Hejhal's Algorithm

Alex Karlovitz

Fourier Expansion of a Maass Form

Let f be a Maass form for some Fuchsian group Γ which contains $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$. Since $f(z+1) = f(z)$, it has a Fourier expansion in its real variable, say

$$f(z) = \sum_{n=-\infty}^{\infty} a_n(y) e(nx)$$

Now, let's choose $\nu \in \mathbb{C}$ such that the Laplacian eigenvalue of f is $1/4 - \nu^2$. Then we know that

$$\Delta(a_n(y)e(nx)) = \left(\frac{1}{4} - \nu^2\right) a_n(y)e(nx)$$

(by uniqueness of Fourier coefficients). A clever choice of $a_n(y)$ makes this differential equation work beautifully.

Exercise: for $n \neq 0$, check that setting $a_n(y) = \sqrt{y}k(2\pi|n|y)$ reduces to Bessel's differential equation

$$\left\{ y^2 \frac{d^2}{dy^2} + y\frac{d}{dy} - (y^2 + \nu^2) \right\} k(y) = 0$$

Now, the above differential equation has two solutions, K_ν and I_ν. However, while $K_\nu(y)$ decays exponentially as $y \to \infty$, $I_\nu(y)$ grows exponentially. Since Maass forms cannot grow exponentially at a cusp, we must have that $k = K_\nu$.

Conclusion. If f is a Maass cusp form with Laplacian eigenvalue $1/4-\nu^2$, then it has a Fourier expansion of the form

$$f(z) = \sum_{n \neq 0} a_n \sqrt{y} K_\nu(2\pi|n|y)e(nx)$$

Eigenvalues in L^2

By applying integration by parts multiple times, one can show that the hyperbolic Laplacian is a positive semidefinite operator with respect to the Petersson inner product on $L^2(\Gamma\backslash\mathbb{H})$. Thus, if $f \in L^2(\Gamma\backslash\mathbb{H})$ is an eigenfunction of the Laplacian with eigenvalue $\lambda = 1/4 - \nu^2$, we must have that λ is real and nonnegative. This can only occur if $\nu = ir$ for $r \in \mathbb{R}$, or if $\nu \in \mathbb{R}$ with $|\nu| < 1/2$. If ν is real and nonzero, then $\lambda < 1/4$ is called an *exceptional* eigenvalue. In the case of Maass forms on $SL_2(\mathbb{Z})$, Selberg [**?**] showed that there are no exceptional eigenvalues.

Hejhal's Algorithm

The discussion I am presenting is taken from a paper by Booker/Strömbergsson/Venkatesh called "Effective Computation of Maass Cusp Forms." They describe the algorithm in great detail, as well as details on how to implement the algorithm in high precision.

Assumptions

I will be describing the algorithm for $\Gamma = SL_2(\mathbb{Z})$, but as I state it, the algorithm will work under the following assumptions.

- $\Gamma \subset SL_2(\mathbb{Z})$ is a cofinite Fuchsian group
- $\Gamma\backslash\mathbb{H}$ has exactly one cusp, which we can take to be positioned at ∞
- Γ_∞ is generated by $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$

The first assumption is necessary for the algorithm; see the paper by BSV for multiple cusps and/or different widths.

Setup

Let us fix a fundamental domain $\mathcal{F} \subset \mathbb{H}$ for Γ with the cusp positioned at ∞. Since this is the only cusp, we have that $Y_0 := \inf\{\mathrm{im}(z) : z \in \mathcal{F}\}$ is strictly positive. Let us also fix a positive integer D indicating the number of digits of precision we are aiming for.

Now we suppose that $f(z)$ is a fixed Maass cusp form of eigenvalue $\lambda = \frac{1}{4} + r^2$ for some $r \in \mathbb{R}$. From our earlier discussion, we know that the Fourier expansion of f has a very specific form. We follow a common convention and write

$$f(z) = \sum_{n \neq 0} a_n \sqrt{y} \kappa_{ir}(2\pi|n|y) e(nx)$$

where $\kappa_{ir}(u) := e^{\frac{\pi}{2}r} K_{ir}(u)$. (This scaling is a numerical convention meant to control the decay rate of κ_{ir} independently of r).

Now, it is known that the coefficients satisfy $a_n = O(|n|^{\frac{1}{3}+\epsilon})$. (Notice that the exponential decay of κ_{ir} ensures the entire Fourier coefficient still goes to 0 exponentially as $|n| \to \infty$). Knowing this, we should be able to construct (perhaps after some experimentation) a decreasing function $M(y)$ so that

$$f(z) = \sum_{0<|n|\leq M(y)} a_n \sqrt{y} \kappa_{ir}(2\pi|n|y) e(nx) + [[10^{-D}]]$$

where $[[10^{-D}]]$ is meant to denote a quantity of absolute value less than 10^{-D}. The fact that $M(y)$ is decreasing follows from the exponential decay of κ_{ir}. To keep at least some of our finite Fourier expansions the same size, let us declare now that $M(y) = M(Y_0) =: M_0$ for all $y \geq Y_0$.

So far, we have a way of viewing $f(x+iy)$ as a finite Fourier series for each $y > 0$. Next, we will use the Γ-automorphicity of f to create a linear system. To do this, we will start with sample points taken from a horocycle at height $Y < Y_0$ (recall the theorem regarding equidistribution of low-lying horocycles). Let Q be an integer greater than $M(Y)$, then define the $2Q$ points

$$z_j = x_j + iY = \frac{1}{2Q}\left(j - \frac{1}{2}\right) + iY \qquad\qquad 1 - Q \leq j \leq Q$$

These are the $2Q$ equally-spaced points starting at $-\frac{1}{2} + \frac{1}{4Q}$ and ending at $\frac{1}{2} - \frac{1}{4Q}$. Applying the finite Fourier series at each of these points and swapping the order of summation, we get a formula for the coefficients in terms of the function at the sample points; namely, for each $|n| \leq M(y)$, we have

$$a_n \sqrt{Y} \kappa_{ir}(2\pi|n|Y) = \frac{1}{2Q} \sum_{j=1-Q}^{Q} f(z_j) e(-nx_j) + [[10^{-D}]]$$

Of course, we don't know what f is, so we can't evaluate it at the sample points. Instead, we apply the automorphicity of f to find for each j the $\mathcal{F}$-pullback of z_j. Let's write T_j for the unique map in Γ satisfying $z_j^* := T_j(z_j) \in \mathcal{F}$. Plugging $f(z_j) = f(z_j^*)$ into the previous equation, then applying the finite Fourier expansion at z_j and again swapping the order of integration, we get for each $|n| \leq M(Y)$ that

$$a_n \sqrt{Y} \kappa_{ir}(2\pi|n|Y) = \sum_{0<|\ell|\leq M_0} a_\ell V_{n\ell} + 2[[10^{-D}]] \tag{1}$$

where

$$V_{n\ell} = \frac{1}{2Q} \sum_{j=1-Q}^{Q} \sqrt{y_j^*} \kappa_{ir}(2\pi|\ell|y_j^*) e(\ell x_j^* - n x_j)$$

Now, notice that the linear system has an obvious solution: $a_n = 0$ for all n. That the system is singular should not actually come as a surprise, for recall that f being a Maass form means that f is an eigenfunction of the hyperbolic Laplacian. Hence, any scaling of f by a complex scalar should preserve all of the work we've done so far. To account for this, we look for the solution with $a_1 = 1$. So long as the true solution does not have $a_1 = 0$ (which it can be shown is impossible in the case $\Gamma = SL(2, \mathbb{Z})$), this method will apply.

The Algorithm

Let's review what we've learned so far. For any given Maass cusp form $f(z)$ with eigenvalue $\lambda = \frac{1}{4} + r^2$, we have that Equation 1 holds for each $|n| \leq M(Y)$. If we assume r is known, the only unknowns in this equation are the coefficients a_n. To get a square system, we simply restrict our attention to the $2M_0$ equations where $|n| \leq M_0$ (recall that $M_0 = M(Y_0) \leq M(Y)$). Hence, we can solve the linear system for the a_n's.

Now remember, the point of this algorithm is to compute the Laplacian eigenvalue r. So we proceed as follows.

- Guess a value for r.
- Solve the linear system for two different Y values, say $0 < Y_2 < Y_1 < Y_0$.
- Successively adjust r and repeat this process to make the two solution vectors as close as possible.

The idea is that the coefficients a_n are independent of the height Y. So if the r we are using is indeed an eigenvalue of a Maass cusp form, then the two solution vectors obtained from solving the linear system will be identical.

A few questions may come to mind at this moment. Will the linear system be well-conditioned enough to step towards a correct r? If the algorithm converges on some value for r, is it a true eigenvalue of a Maass form?

A More General Version of Hejhal's Algorithm

In this section, we describe how Hejhal's algorithm can be applied in a larger variety of situations. Specifically, we allow for the possibility of different Fourier expansions (not just the one at the cusp). Currently, we are experimenting with three types of expansions:

- cuspidal (described above)
- flare expansions (in the case of an infinite volume Fuchsian group)
- expansions in the disk model

In every case, we set up linear systems parameterized by the eigenvalue guesses to solve for the Fourier coefficients, then make new guesses based on those results. We now describe this process in detail.

The Linear System

We assume that $\Gamma \subseteq \mathrm{SL}(2, \mathbb{R})$ is a Fuchsian group, and f is a Maass form in $L^2(\Gamma \backslash \mathbb{H})$. We also assume that f has some Fourier expansion; using the fact that f is an eigenfunction of the Laplacian, we assume that we have an explicit formula for the Fourier coefficients. Let us write

$$f(x+iy) = \sum_{n \in \mathbb{Z}} a_n c_n(y) e(nx)$$

(In the flare and disk expansions, the expansion is in polar coordinates; but the argument goes through in the same way). It is an important fact that the functions $c_n(y)$ will depend on the eigenvalue associated with f; i.e., $c_n(y) = c_n(y; \lambda)$.

We are now ready to describe the linear system. First, we approximate the Maass form by a finite Fourier expansion; that is, we sum over $|n| \leq M < \infty$. Then, we take $N \geq 2M + 1$ test points $z_j \in \mathbb{H}$ (or in $\mathbb{D}$, in the case of the disk model). Finally, the linear system is

$$f_a(z_j) \approx f_a(z_j^*),\ j = 1, \ldots, N$$

where z_j^* is the pullback of z_j to some fixed fundamental domain and $f_a(z)$ is the approximation to $f(z)$ achieved from the truncated expansion.

We make a few quick notes about the above linear system.

- If $N > 2M + 1$, we will have more equations than variables; in this case, we solve for the coefficients using some sort of least squares.
- The linear system is only accurate if we input the correct eigenvalue into the $c_n(y; \lambda)$'s; we are hoping that the dependence on λ is smooth, so that we may employ some algorithm to step towards the true value.

Approximating the True Eigenvalue

Given a guess for the eigenvalue λ and a set of test points, we now know how to set up a linear system to solve for the Fourier coefficients. If we do this for two different sets of test points, we will get two different sets of coefficients (since the eigenvalue may not be exactly right). Call these vectors

$$\vec{x}_1(\lambda) \qquad \text{and} \qquad \vec{x}_2(\lambda)$$

We are writing these as functions of λ since - as noted above - the linear system depends on λ. Now we want to describe an algorithm for making a new guess for the eigenvalue. There are various ways to do this, some of which are described below.

Grid Method

Write $\lambda = \frac{1}{4} - \nu^2$, so that ν is either real with $|\nu| < 1/2$ or $\nu = ir$ is purely imaginary. We start with a guess ν_0 for ν, then picking a grid of guesses nearby ν_0; e.g., take 10 values of ν each a step size of 1/10 apart, centered around ν_0. For each grid point ν_*, compute the two sets of Fourier coefficients $\vec{x}_1(\nu_*)$ and $\vec{x}_2(\nu_*)$ (note that these are functions of ν just as well as functions of λ). Then compute the error

$$||\vec{x}_1(\nu_*) - \vec{x}_2(\nu_*)||$$

Our next guess ν_1 will be the grid point with minimal error. We can repeat this process as many times as we wish.

Secant Method

Write $\lambda = s(1 - s)$ (this is yet another common notation). We start with two guesses: s and $s + \delta$ (assume $\delta > 0$). Consider the guess s first: we compute the two sets of Fourier coefficients $\vec{x}_1(s)$ and $\vec{x}_2(s)$ as above. We then compute the difference between the coefficients at four specific indices (this arbitrary; we could use more or less indices if we wish):

$$c_j(s) = \vec{x}_1(s)[i_j] - \vec{x}_2(s)[i_j], \quad j = 1, \ldots, 4$$

We then do the same for the guess $s + \delta$. Note that $c_j(s)$ will be very close to 0 if $s(1 - s)$ is very close to the true eigenvalue. Thus, we employ the secant method from numerical anlysis to make a new guess:

$$s_j = \frac{c_j(s)(s + \delta) - c_j(s + \delta)s}{c_j(s) - c_j(s + \delta)}$$

Of course, we have just formed 4 new guesses. (The heuristic reason for this is that one of c_j's may be a bad indicator of how good a guess s is for some random reason; it is less likely that all four c_j's are bad). We proceed by setting

- $s_{\max} = \max\{s_j\}, s_{\min} = \min\{s_j\}, s_{\text{mid}} = (s_{\max} + s_{\min})/2$
- $\delta_{\text{new}} = 5(s_{\max} - s_{\min})$

If $\delta_{\text{new}} > \delta/2$, we don't expect to improve by taking any more steps, so we make our final guess s_{mid}. Otherwise, we repeat the above process with the new guesses s_{mid} and $s_{\text{mid}} + \delta_{\text{new}}$.