Modular Forms Project: Computing Expansions at Cusps

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

June 20, 2018

1 Introduction

The present project is extracted from a paper of D. Collins [2]. It involves some numerical and implementation work. For this I strongly suggest to use the most recent (2.11, or at least 2.10.1) version of `Pari/GP`, although it can be done, with more difficulty, with other packages such as `Sage`.

Let $F \in M_k(\Gamma_0(N), \chi)$ be some modular form. The problem of *representing* F so as to be able to perform computations on a computer is not completely trivial. The most common and simplest is to give its *Fourier expansion* at infinity $F(\tau) = \sum_{n\ge 0} a(n)q^n$ with as usual $q = e^{2\pi i\tau}$. An alternative method is to give a *Taylor expansion* around some CM point in the upper-half plane (see e.g., Section 5.4 of [1] for details). We can also use *modular symbols* (see e.g., [3] for details).

For this project, we will only consider the first representation, using the Fourier expansion at infinity, and we assume that we can compute in reasonable time as many coefficients $a(n)$ as we need.

The problem that we want to discuss is the following: given a matrix $\gamma = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in \Gamma = \mathrm{PSL}_2(\mathbb{Z})$, find the Fourier expansion of $F|_k\gamma$, where by definition

$$(F|_k\gamma)(\tau) = (C\tau + D)^{-k} F\left(\frac{A\tau + B}{C\tau + D}\right) .$$

(note that we assume that $AD - BC = 1$).

The importance of this problem comes from that fact that it is essentially equivalent to the computation of the expansions of F at any *cusp*, since if A/C is some cusp with $\gcd(A, C) = 1$ we can always find $\gamma = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in \Gamma$ sending $i\infty$ to the given cusp.

Exercise 1. Let $F \in M_k(\Gamma_0(N), \chi)$, $\gamma = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in \Gamma$, and set $w_1 = N/\gcd(C, N)$.

(1) Show that $(F|_k\gamma)(\tau + w_1) = (F|_k\gamma)(\tau)$, hence that the above notion of Fourier expansion makes sense, more precisely that there exist coefficients $a_\gamma(n)$ such that

$$(F|_k\gamma)(\tau) = \sum_{n\ge 0} a_\gamma(n) q^{n/w_1} ,$$

where we always set $q^{u/v} := e^{2\pi i(u/v)\tau}$. Note that $a_\gamma(n) = 0$ for $n < 0$ since F is a modular *form*.

(2) Set $w_0 = N/\gcd(C^2, N)$ and $H = w_1/w_0 = \gcd(C^2, N)/\gcd(C, N)$. Show that there exists a unique integer $\alpha = \alpha(\gamma, \chi)$ such that

$$\chi(1 + ACw_0) = e^{2\pi i\alpha/H} \quad \text{and} \quad 0 \le \alpha < H \ .$$

Note that we will have $\alpha = 0$ either if χ is trivial, or if $H = 1$.

(3) Show that $a_\gamma(n) = 0$ unless $n \equiv \alpha \pmod{H}$, and deduce that we can write more precisely

$$(F|_k\gamma)(\tau) = q^{\alpha(\gamma,\chi)/w_1} \sum_{n \ge 0} b(n) q^{n/w_0}$$

for $b(n) = a_\gamma(nH + \alpha(\gamma, \chi))$.

The following exercise is not needed in the sequel, but is interesting in itself.

Exercise 2.

(1) Let $\gamma = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in \mathrm{GL}_2^+(\mathbb{Q})$ be a 2×2 matrix with rational entries and strictly positive determinant. Show that there exists a matrix $\delta \in \Gamma$ and an upper triangular matrix $\left(\begin{smallmatrix} a & b \\ 0 & d \end{smallmatrix}\right)$ such that $\gamma = \delta \left(\begin{smallmatrix} a & b \\ 0 & d \end{smallmatrix}\right)$, and give an algorithm for computing δ, a, b, and d.

(2) In the next section we will give an algorithm for computing $F|_k\delta = \sum_{n \ge 0} a_\delta(n) q^{n/w_1}$. Deduce from (1) the expansion of $F|_k\gamma$.

2 The Least Squares Approach

The method suggested by D. Collins, which is quite natural, is to use a *least squares* method as follows.

Choose some points $\tau_1, \dots, \tau_L$ in the upper half-plane, set $q_j = e^{2\pi i\tau_j/w_0}$, $r_j = e^{2\pi i\tau_j\alpha/w_1}$ (which will be equal to 1 when $\alpha = 0$, which is a frequent occurrence), and assume that we truncate the Fourier expansion $q^{\alpha/w_1} \sum_{n \ge 0} b(n) q^{n/w_0}$ to K terms so that the remainder is smaller than the accuracy that we want. We can then write an approximate matrix equality $MX \approx Y$ with

$$M = (r_i q_i^j)_{1 \le i \le L, 0 \le j < K} \ , \quad X = (b(0), \dots, b(K-1))^t \ , \quad \text{and}$$
$$Y = ((F|_k\gamma)(\tau_1), \dots, (F|_k\gamma)(\tau_L))^t$$

(I use ^{t} to denote transpose, so X and Y are column vectors).

Note that this is a *least squares* problem since it will be useful and necessary to have many more equations than unknowns (in practice, Collins suggests to use $L = 2K$). The best solution to such as least squares problem is well-known: it is the true solution to the pure linear algebra problem $(M^*M)X = M^*Y$, where M^* is the conjugate transpose of M, and M^*M is now a square positive-definite Hermitian $K \times K$ matrix.

The main thing that remains to be done is to choose appropriately M, K, and the points τ_i. Collins suggests the following algorithm (modified to use my notation):

Algorithm. Given $F = \sum_{n\ge 0} a(n)q^n \in M_k(\Gamma_0(N),\chi)$ and $\gamma = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right) \in \Gamma$, we want to compute $F|_k\gamma = q^{\alpha/w_1}\sum_{n\ge 0} b(n)q^{n/w_0}$. We fix constants E, K_0, and G_0, and we want to compute $b(n)$ for $0 \le n < K_0$ with accuracy approximately e^{-E+nG_0}.

(1) First increase $K = K_0$ (if $K_0G_0 < E$), or decrease $G = G_0$ (if $K_0G_0 > E$) so that $KG \approx E$.

(2) Set $L = 2K$, and choose L points $\tau_1, \dots, \tau_L$ such that $\Im(\tau_j) = Gw_0/(2\pi)$, and $\Re(\tau_j)$ is picked randomly in an interval of length w_0 centered around $-D/C$.

(3) Using the given q-expansion of F and truncating after we reach an accuracy somewhat smaller than e^{-E}, compute numerically $(F|_k\gamma)(\tau_j) = (C\tau_j + D)^{-k}F(\gamma(\tau_j))$ and put these values in a column vector Y.

(4) For $1 \le j \le L$ compute $s_j = e^{2\pi i\tau_j/w_1}$, then $q_j = s_j^H$, $r_j = s_j^{\alpha}$, and then the matrix $M = (r_iq_i^j)_{1\le i\le L, 0\le j<K}$.

(5) Compute the solution $X = (x(0), \dots, x(K-1))^t$ to the linear system $(M^*M)X = M^*Y$, and output $(x(0), \dots, x(K_0-1))^t$ (only up to $K_0 - 1$) as the approximate coefficients of the expansion of $F|_k\gamma$.

Note that the requirement that the accuracy of $b(n)$ is e^{-E+nG_0} (i.e., decreases by a factor e^{G_0} when n increases by 1) is quite a natural one.

Exercise 3.

(1) Implement the above algorithm in your favorite language/package.

(2) As a first check of correctness, check that $F|_k\gamma = F$ for some $F \in M_k(\Gamma)$ (i.e., F modular for the full modular group), such as $F = E_4$ or $F = \Delta$ and some $\gamma \in \Gamma$.

(3) Test the following example from Collins's paper: the unique newform $F \in S_4(\Gamma_0(6))$ with Fourier expansion $F = q - 2q^2 - 3q^3 + 4q^4 + 6q^5 + 6q^6 - 16q^7 + \cdots$, and $\gamma = \left(\begin{smallmatrix} 1 & -1 \\ 3 & -2 \end{smallmatrix}\right)$, and check that numerically $F|_4\gamma \approx F/4$.

To evaluate $F(\tau)$ you have at least two methods: the first is to use directly the `Pari/GP` commands:

```
mf=mfinit([6,4],0); F=mfeigenbasis(mf)[1]; mfeval(mf,F,tau)
```

The second method, which is the simplest in this specific case, prove that

$$F(\tau) = (\eta(\tau)\eta(2\tau)\eta(3\tau)\eta(6\tau))^2,$$

and it is then immediate to compute $\eta(\tau)$ either by its power series expansion or by the GP command `eta(tau,1)` (do not forget the `,1`).

You can also check the result by using the `Pari/GP` command

```
mfslashexpansion(mf,F,ga,20,1)
```

(see the explanation of this below). Note that `Pari/GP` is the only software which is currently able to do this kind of computation.

(4) For 20 coefficients, say, try different values of G_0 such as $G_0 = 1/2$, 1, 2, 3 to see the accuracy that you obtain and the time taken for a given accuracy.

(5) Test the following example from Collins's paper, which is more complicated because the level is not squarefree: there exists an eigenform $F \in S_4(\Gamma_0(27))$ whose Fourier expansion begins

$$F(\tau) = q - 3q^2 + q^4 - 15q^5 - 25q^7 + 21q^8 + 45q^{10} + \cdots$$

This specific form can be obtained in `Pari/GP` by

```
mf=mfinit([27,4],0);F=mfeigenbasis(mf)[2];
```

(to check the Fourier expansion: `mfcoefs(F,10)`). Compute again $F|_k\gamma$ with the same γ as before, and check that the coefficients are (close to) simple linear combinations of ζ_9^j for $0 \le j \le 5$, where $\zeta_9 = e^{2\pi i/9}$ is a root of $t^6 + t^3 + 1 = 0$ (if you are using `Pari/GP`, this is the `lindep` command, such as `lindep(concat(-z,powers(exp(2*Pi*I/9),5)))`). Check your computations using the `mfslashexpansion` command. Once again experiment with the value of G_0.

Note on the `mfslashexpansion` command: in the present release of `Pari/GP`, this command takes six parameters, the last being optional:

```
mfslashexpansion(mf,F,gamma,n,fl,&params)
```

Here `mf`, `F`, and `gamma` are self-explanatory, as above. `n` is *one less* than the number of Fourier coefficients that you want since the coefficients will be $b(0), \ldots, b(n)$ (so it is equal to $K_0 - 1$ with K_0 as in Collins's algorithm). `fl` is set to 0 if you want the result as a complex number, or to 1 if you want to obtain the result as an explicit algebraic number (if possible: you may have an error message if the software cannot find reasonable algebraic approximations). Finally the optional parameter `params` will contain a triple $[\alpha/w_1, w_0, g]$ with the same notation as above, where g is a 2×2 matrix which for now should be ignored (and in any case is usually equal to the identity).

In particular, if you simply want to compute the parameter α which is necessary in the algorithm, you can slightly cheat and simply write

```
mfslashexpansion(mf,F,gamma,0,0,&params)
```

and recover α as `params[1]*w1`, where `w1=N/gcd(C,N)`.

References

[1] H. Cohen and F. Strömberg, *Modular Forms: A Classical Approach*, Graduate Studies in Math. **179**, American Math. Soc., (2017).

[2] D. Collins, *Numerical Computation of Petersson Inner Products and q-Expansions*, arXiv 1802.09740.pdf (Feb. 2018), 24 p.

[3] W. Stein, *Modular forms, a computational approach*, Graduate Studies in Math. **79**, American Math. Soc., 2007.