Numerical computations of Maass cusp forms

Andrei Seymour-Howell

University of Bristol

June 9, 2021

Maass cusp forms

Let $\mathbb{H} = \{z = x + iy | y > 0\}$ denote the upper half-plane. We define the Hecke congruence subgroup $\Gamma_0(N) < \mathrm{SL}(2, \mathbb{Z})$ by

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2, \mathbb{Z}) | c \equiv 0 \pmod{N} \right\}$$

for $N > 0$.

Maass cusp forms

Let $\mathbb{H} = \{z = x + iy | y > 0\}$ denote the upper half-plane. We define the Hecke congruence subgroup $\Gamma_0(N) < \mathrm{SL}(2, \mathbb{Z})$ by

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2, \mathbb{Z}) | c \equiv 0 \pmod{N} \right\}$$

for $N > 0$. This group acts on $\mathbb{H}$ by linear fractional transformations, i.e

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} z = \frac{az + b}{cz + d} \qquad \forall \gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(N), z \in \mathbb{H}.$$

Maass cusp forms

Let $\mathbb{H} = \{z = x + iy | y > 0\}$ denote the upper half-plane. We define the Hecke congruence subgroup $\Gamma_0(N) < \mathrm{SL}(2, \mathbb{Z})$ by

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2, \mathbb{Z}) | c \equiv 0 \pmod{N} \right\}$$

for $N > 0$. This group acts on $\mathbb{H}$ by linear fractional transformations, i.e

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} z = \frac{az + b}{cz + d} \qquad \forall \gamma = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(N), z \in \mathbb{H}.$$

The modular surface $X = \Gamma_0(N) \backslash \mathbb{H}$ is a finite volume non-compact surface with Laplacian

$$\Delta = -y^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right).$$

We also have the measure

$$\frac{dx\, dy}{y^2}.$$

Level 1 - SL$(2, \mathbb{Z})$

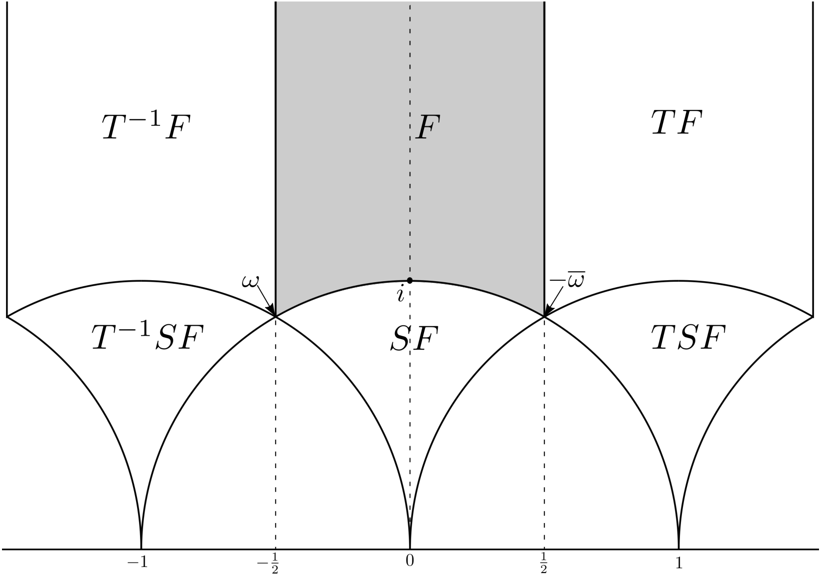
$\Gamma_0(1) = \mathrm{SL}(2, \mathbb{Z})$ is generated by the two matrices

$$T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad\qquad S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

This gives the following fundamental domain for this action

$$F = \left\{ z \in \mathbb{H} : |z| \geq 1 \text{ and } |\mathrm{Re}(z)| \leq \frac{1}{2} \right\}.$$

Fundamental domain



Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

1. f is an eigenfunction of the Laplacian, $\Delta f = \lambda f, \lambda \geq 0$,

Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

1. f is an eigenfunction of the Laplacian, $\Delta f = \lambda f, \lambda \geq 0$,
2. f is automorphic, $f(\gamma z) = f(z)$ for all $\gamma \in \Gamma_0(N)$,

Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

1. f is an eigenfunction of the Laplacian, $\Delta f = \lambda f, \lambda \geq 0$,
2. f is automorphic, $f(\gamma z) = f(z)$ for all $\gamma \in \Gamma_0(N)$,
3. $f \in L^2(X)$, i.e f is square-integrable,

Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

1. f is an eigenfunction of the Laplacian, $\Delta f = \lambda f, \lambda \geq 0$,
2. f is automorphic, $f(\gamma z) = f(z)$ for all $\gamma \in \Gamma_0(N)$,
3. $f \in L^2(X)$, i.e f is square-integrable,
4. f vanishes at all of the cusps of X.

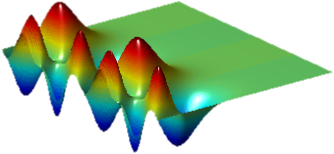
Maass cusp forms

We call a function $f : \mathbb{H} \to \mathbb{C}$ a **Maass cusp form** of level N (trivial character) if

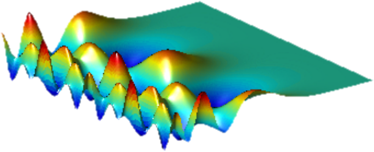
1. f is an eigenfunction of the Laplacian, $\Delta f = \lambda f, \lambda \geq 0$,
2. f is automorphic, $f(\gamma z) = f(z)$ for all $\gamma \in \Gamma_0(N)$,
3. $f \in L^2(X)$, i.e f is square-integrable,
4. f vanishes at all of the cusps of X.

We will denote the space of Maass cusp forms of level N and Laplace eigenvalue λ by $\mathcal{S}_\lambda(N)$.
The set of functions that just satisfy points (2), (3) and (4) we shall denote as $L^2_{\text{cusp}}(X)$.

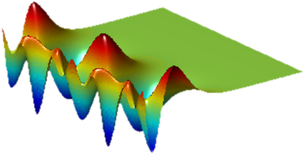
Pictures of Maass forms



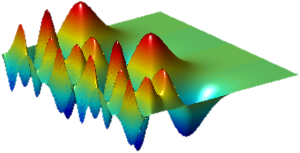
(a) Level 1, $\lambda = 91.141345\ldots$



(b) Level 1, $\lambda = 190.131547\ldots$



(c) Level 2, $\lambda = 79.867724\ldots$



(d) Level 3, $\lambda = 182.713668\ldots$

Figure: Images of Maass forms from the LMFDB.

Hecke operators

For any $f \in \mathcal{S}_\lambda(N)$ and any non-zero integer n coprime to N, we define the **Hecke operator** T_n by

$$T_n f(z) = \frac{1}{\sqrt{|n|}} \sum_{\substack{ad=n \\ (a,N)=1 \\ d>0}} \sum_{j=0}^{d-1} \begin{cases} f\left(\dfrac{az+j}{d}\right) & \text{if } n > 0, \\ f\left(\dfrac{a\bar{z}+j}{d}\right) & \text{if } n < 0. \end{cases}$$

This will map $\mathcal{S}_\lambda(N) \to \mathcal{S}_\lambda(N)$.

Hecke operators

For any $f \in \mathcal{S}_\lambda(N)$ and any non-zero integer n coprime to N, we define the **Hecke operator** T_n by

$$T_n f(z) = \frac{1}{\sqrt{|n|}} \sum_{\substack{ad=n \\ (a,N)=1 \\ d>0}} \sum_{j=0}^{d-1} \begin{cases} f\left(\frac{az+j}{d}\right) & \text{if } n > 0, \\ f\left(\frac{a\bar{z}+j}{d}\right) & \text{if } n < 0. \end{cases}$$

This will map $\mathcal{S}_\lambda(N) \to \mathcal{S}_\lambda(N)$.
Now a famous result then tells us that there exists an orthogonal basis $\{f_j\}$ in $L^2_{\text{cusp}}(X)$ consisting of eigenfunctions to all Hecke operators T_n with $(n, N) = 1$.

Hecke eigenvalues

A Maass cusp form f of level N and with Laplace eigenvalue $\lambda = \frac{1}{4} + R^2$ has a Fourier expansion (at ∞) of the form

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

where $e(nx) = \exp(2\pi inx)$ and $K_\nu(u)$ is the K-Bessel function.

Hecke eigenvalues

A Maass cusp form f of level N and with Laplace eigenvalue $\lambda = \frac{1}{4} + R^2$ has a Fourier expansion (at ∞) of the form

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

where $e(nx) = \exp(2\pi inx)$ and $K_\nu(u)$ is the K-Bessel function.
We call a Maass form f **even** if $a(-n) = a(n)$ or **odd** if $a(-n) = -a(n)$.

Hecke eigenvalues

A Maass cusp form f of level N and with Laplace eigenvalue $\lambda = \frac{1}{4} + R^2$ has a Fourier expansion (at ∞) of the form

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

where $e(nx) = \exp(2\pi i n x)$ and $K_\nu(u)$ is the K-Bessel function.
We call a Maass form f **even** if $a(-n) = a(n)$ or **odd** if $a(-n) = -a(n)$.
If f is also a Hecke eigenfunction for all Hecke operators T_n with $(n, N) = 1$, i.e $T_n f = \lambda(n) f$, then we can normalise such that $a(1) = 1$ and we have

$$a(n) = \lambda(n)$$
$$a(-n) = \varepsilon \lambda(n)$$

where ε is 1 is f is even and -1 if f is odd.

Hejhal's Algorithm

There are a few methods known for computing Maass forms, the most widely used is an algorithm due to Hejhal from the 1990's. The algorithm goes in the following steps

Hejhal's Algorithm

There are a few methods known for computing Maass forms, the most widely used is an algorithm due to Hejhal from the 1990's. The algorithm goes in the following steps

1. Truncate the Fourier series and treat it like a discrete Fourier series.

$$f(z) = f(x+iy) = \sum_{0<|n|\leq M} a(n)\sqrt{y}K_{iR}(2\pi|n|y)e(nx) + [[\varepsilon]].$$

Hejhal's Algorithm

There are a few methods known for computing Maass forms, the most widely used is an algorithm due to Hejhal from the 1990's. The algorithm goes in the following steps

1. Truncate the Fourier series and treat it like a discrete Fourier series.

$$f(z) = f(x+iy) = \sum_{0<|n|\leq M} a(n)\sqrt{y}K_{iR}(2\pi|n|y)e(nx) + [[\varepsilon]].$$

2. Do an inverse Fourier transform along a certain horocycle of points away from the fundamental domain.

Hejhal's Algorithm

There are a few methods known for computing Maass forms, the most widely used is an algorithm due to Hejhal from the 1990's. The algorithm goes in the following steps

1. Truncate the Fourier series and treat it like a discrete Fourier series.

$$f(z) = f(x+iy) = \sum_{0<|n|\leq M} a(n)\sqrt{y}K_{iR}(2\pi|n|y)e(nx) + [[\varepsilon]].$$

2. Do an inverse Fourier transform along a certain horocycle of points away from the fundamental domain.
3. This will give an expression for the Fourier coefficients, however to make it a non-tautology, we use the automorphy of the Maass form to produce a linear system for the Fourier coefficients.

Hejhal's Algorithm

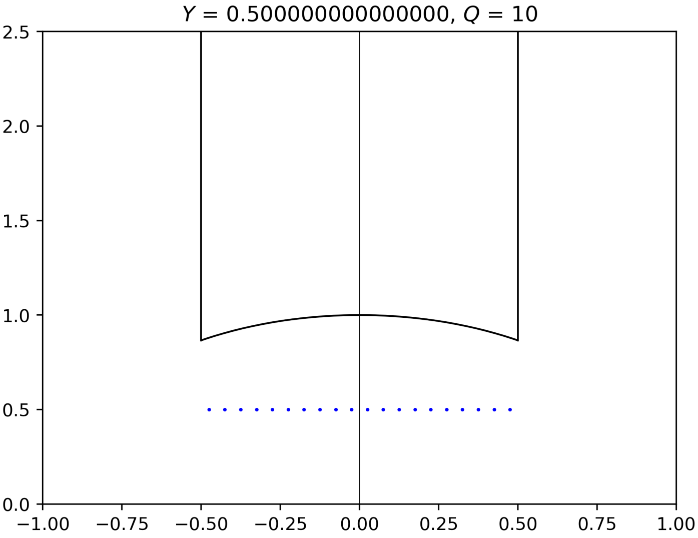
There are a few methods known for computing Maass forms, the most widely used is an algorithm due to Hejhal from the 1990's. The algorithm goes in the following steps

1. Truncate the Fourier series and treat it like a discrete Fourier series.

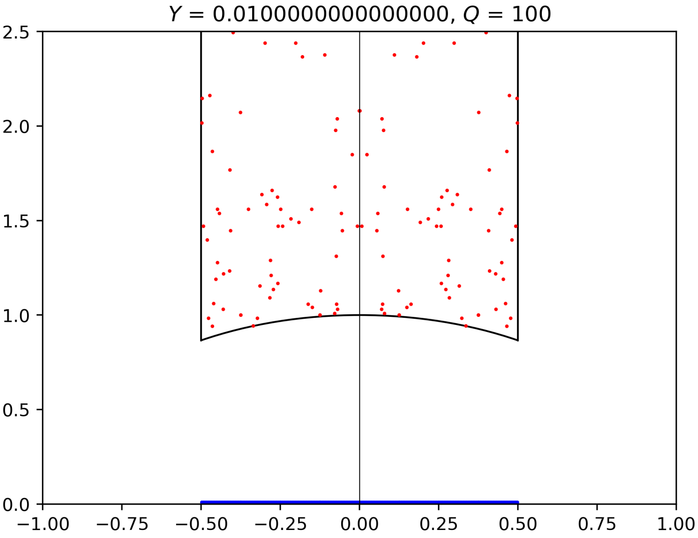
$$f(z) = f(x+iy) = \sum_{0<|n|\leq M} a(n)\sqrt{y}K_{iR}(2\pi|n|y)e(nx) + [[\varepsilon]].$$

2. Do an inverse Fourier transform along a certain horocycle of points away from the fundamental domain.
3. This will give an expression for the Fourier coefficients, however to make it a non-tautology, we use the automorphy of the Maass form to produce a linear system for the Fourier coefficients.
4. We then use a non-linear search strategy to zoom in on an Laplace eigenvalue.

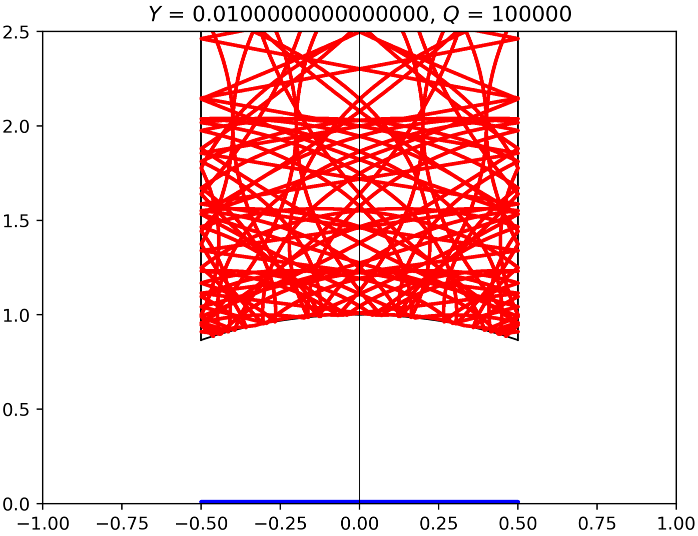
Picture of z_m



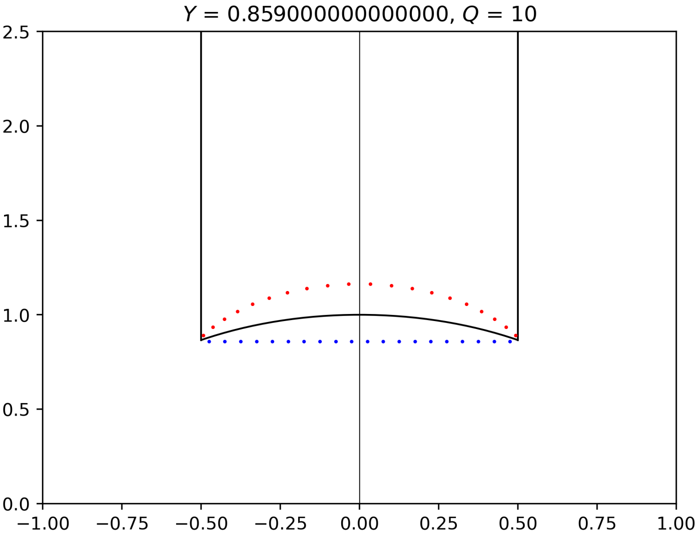
Picture of z_m



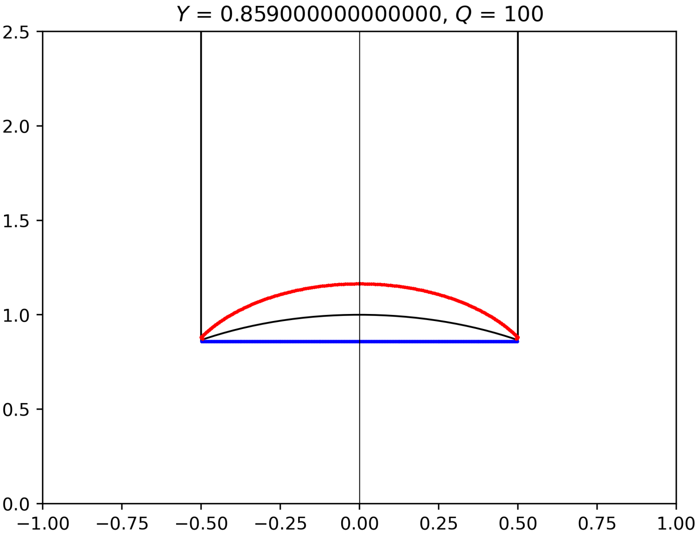
Picture of z_m



Picture of z_m



Picture of z_m



List of eigenvalues for $\mathrm{SL}(2, \mathbb{Z})$

R_1	$9.53369526135\ldots$
R_2	$12.1730083247\ldots$
R_3	$13.7797513519\ldots$
R_4	$14.3585095183\ldots$
R_5	$16.1380731715\ldots$
R_6	$16.6442592019\ldots$
R_7	$17.7385633811\ldots$
R_8	$18.1809178345\ldots$
R_9	$19.4234814708\ldots$
R_{10}	$19.4847138547\ldots$

Table: List of first 10 eigenvalues R on $\mathrm{SL}(2, \mathbb{Z})$. Data from the LMFDB.

Remarks

- The search strategy for finding R tends to involve computing a set of Fourier coefficients for 2 different R values and then minimise by either measuring the difference or testing their multiplicativity.

Remarks

- The search strategy for finding R tends to involve computing a set of Fourier coefficients for 2 different R values and then minimise by either measuring the difference or testing their multiplicativity.
- This method is heuristic, so we require another method to certify if the data produced is correct.

Remarks

- The search strategy for finding R tends to involve computing a set of Fourier coefficients for 2 different R values and then minimise by either measuring the difference or testing their multiplicativity.
- This method is heuristic, so we require another method to certify if the data produced is correct.
- Despite being heuristic, in practice this method is very stable and can produce results to high precision.

Remarks

- The search strategy for finding R tends to involve computing a set of Fourier coefficients for 2 different R values and then minimise by either measuring the difference or testing their multiplicativity.
- This method is heuristic, so we require another method to certify if the data produced is correct.
- Despite being heuristic, in practice this method is very stable and can produce results to high precision.
- There are nearly 15000 examples of these Maass cusp forms computed and stored on the LMFDB.

Remarks

- The search strategy for finding R tends to involve computing a set of Fourier coefficients for 2 different R values and then minimise by either measuring the difference or testing their multiplicativity.
- This method is heuristic, so we require another method to certify if the data produced is correct.
- Despite being heuristic, in practice this method is very stable and can produce results to high precision.
- There are nearly 15000 examples of these Maass cusp forms computed and stored on the LMFDB.
- There is also a phase 2 to this algorithm which allows us to compute more Fourier coefficients once we have a good approximation to R and its first few Fourier coefficients.

Verification methods

- In 2006, Booker, Strömbergsson and Venkatesh proved that it is possible to certify whether a candidate Maass form is "close" to a true Maass form. Roughly, suppose you have a computed eigenvalue $\tilde{\lambda} = \frac{1}{4} + \tilde{R}^2$ and the coefficients of a suspected Maass form $\tilde{f}$. Then they showed that if $\tilde{f}$ is "almost automorphic", then $\tilde{f}$ is "close" to a true Maass cusp form f. They only showed this for level 1, i.e $\mathrm{SL}(2, \mathbb{Z})$ and computed and verified the first few Maass cusp forms to a hundred digits.

Verification methods

- In 2006, Booker, Strömbergsson and Venkatesh proved that it is possible to certify whether a candidate Maass form is "close" to a true Maass form. Roughly, suppose you have a computed eigenvalue $\tilde{\lambda} = \frac{1}{4} + \tilde{R}^2$ and the coefficients of a suspected Maass form $\tilde{f}$. Then they showed that if $\tilde{f}$ is "almost automorphic", then $\tilde{f}$ is "close" to a true Maass cusp form f. They only showed this for level 1, i.e $\mathrm{SL}(2, \mathbb{Z})$ and computed and verified the first few Maass cusp forms to a hundred digits.
- For congruence subgroups of $\mathrm{SL}(2, \mathbb{Z})$ current work is being done to verify the Laplace eigenvalues using that method that relies on an explicit version of the Selberg trace formula.

Verification methods

- In 2006, Booker, Strömbergsson and Venkatesh proved that it is possible to certify whether a candidate Maass form is "close" to a true Maass form. Roughly, suppose you have a computed eigenvalue $\tilde{\lambda} = \frac{1}{4} + \tilde{R}^2$ and the coefficients of a suspected Maass form $\tilde{f}$. Then they showed that if $\tilde{f}$ is "almost automorphic", then $\tilde{f}$ is "close" to a true Maass cusp form f. They only showed this for level 1, i.e $\mathrm{SL}(2,\mathbb{Z})$ and computed and verified the first few Maass cusp forms to a hundred digits.
- For congruence subgroups of $\mathrm{SL}(2,\mathbb{Z})$ current work is being done to verify the Laplace eigenvalues using that method that relies on an explicit version of the Selberg trace formula.
- Little plug: In order to do rigorous computations we use interval arithmetic. A library I use for this is ARB library for C which does arbitrary ball arithmetic.

Thanks for listening!