

COMPUTATION OF MAASS CUSP FORMS ON  
MODULAR GROUP IN *MATHEMATICA*

Abubaker Ahmed M. Siddig<sup>1</sup>, Hishamuddin Zainuddin<sup>2</sup> §

<sup>1,2</sup>Institute for Mathematical Research

University Putra Malaysia

43400 UPM, Serdang, Selangor, MALAYSIA

<sup>1</sup>e-mail: siddig04@gmail.com

<sup>2</sup>e-mail: hisham@putra.upm.edu.my

<sup>2</sup>Department of Physics

Faculty of Science

University Putra Malaysia

43400 UPM, Serdang, Selangor, MALAYSIA

**Abstract:** Maass cusp forms of the Fuchsian group are eigenfunctions of the hyperbolic Laplacian representing bound states. It is known that the Maass cusp forms can only be solved numerically. The aim of this paper is to extend Hejhal's algorithm to compute Maass cusp form on the modular group  $SL(2, \mathbb{R}) \backslash H$  using *Mathematica* with the intent of exploiting its graphical and symbolic capabilities. We compute the eigenvalues of the Maass cusp forms employing a graphical scheme to locate the approximate value of the eigenvalue. We also plotted Maass cusp forms corresponding to selected eigenvalues.

**AMS Subject Classification:** 11F12, 65Y15, 11F7

**Key Words:** maass cusp forms, *Mathematica*, modular group

## 1. Introduction

The computational work of Maass waveforms on  $PSL(2, \mathbb{Z}) \backslash H$  has been consid-

---

Received: June 11, 2009

© 2009 Academic Publications

§Correspondence address: Department of Physics, Faculty of Science, University Putra Malaysia, 43400 UPM, Serdang, Selangor, MALAYSIA

ered by many authors, for example see [7], [3] and [13] and the references listed therein. Nevertheless their work remained known only to the specialists in the area given the intricacy of the algorithm involved. On the other hand, scientific computational softwares such as *Mathematica* has continually improved over the years, making many difficult computations accessible to a wider audience. It is then our intent to implement the computation of Maass waveforms in *Mathematica*, which would be of interest to many physicists. We compute the eigenfunctions of the hyperbolic Laplacian and the eigenvalues for the first few Maass cusp forms on  $\mathrm{PSL}(2, \mathbb{Z}) \backslash H$ . Since there is no known analytical formula for the eigenfunction, numerical approximation is necessary. For this purpose, we develop a *Mathematica* program to compute their Fourier coefficients and the eigenvalues. In the following section we discuss some background concerning the hyperbolic plane, modular group and Maass waveforms. In Section 4 we recall the method due to [7] for computation of Maass waveforms and describe how we restructure the algorithm to locate and find the eigenvalues. We take note that Hejhal's algorithm was a huge step forward in the numerical stability and range of applicability of such computations, see [13, 12, 3, 4]. We present its numerical implementation in *Mathematica* and the method of locating the eigenvalues. In Section 5 we show some of the numerical results of the eigenvalues, plots of the eigenfunctions or the cusp forms.

## 2. Mathematical Setting

Maass waveforms are square integrable, nonconstant,  $\Gamma$ -invariant eigenfunctions of the hyperbolic Laplacian

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

defined on the Poincare upper half plane  $H = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$  and  $\Gamma$  is some specified (discrete) group. The eigenfunctions have a simple expansion in terms of modified Bessel functions in [5], p. 22, and they satisfy

$$\Delta\psi + \lambda\psi = 0, \tag{1}$$

with  $\lambda = \frac{1}{4} + r^2$ . At the same time Maass waveforms are required to satisfy the automorphy condition

$$\psi(z) = \psi(\gamma z) \text{ for all } \gamma \in \Gamma.$$

For completeness, we begin with the Poincare upper half plane  $H$  which comes with a hyperbolic metric of  $ds^2 = \frac{1}{y^2}(dx^2 + dy^2)$ . All the isometries of the

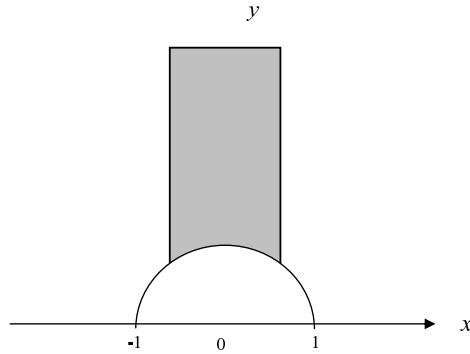


Figure 1: The fundamental domain  $F$  for the modular group

hyperbolic metric are given by the group of linear fractional transformations,

$$\text{PSL}(2, \mathbb{R}) = \left\{ z \mapsto \gamma z = \frac{az + b}{cz + d} \mid a, b, c, d \in \mathbb{R}, ad - bc = 1 \right\}.$$

The group  $\Gamma$  will be the discrete Fuchsian subgroup of  $\text{PSL}(2, \mathbb{R})$  and it realizes the needed hyperbolic surface as the quotient space  $S = \Gamma \backslash H$ . The surface  $S = \Gamma \backslash H$  can be illustrated by use of a fundamental domain, which is a connected domain  $F \subset H$  with the following properties: an arbitrary point of the hyperbolic plane is the image of a point in the interior of the fundamental domain by the action of an element of  $\Gamma$  and two different points inside the fundamental domain cannot be connected by a transformation of the group. The image of  $F$  under action of  $\Gamma$  in fact gives a perfect tiling (cover) of  $H$ . In this work, the subgroup is the modular group  $\Gamma = \text{PSL}(2, \mathbb{Z})$ , i.e.

$$\text{PSL}(2, \mathbb{Z}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} / \{\pm I\} \mid ad - bc = 1, a, b, c, d \in \mathbb{Z} \right\}.$$

Its fundamental domain is  $F = \{z = x + iy \in H; |x| < \frac{1}{2}, |z| > 1\}$ , see Figure 1 (see [1, 11]). The modular group is generated by inversion and translation, respectively given by

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} : z \mapsto -\frac{1}{z}, \quad \text{and} \quad T = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} : z \mapsto z + 1.$$

Identifying the sides of  $F$  using the maps  $S$  and  $T$  we get topologically a sphere with one cusp  $i\infty$ . The modular surface  $\Gamma \backslash H$  is a Riemann surface of constant negative curvature  $-1$  having one puncture (cusp) at  $i\infty$  with finite hyperbolic area. The group  $\text{PSL}(2, \mathbb{Z})$  acts on  $H$  without limit, which means we can use the generator  $T$  and  $S$  to pull back any  $z \in H$  to the fundamental domain

$F$  of  $\text{PSL}(2, \mathbb{Z})$ . The presence of a cusp  $i\infty$  in  $\Gamma \backslash H$  means that the Laplacian operator  $\Delta$  possesses both discrete and continuous spectra. The continuous part is spanned by Eisenstein series while the discrete part is spanned by Maass cusp forms, which vanish at the cusp. If their eigenvalues are ordered in an increasing sequence  $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \dots$ , accumulating at infinity with the number of eigenvalues counted with multiplicity, then Weyl’s law asserts that the counting function of the eigenvalue obeys

$$N_{\Gamma}(\lambda) \sim \frac{\text{Area } |F|}{4\pi} \lambda - \frac{2\kappa}{\pi} \sqrt{\lambda} \log \sqrt{\lambda} + A\sqrt{\lambda}$$

for  $\lambda \rightarrow \infty$ , where  $\kappa$  is the number of cusps, and  $A$  is a certain constant. With the identification map  $Tz = z + 1 \in \Gamma$ , we may expand any eigenfunction in a Fourier series. It is well known that discrete eigenfunctions with eigenvalues  $\lambda \geq \frac{1}{4}$  have no constant term in their Fourier expansion [5, 9] and hence

$$\psi(x + iy) = \sum_{n \in \mathbb{Z}} a_n y^{\frac{1}{2}} K_{ir}(2\pi ny) e^{(2\pi inx)}, \tag{2}$$

for some Fourier coefficients  $a_n$  and  $K_{ir}(2\pi ny)$  is the modified  $K$ -Bessel function. This Fourier series is the main tool in the computation of the eigenfunctions. In the modular group, there is an obvious reflection symmetry with respect to the  $y$ -axis which allows us to write the eigenfunction  $\psi$  as

$$\psi(x + iy) = \sum_{n \in \mathbb{Z}} a_n y^{\frac{1}{2}} K_{ir}(2\pi ny) \text{cs}(2\pi nx), \tag{3}$$

where  $\text{cs}(2\pi mx)$  is  $\cos(2\pi mx)$  for even class and  $\sin(2\pi mx)$  for the odd class.

### 3. Computing Cusp Forms

#### 3.1. The Algorithm

This subsection is to introduce the algorithm for computing the Maass cusp forms for which Hejhal’s algorithm is an essential part. The algorithm is based on Fourier expansion and the use of what we call implicit automorphy. We will write the Maass cusp form (3) for the symmetric fundamental domain as follows:

$$\psi(x + iy) = \sum_{m=1}^{\infty} a_n y^{\frac{1}{2}} K_{ir}(2\pi my) \text{cs}(2\pi mx).$$

Note that one could also have used

$$\psi(x + iy) = \sum_{\substack{m=-\infty \\ n \neq 0}}^{\infty} a_n y^{\frac{1}{2}} K_{ir}(2\pi |m| y) e^{(2\pi i m x)}.$$

Since we are interested in the groups that have symmetric fundamental domain we will use the former, while the latter form can be applied to asymmetric groups. Due to the exponential decay of the K-Bessel function for large arguments (see Section 4.2) and the bound

$$|a_n| \leq d(n)^{\frac{1}{4}},$$

for the coefficients (where  $d(n)$  counts the number of divisor of  $n$ ), we can truncate the absolutely convergent Fourier expansion any time we bound  $y$  from below and we can always find  $M = M(\varepsilon, r, y)$  such that

$$2\pi M y \geq r \quad \text{and} \quad K_{ir}(2\pi m y) \leq \varepsilon \max_x K_{ir}(x)$$

holds (see [7] and [13]). Here larger  $y$  will allow smaller  $M$ . We denote

$$[[\varepsilon]] = \sum_{M+1}^{\infty} a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) \text{cs}(2\pi m x),$$

for all terms whose absolute value is less than  $\varepsilon$ . Due to the K-Bessel function decays exponentially in  $m$ , the summand can be truncated for some large  $M$  and the remaining terms are smaller than  $\varepsilon$ , i.e.

$$\psi(z) = \sum_{m=1}^M a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) \text{cs}(2\pi m x) + [[\varepsilon]]. \tag{4}$$

The next step is to solve the above Fourier expansion for its coefficients by using a finite Fourier transform:

$$a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) = \frac{1}{2Q} \sum_{x \in X} \psi(x + iy) \text{cs}(-2\pi m x) + [[\varepsilon]], \tag{5}$$

where  $X$  is an equidistributed set of  $\mathbb{Q}$  numbers

$$X = \left\{ \frac{\frac{1}{2}}{2Q}, \frac{\frac{3}{2}}{2Q}, \dots, \frac{Q - \frac{3}{2}}{2Q}, \frac{Q - \frac{1}{2}}{2Q} \right\}.$$

The above actually performs an inverse transform over the following set of sampling points along a horocycle, with  $2Q > M + m$ :

$$2Q \geq M + M, \quad \text{and} \quad Q = \frac{M + M}{2} + 1 = M + 1.$$

For simplification, we write  $z_j = x_j + iy \mid x_j = \frac{1}{2Q}(j - \frac{1}{2}), 1 \leq j \leq Q$ .

The next step is to use the automorphy of  $\psi(z)$  under the group  $\Gamma$ ,

$$\psi(z) = \psi(z^*),$$

where  $z^*$  is the  $\Gamma$ -pullback of the point  $z$  into the fundamental domain  $D$  (see Section 2), i.e.  $z^* = \gamma z$ ,  $\gamma \in \Gamma$ ,  $z^* \in F$ . Any Maass cusp form can now be approximated by

$$\psi(x + iy) = \psi(x^* + iy^*) = \sum_{n=1}^{M_0} a_n y^{\frac{1}{2}} K_{ir}(2\pi n y^*) \text{cs}(2\pi n x^*) + [[\varepsilon]], \quad (6)$$

where we have set  $M_0 = M(y_{\min})$  and choose any  $y < y_{\min} = y_0$  in the expansion where  $y^* \geq y_{\min}$ . For the modular group  $\text{PSL}(2, \mathbb{Z})$ ,  $y_0$  is the lowest point in the fundamental domain  $F$ , i.e.  $y_0 = \min_{z \in F}(y) = \frac{\sqrt{3}}{2}$ . By a familiar flip flop through the generators of  $\Gamma$ ,  $\Gamma$ -pullback of any point into the fundamental domain make use of the inversion  $z \mapsto -\frac{1}{z}$  at least once, and possibly together with the transformation  $z \mapsto z + 1$ . It guarantees the invariance

$$\psi(z) = \psi\left(-\frac{1}{z}\right).$$

The implicit automorphy (cf. [5]) is what enables us to determine the whole Fourier series. We now use this relation and replace  $\psi(x + iy)$  in (5) with the right-hand side of (6), yielding

$$\begin{aligned} & a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) \\ &= \frac{1}{2Q} \sum_{j=1}^Q \sum_{n=1}^{M_0} a_n y^{*\frac{1}{2}} K_{ir}(2\pi n y^*) \text{cs}(2\pi n x_j^*) \text{cs}(-2\pi m x_j) + 2[[\varepsilon]], \end{aligned}$$

where  $1 \leq m \leq M$ , which is the central identity in the algorithm. Taking  $1 \leq m \leq M_0$ , and neglecting the error  $[[\varepsilon]]$  we have the set of equations

$$\sum_{n=1}^{M_0} V_{mn}(r, y) a_n = 0, \quad m \geq 1, \quad (7)$$

where the matrix  $V = (V_{mn})$  is given by

$$\begin{aligned} V_{mn}(r, y) &= a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) \delta_{mn} \\ &\quad - \frac{1}{2Q} \sum_{j=1}^Q \sum_{n=1}^{M_0} y^{*\frac{1}{2}} K_{ir}(2\pi n y^*) \text{cs}(2\pi n x_j^*) \text{cs}(-2\pi m x_j). \end{aligned} \quad (8)$$

We now have a linear system that can be used to obtain the Fourier coefficients  $a_n$  and the eigenvalues  $r$ . Note that the matrix  $V_{mn}$  can be small due to the K-Bessel function decay and this can be avoided by choosing a suitable  $y < y_0$ . Since the eigenvalues  $r$  are unknown, we discretize the  $r$ -axis and solve

for each value on the grid, the equation

$$\sum_{n=2}^{M_0} V_{mn}(r, y^{\#1}) a_n = -V_m(r, y^{\#1}), \text{ for } 2 \leq m \leq M_0. \tag{9}$$

After dropping one of the equations, we may introduce a normalization in order to avoid the trivial solution. This is done by setting for  $a_n = 1$  (see [10]). The variable  $y^{\#1} < y_0$  is chosen such that the Bessel function is not too small for  $2 \leq m \leq M_0$ . This is the final system of equations and it depends on  $y$  and  $M_0$ , but the results should be independent of the choice of these values. One may choose the following as a good value for  $y^{\#1}$ :

$$2\pi M_0 y^{\#1} = r.$$

Hejhal solves the above system repeatedly by using a different set of  $y$ , and checks whether the coefficients are independent of the choice of  $y$ . At this juncture, we can now make use of *Mathematica* in-built functions to solve the linear system of equations (equation (9)), written now as matrix equation

$$VC = Y,$$

for the Fourier coefficient matrix  $C$ .

### 3.2. Finding the Eigenvalues

The eigenvalues  $\lambda = \frac{1}{4} + r^2$  to be sought for are in the form of the values  $r$ . We compute equation (7) for an independent value of  $y$  by defining  $g_m$

$$g_m = \sum_{n=1}^{M_0} V_{mn}(r, y^{\#1}) a_n^{\#1}, \quad 1 \leq m \leq M_0,$$

where we have used the Fourier coefficients  $a_n = a_n^{\#1}$  and  $y^{\#2} = \frac{9}{10}y^{\#1}$ . It is clear from (7) that the computed  $a_n$  be the Fourier coefficients of a Maass cusp form and  $r$  is an eigenvalue only if all the  $g_m$ 's vanish simultaneously. But the probability of finding an  $r$  value such that all  $g_m$  vanished is zero, because the discrete eigenvalues are of measure zero in the real numbers. Therefore, we let  $r$  run through a grid of discretized  $r$  values and look for simultaneous changes of sign in the  $g_m$ .

It is crucial to find a suitable  $r$ -grid to minimize computer time without missing any eigenvalue. We try to minimize  $g_m$  and search for the simultaneous sign changes. Once we have found such simultaneous change in at least half of all the  $g_m$ 's, we have found a candidate interval which contains an eigenvalue  $r$  with high probability. We let the candidate interval be  $[r_{i-1}, r_i]$ . The next step

is to check whether this interval really contains an eigenvalue. If an eigenvalue is in between  $r_{i-1}$  and  $r_i$ , then

$$g_m(r_{i-1}, y) \cdot g_m(r_i, y) \leq 0$$

for almost all  $m$ . We then bisect the interval and re-examine the sign changes. We zoom in the interval with the most sign changes in  $g_m$ . If there is an eigenvalue contained in the bisected interval, the number of  $g_m$ 's that simultaneously change their sign increases from step to step in the iteration until the size of the interval approaches zero. In this event, we declare that  $r$  is an eigenvalue of the Laplacian and that our  $a_n$ 's are close to the Fourier coefficients of the corresponding Maass cusp form.

In the opposite case, the number of  $g_m$ 's which simultaneously change their sign decreases from step to step in the iteration until we suspect that there is no eigenvalue contained in the interval  $[r_{i-1}, r_i]$ . The condition of no eigenvalue can be written as follows

$$g_m(r_{i-1}, y) \cdot g_m(r_i, y) > 0,$$

for almost all  $m$ . One may need to plot  $g_m$  against  $r$ -axis to find the candidate intervals to save the computing time. The plot also may give a reasonable small interval to make our search easier faster (see Figure 2). If we assume that there are two eigenvalues of Maass cusp forms lie close together then their Fourier coefficients change rapidly and we may miss one of them. In order not to miss any eigenvalue which lies close together, at least one point of the  $r$ -grid lies between any two successive eigenvalues.

## 4. Numerical Results

### 4.1. The Parameters

As stressed in previous section, with sharp parameters  $M_0, Q, y$ , the algorithm should succeed in finding correct data for all existing cusp forms and that it should never give false cusp forms. This has been corroborated by the experiments carried out so far (see [3, 7, 13] as well as our experiments in Section 7. First, we set the parameters  $M_0, Q, y$  to find the values of  $y = y^{\#1}, y^{\#2}, Q_2$ , and  $M$ . It is very important to determine the smallest truncation point  $M = M(r, y)$  and we know that

$$2\pi My \geq r.$$



We choose the lowest point in the fundamental domain  $y_0 = \frac{\sqrt{3}}{2}$ , and solve the following

$$2\pi M_0 y_0 = r \quad \text{and} \quad 2\pi M y_0 = r.$$

Taking an independent value of  $y$  for which  $y^{\#2}$  defined earlier is a good choice, we now have the following parameters  $M, M_0, y^{\#1}, y^{\#2}$ . With  $2Q \geq M + m$ , we get  $Q = \frac{M+m}{2} + 1 = \frac{M+M}{2} + 1 = M + 1$  and the second value of  $Q_2 = \frac{9}{10}Q$ . Using similar observation as in Hejhal [5] one also can take

$$M = \frac{r + Ar^{\frac{1}{3}}}{2\pi y}$$

for some constant  $A$ . In practice it turns out that  $A = 8$  is good enough.

### 4.2. K-Bessel Test

In the computations, the accuracy of evaluation of the K-Bessel function is crucial. Here, we have tested the *Mathematica* K-Bessel function against Then's Bessel function routine written in C-language [13] but now implemented in *Mathematica*. Our test extends to the imaginary order, unlike previous tests that concern the Bessel function of integral order [14]. We found that the in-built *Mathematica* K-Bessel is limited for large order of  $r$ ; *Mathematica* failed to plot the Bessel function if  $r = 22850$ . Notice that we have to compensate the Bessel function  $K_{ir}$  with  $e^{\frac{\pi r}{2}}$  and it is working well with small  $r$  values and have the advantage of working with higher accuracies but slower computing time. On the other hand, K-Bessel function of *Then* subroutine has the accuracy of  $10^{-7}$ , but with advantage of a very wide range and slightly faster computation than the *Mathematica* built-in function. Table 1 shows example of the test out put. When implementing the algorithm outlined above in a *Mathematica* software or more generally other programs, the most time consuming task is by far, that of computing the values of the K-Bessel function  $K_{ir}(2\pi my)$ .

### 4.3. The Module

We developed two modules for even and odd separately to ease the usage of the program. In order to compute any Maass cusp we make use of the following *Mathematica* built-in functions, `BesselK[order, argument]` and `LinearSolve[ ]`. The Linear solve function is to calculate the Fourier coefficients. One can compute Maass cusp forms with wisely choosing the interval  $[r_i, r_f]$ . Large interval will result in a longer computation time. The third input for the module

r	Holger kbes routine	Mathematica K-Bessel	n	y
0.25	0.005279493017928984	0.00527952836892597	2	0.4
0.5	0.011062511961735346	0.011062513299103535	2	0.4
1	0.042285153343279036	0.04228520796401373	2	0.4
10	6.98257*10 <sup>-17</sup>	0	20	0.4
20	-0.044436983296885564	-0.044436983296885495	2	0.4
20	0.0343102	0.03431021351361238	10	0.5
50	0.389677	0.3896772381471491	10	0.5
100	0.267158	0.2671576415303035	30	0.2
500	-0.110422	-0.11042150985910876	40	0.2
1000	-0.0569238	-0.056923783169770384	50	0.2
5000	-0.0144842	-0.014484213209317818	300	0.4
10000	0.0250919	0.025091909877987786	300	0.5
20000	0.000196072	0.0001960722974269738	1000	0.5
20000	0.02631	0.0263100442621	6207	0.5
40000	-0.0117288	Indeterminate		0.4

Table 1: Comparison of the Bessel function values

r-value			
13.77975135189	31.56627541175	40.54335121050	46.48140241232
17.73856338105	32.50811775990	40.68866444930	46.65331835999
19.42348147082	32.89117021351	41.55557767357	47.42289589850
21.31579590020	34.02788420010	41.88300285420	47.92655833060
22.78590849418	34.45527153303	42.64348841466	48.03933090509
24.11235272984	35.50234977136	42.92222778356	48.74166634760
25.82624371270	35.84167643258	43.26718203880	48.99830765407
26.15208544922	36.67755494959	44.07740476166	49.68352007525
27.33270808314	37.82507229050	44.42634811860	48.99803765407
28.53074769291	38.30327615249	45.28743844310	49.68352007525
28.86339435392	39.16808496792	45.36161360214	49.96169629050
30.41067804650	39.40753186152	45.39846953130	
31.52658219679	39.77362261910	46.10145632150	

Table 2: r-Values for the even Maass cusp form for the interval I=[0,50]

will be the step of the grid  $dr$ . We call the modules `maasseven` and `maassodd` and the output will be the list of possible interval,  $g_m$  plot, and the computed eigenvalues. The accuracy can be increased by increasing the iteration in the bisection module. The input and the output of the module takes the following forms

```
In[1]:= maasseven[13.75, 13.79, 0.01]
```

r-value			
9.533695261353	28.51027770314	37.29558319925	44.29496725330
12.17300832467	29.13758755783	37.74392474203	44.77704595446
14.35850951825	29.54638812414	38.12090078958	45.11220116779
16.13807317152	29.13758755783	38.44200387442	45.68637998246
16.64425920189	30.02790484991	38.86960709783	45.78282106056
18.18091783453	30.40432705404	39.43247710882	45.95441964609
19.48471385474	31.05653396210	39.82686791840	46.56634624132
20.10669468254	31.91618247091	40.27211125778	46.83921936180
21.47905754474	32.01840643362	40.85812755625	47.17836614028
22.19467397757	32.93246556807	40.88046730299	47.54623011832
23.20139618122	33.49233128239	40.99043681909	47.82337315975
23.23611447394	33.57098962762	41.75447366271	48.14980959512
23.26371195379	34.18596993308	42.15273288130	48.35541196866
24.41971544232	34.69531040976	42.48562254312	48.98668260607
25.05085485076	35.43166464634	42.64636287866	49.10572455465
26.05691776066	35.66639637099	42.97865489379	49.43917776522
26.44699641804	35.85867349169	43.38568692359	49.99122049300
27.28438401168	36.33112904749	43.85938217734	
27.77592070179	36.98881539054	44.28210989600	

Table 3: r-Values for the odd Maass cusp form for the interval I=[0,50]

$a_n$	
1.00000000000	0.02889375399
0.51088306653	1.89136677301
1.70054792393	-0.69707687997
-0.73917113477	-0.26424696030
-1.35864012264	-1.34528249726
0.86914820432	-0.14725898511

Table 4: The Fourier coefficients for the lowest even and odd Maass cusp forms where r=47.926558330595

r=13.75 -0.0384855,0.0473493,0.0255451,0.00865618,-0.00744903,0.00716464,-0.00709113

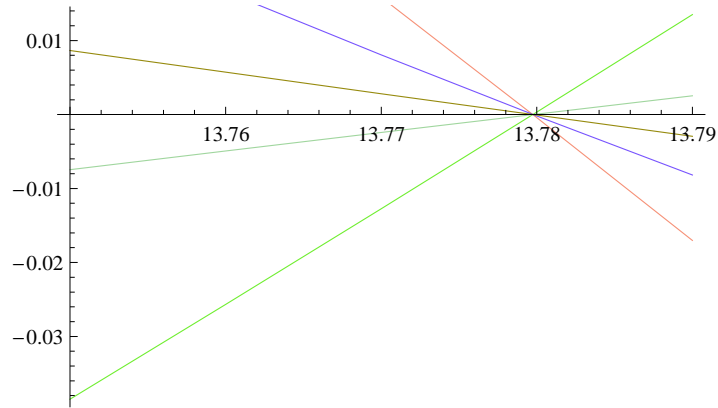
r=13.76 -0.0256765,0.0317665,0.016619,0.00572741,-0.00493001,0.00474417,-0.00469701

r=13.77 -0.0127383,0.0158517,0.00805034,0.00281748,-0.00242591,0.00233568,-0.00231323

r=13.78 0.000326361,-0.000408613,-0.000201649,-0.0000715657,0.000061639,-0.000059378,0.0000588273

Possible Interval is (13.77 13.78)

r=13.79 0.0135148,-0.017029,-0.00817429,-0.00293762,0.00253102,-0.00243



953,0. 00241776

(13.77)

Interval= 1 m= 1

25            13.7798

Interval= 1 m= 2

25            13.7798

#### 4.4. Computational Results from *Mathematica*

We have recalculated the  $r$ -values of cusp forms on the modular surface for  $0 < r < 50$ . There are two basic classes of cusp forms depending on whether the Fourier expansion is even or odd. We have found 122 eigenvalues. These results are very close with those found in the literature and with Weyl’s law prediction. For  $F = \text{PSL}(2, \mathbb{Z}) \backslash H$ , the area is  $|F| = \frac{\pi}{3}$  and with  $A = 1$ , Weyl’s law gives  $N(\lambda) = N(\frac{1}{4} + 50^2) = 122.8$ . Our computations are based on both *Mathematica* in-built K-Bessel function and the k-Bessel routine used by [13] implemented in *Mathematica*.

For a given interval  $I = [r_a, r_b]$ , we want to find all Maass cusp forms with eigenvalues  $r$  in this interval. The idea is to solve equation (9) and try to find  $r \in I$ . In practice we first divide the interval  $I$  into a number of equally small chunks:

$$r_a = r_0 < \dots < r_i < \dots < r_N = r_b .$$

Then we compute  $g_m$  and look for sign changes. We plot  $g_m$  and zoom in in the intervals that have the most changes in sign for the  $g_m$ ’s. For our experiment

we search for the eigenvalues in the interval  $I = [0, 50]$ . We run the developed program for  $r$ -grid with step size of  $dr = 5e^{-5}$ . One may use Weyl's law as a starting point to find an estimate of average distance between the eigenvalues. But the real computation required increasingly finer grids until no other candidate eigenvalues are found. All  $g_m$  will then be plotted against  $r$  and the intervals where most of the  $g_m$ 's intersect are sought for. These intervals will then be zoomed-in with a finer grid (see for example Figure 2). We can also reset the step size if we suspect any missing eigenvalues and plot the  $g_m$ 's again until the output interval is very small; for example see Figure 3. Now we are ready to proceed to the next stage of using the bisection together with "zooming in" of the intervals and re-checking the sign changes. We take note instances when the  $g_m$ 's that changed their signs increase with the iterations until the size of the interval approaches zero and the eigenvalues  $r$  are considered to have been found (see Figure 3). We present the obtained eigenvalues which are divided into even and odd classes in Tables 2 and 3. We have checked the accuracy of our results with the aid of the multiplicative relation of the Fourier coefficients where the first few coefficients satisfy  $a_{mn} = a_m \cdot a_n$ . Also the accuracy can be checked by computing the coefficients a second time with an independent different value of  $y$ . All Fourier coefficients satisfy the Ramanujan-Petersson conjecture, i.e.

$$|a_p| \leq 2 \text{ for all primes } p.$$

In Table 4, we list the first few Fourier coefficients of the Maass cusp form corresponding to  $r = 47.926558330595$ , and Figure 4 shows a picture of the Maass cusp form corresponding to the eigenvalue. The computing time to locate the eigenvalues ranges from few minutes to hours depending on how large the eigenvalue we are searching for. On the other hand plotting Maass cusp form picture usually takes much longer time. For example for  $r = 47.926558330595$ , the figure took a CPU time of 11 hours and 59 minutes on a CPU of dual processor of 1.7 GHz and 1GB of memory. Figure 5 shows the nodal line for the lowest odd and even cusp forms while Figure 6 shows the picture of the odd and even Maass cusp form for the lowest third eigenvalues. Figures 4, 5 and 6 show good agreement with some earlier work; see for example [8].

## 5. Conclusion

We have successfully computed Maass cusp forms for the modular surface and their eigenvalues using *Mathematica* and our results are equivalent to those

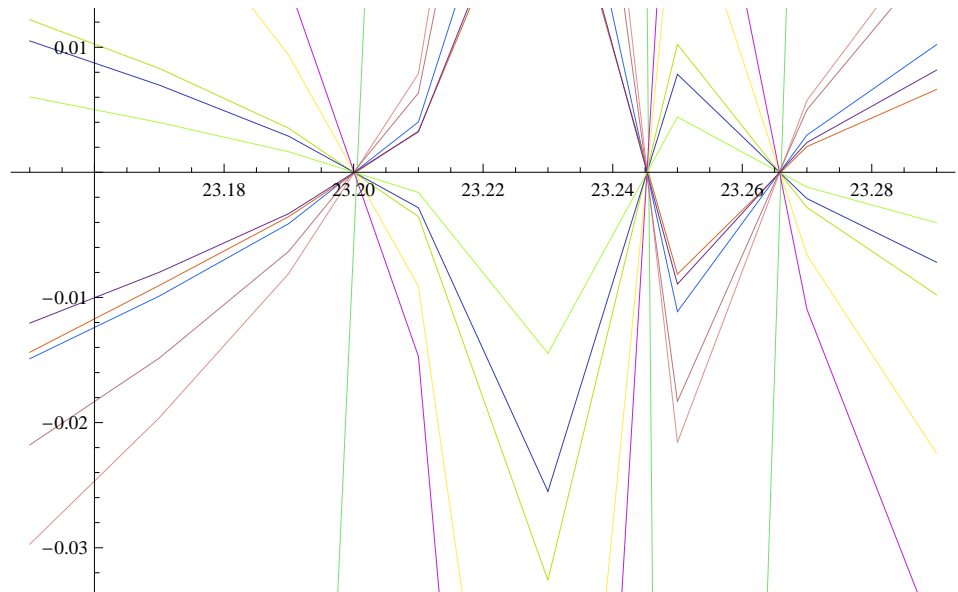


Figure 2: Plot of  $g_m$ 's shows the small steps size for the grid and close eigenvalues

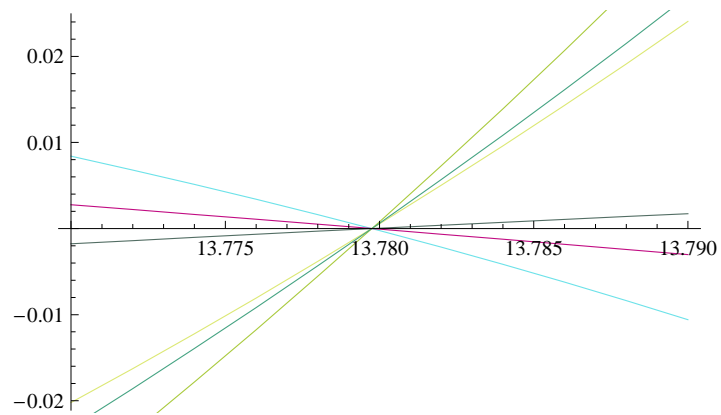


Figure 3: Plot of  $g_m$ 's shows the small steps size and that interval is small to search for the eigenvalue

found in the literature (see [2, 7]). We highlight the intention of using a commercially available software, i.e. *Mathematica* primarily for its wider accessi-

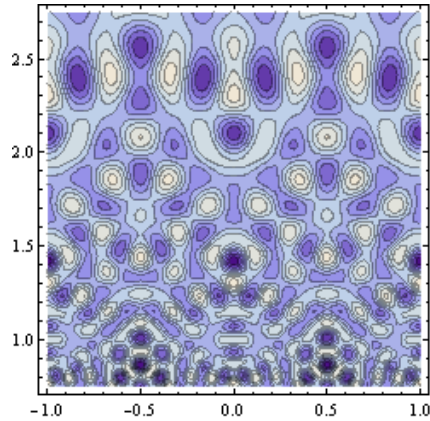


Figure 4: Maass cusp form for  $r = 47.926558330595$  in the region  $I = [-1, 1] \times [2.5, 2.5]$

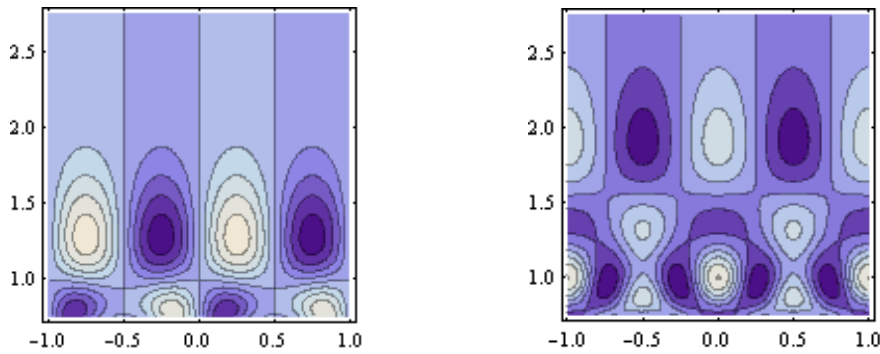


Figure 5: Nodal lines for the odd cusp form for  $r = 9.533695261353$  and for the even cusp form for  $r = 13.779751351890$ . The illustrated region is  $[-1, 1] \times [0, 2.5]$

bility and not necessarily for effectiveness of computations at all ranges. From our computations, we amply demonstrate that the chosen software can be used readily for the complex computations of Maass cusp forms. The computations for the modular surface also serves as a basis for further development of computations of cusp forms on other surfaces, which are currently in progress.

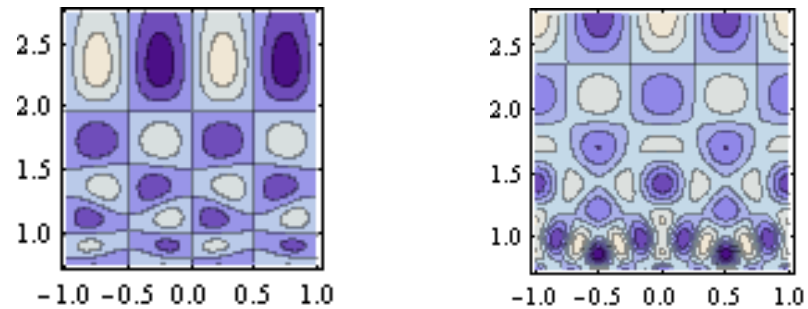


Figure 6: Plot of the Maass cusp forms, the left picture corresponds to  $r = 14.35850951825$  and the right picture corresponds to  $r=19.42348147082$

### Acknowledgements

This work is supported by the Malaysian Ministry of Science, Technology and Innovations (MOSTI), under the project no. 06-01-04-SF0256. We would like to thank the Institute of Advanced Technology, Universiti Putra Malaysia where the project is initiated. We are most grateful to Holger Then for his friendly advice and valuable discussions in carrying out this numerical work.

### References

- [1] T.M. Apostol, *Modular Functions and Dirichlet Series in Number Theory*, Springer-Verlag, New York (1976).
- [2] H. Avelin, *On the Deformation of Cusp Forms*, Licentiate Thesis, University of Uppsala, Sweden (2003).
- [3] A. Booker, A. Strömbergsson, A. Venkatesh, Effective computations of Maass cusp forms, *Int. Math. Res. Not.* (2006), 34.
- [4] D.W. Farmer, S. Lemurell, Deformations of Maass forms, *Math. Comp.*, **74** (2005), 1967-1982.
- [5] D.A. Hejhal, *The Selberg Trace Formula for  $PSL(2; R)$* , Lecture Notes in Mathematics, Volume 2, Springer-Verlag, Berlin (1983).



- [6] D.A. Hejhal, Eigenvalues of the Laplacian for Hecke triangle groups, *Mem. Amer. Math. Soc.*, **97**, No. 469 (1992).
- [7] D.A. Hejhal, On eigenfunctions of the Laplacian for Hecke triangle groups, In: *Emerging Applications of Number Theory*, Minneapolis, MN (1996); *IMA – Math. Appl.*, **109**, Springer, New York (1999), 291-315.
- [8] D.A. Hejhal, B.N. Rackner, On the topography of Maass waveforms for  $PSL(2, \mathbb{Z})$ , *Exp. Math.*, **1**, No. 4 (1992), 275-305.
- [9] H. Iwaniec, *Introduction to the Spectral Theory of Automorphic Forms*, Biblioteca de la Revista Matematica Iberoamericana (Library of the Revista Matematica Iberoamericana), Revista Matematica Iberoamericana, Madrid (1995).
- [10] T. Miyake, *Modular Forms*, Springer-Verlag (1989).
- [11] R.A. Rankin, *Modular Forms and Functions*, Cambridge University Press, Cambridge-New York-Melbourne (1977).
- [12] F. Stromberg, *Computational Aspects of Maass Waveforms*, Thesis, Uppsala Universitet (2005).
- [13] H. Then, Maass cusp forms for large eigenvalues, *Math. Comp.*, **74**, No. 249 (2005), 363-381.
- [14] H.A. Yousif, R. Melka, Computing Bessel functions of the second kind in extreme parameters regimes, *Computer Physics Communications*, **151** (2003), 25-34.

