

When n has special form, we can use known congruences for c_n to determine a residue. For $a > 1$, we have

$$\begin{aligned}
c_{2^a m} &\equiv -2^{3a+8}3^{a-1}\sigma_7(m) \pmod{2^{3a+13}}, && m \text{ odd} \\
c_{3^a m} &\equiv \mp 3^{2a+3}10^{a-1}\sigma(m)/m \pmod{3^{2a+6}}, && m \equiv \pm 1 \pmod 3 \\
c_{5^a m} &\equiv -5^{a+1}3^{a-1}m\sigma(m) \pmod{5^{a+2}} \\
c_{7^a m} &\equiv -7^a 5^{a-1} m\sigma_3(m) \pmod{7^{a+1}}
\end{aligned}$$

and there are also congruences modulo 11 and 13 [1, 12]. If n is divisible by sufficiently many small primes, we can thus immediately construct a residue modulo an M that is large enough (say $M > 1000$) to use the Petersson-Rademacher series. In that case, we do not need a power series evaluation, and the entire computation can be done using $n^{0.5+o(1)}$ memory. Indeed, it is sufficient that n is even, since this gives the comfortably large $M \ge 65536$. If the M obtained from congruences alone is too small, we can amend it with a power series computation modulo some larger prime.

5. Computations

We have implemented evaluation of c_n using direct power series methods as well as the hybrid algorithm, using Flint [6] for power series and integer operations and Arb [10] for arbitrary-precision ball arithmetic. The following computations were performed on a laptop with 16 GB RAM and an Intel i5-4300U CPU.

TABLE 1. Time in seconds to compute c_n or $c_{n'}$ where n' = next prime after n. PS denotes the power series method using arithmetic directly in $\mathbb{Z}$. PS2 denotes the power series method using a multimodular approach (conserving memory). NEW denotes the hybrid algorithm using a congruence together with the Petersson-Rademacher series. The entry (mem) indicates that the computation could not be run on the machine due to insufficient memory.

n	Digits in c_n	PS (c_n)	PS2 (c_n)	NEW (c_n)	n'	NEW ($c_{n'}$)
10^2	53	0.00011	0.00017	0.00033	10^2+1	0.00087
10^3	171	0.0068	0.015	0.0016	10^3+9	0.0041
10^4	543	0.33	0.96	0.0054	10^4+7	0.027
10^5	1722	15	48	0.025	10^5+3	0.22
10^6	5453	625	2169	0.13	10^6+3	2.5
10^7	17253	(mem)	$\approx 8.2 \cdot 10^4$	0.83	10^7+19	32
10^8	54569		$\approx 3.2 \cdot 10^6$	6.5	10^8+7	380
10^9	172575		(mem)	60	10^9+7	(mem)
10^{10}	545743			636		

5.1. **Large values.** As a benchmark (Table 1), we compute c_n for various powers of ten $n = 10^k$ as well as $c_{n'}$ where n' is the first prime number after 10^k. Powers of ten are numbers of special form (n being divisible by $2^k 5^k$), meaning that the hybrid algorithm can use a congruence, skipping the power series evaluation. The subsequent prime number n' is of generic form (worst-case input) for the algorithm, forcing a power series evaluation to determine a residue.

For comparison purposes, we have implemented two versions of the power series algorithm to compute c_n: the first using arithmetic in $\mathbb{Z}$ and the second using arithmetic modulo many small primes to conserve memory (which turns out to be roughly three times slower). The first power series implementation is roughly equivalent to the function `j_invariant_qexp` in SageMath [17].

We see that the hybrid algorithm is faster than the power series method already around $n = 10^3$. At $n = 10^6$, it is 250 times faster for the generic input (prime n) and 4800 times faster for the special-form input (power-of-ten n). The $n^{1+o(1)}$ complexity of the hybrid algorithm is apparent in the running times.[1]

5.2. **Prime values.** The plethora of congruences satisfied by c_n conspire to rule out c_n being a prime number for small n, but nothing suggests that this pattern must hold asymptotically. Indeed, while $c_0, \ldots, c_{70}$ are all divisible by 2, 3 or 5, the smallest factor of c_{71} is 353.

To search for prime values of c_n, we used a single power series evaluation to compute several c_n simultaneously modulo the primorial $M = 2 \cdot 3 \cdots 47$, which fits in a 64-bit word. We then selected the entries with $\gcd(c_n, M) = 1$ and computed their full values using the hybrid algorithm with $c_n \bmod M$ as precomputed input. Prime values of c_n were then identified using a standard probabilistic primality test (trial factoring to rule out simple composites followed by the BPSW test).

[1] Arb does not presently use a quasi-optimal algorithm for the Bessel function $I_1(x)$ at high precision, so if we were to continue the table beyond 10^{10}, the timings would likely get worse.

TABLE 2. All prime values of c_n with $n \leq 2 \cdot 10^7$.

n	Digits in c_n	c_n
457871	3689	$3080163651 \ldots 2714076699$
685031	4513	$2912989222 \ldots 8765523019$
1029071	5532	$4025516131 \ldots 1099172019$
1101431	5723	$8315472348 \ldots 7940410921$
9407831	16734	$9603424490 \ldots 8550890201$
11769911	18718	$5971402918 \ldots 6331345197$
18437999	23429	$4474491259 \ldots 2242965811$

To search up to $n = 2 \cdot 10^7$, the initial power series calculation took three minutes. Filtering by $\gcd(c_n, M) = 1$ left 28971 candidate c_n to compute exactly (around 7 hours) and check for primality (around 45 hours).

Table 2 lists the first prime values of c_n. At the time of writing, Jeremy Rouse reports having certified the primality of c_{457871} using ECPP. The remaining numbers are only confirmed as probable primes, but the BPSW test has no known counterexamples, and we anticipate that ECPP certificates can be generated with some months of computation.

Although primes seem to appear sparsely in the sequence c_n (much more sparsely than for the partition function $p(n)$, for example), we speculate:

Conjecture 5.1. There are infinitely many n such that c_n is prime.

With more memory and a large number of cores, the search could easily be extended further (at least to $n = 10^9$). The method presented here could also be used to investigate other divisibility properties of the numbers c_n.

6. Generalization

The methods discussed here are not specific to the j-function. The Petersson-Rademacher series can be generalized to any modular function of weight 0 (see [3, Theorem 5.1]), and similar series can be constructed for the coefficients of a wide range of modular forms, or viewed combinatorially, for various partition-type sequences [16]. An interesting problem is to automate the efficient computation of such coefficients. Obtaining tight truncation bounds for Rademacher-type series seems difficult at the moment, but with the hybrid numerical-modular approach, more crude and generally applicable bounds can be used.

7. Source code

The author has made all source code and data behind this paper publicly available at `https://github.com/fredrik-johansson/jfunction`

References

[1] Hans-Fredrik Aas. Congruences for the coefficients of the modular invariant $j(\tau)$. *Mathematica Scandinavica*, 14(2):185–192, 1964.

[2] Harald Baier and Gunter Köhler. How to compute the coefficients of the elliptic modular function $j(z)$. *Experimental Mathematics*, 12(1):115–121, January 2003.

[3] Nicolas Brisebarre and Georges Philibert. Effective lower and upper bounds for the Fourier coefficients of powers of the modular invariant j. *Journal of the Ramanujan Mathematical Society*, 20(4):255–282, 2005.
[4] John H. Conway and Simon P. Norton. Monstrous moonshine. *Bulletin of the London Mathematical Society*, 11(3):308–339, 1979.
[5] David Feldman. Does the Fourier expansion of the j-function have any prime coefficients? `https://mathoverflow.net/q/377061`, 2020.
[6] William B. Hart. Fast library for number theory: An introduction. In *Mathematical Software – ICMS 2010*, pages 88–91. Springer Berlin Heidelberg, 2010.
[7] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/A000521`, 2020.
[8] Fredrik Johansson. Efficient implementation of the Hardy-Ramanujan-Rademacher formula. *LMS Journal of Computation and Mathematics*, 15:341–359, 2012.
[9] Fredrik Johansson. *Fast and rigorous computation of special functions to high precision*. PhD thesis, RISC, Johannes Kepler University, Linz, 2014. `http://fredrikj.net/thesis/`.
[10] Fredrik Johansson. Arb: Efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Transactions on Computers*, 66(8):1281–1292, August 2017.
[11] James Mc Laughlin and Scott Parsell. A Hardy-Ramanujan-Rademacher-type formula for (r, s)-regular partitions. *The Ramanujan Journal*, 28(2):253–271, April 2012.
[12] Morris Newman. Congruences for the coefficients of modular forms and for the coefficients of $j(\tau)$. *Proceedings of the American Mathematical Society*, 9(4):609, August 1958.
[13] Hans Petersson. Über die Entwicklungskoeffizienten der automorphen Formen. *Acta Mathematica*, 58(1):169–215, 1932.
[14] Hans Rademacher. The Fourier coefficients of the modular invariant $J(\tau)$. *American Journal of Mathematics*, 60(2):501, April 1938.
[15] Peter Sarnak and Jacob Tsimerman. On Linnik and Selberg's conjecture about sums of Kloosterman sums. In *Algebra, Arithmetic, and Geometry*, pages 619–635. Springer, 2009.
[16] Andrew Sills. Towards an automation of the circle method. *Gems in experimental mathematics*, pages 321–338, 2010.
[17] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.0)*, 2020. `https://www.sagemath.org`.
[18] Albert Leon Whiteman. A note on Kloosterman sums. *Bulletin of the American Mathematical Society*, 51(6):373–377, 1945.

Inria Bordeaux-Sud-Ouest and Institut Math. Bordeaux, U. Bordeaux, 33400 Talence, France
Email address: `fredrik.johansson@gmail.com`