Sains Malaysiana 42(5)(2013): 655–660

Computation and Visualization of Cuspidal Waveforms for Modular Group Using GridMathematica

(Pengiraan dan Visualisasi Gelombang Berjuring untuk Kumpulan Modul dengan Menggunakan GridMathematica)

CHAN KAR TIM*, HISHAMUDDIN ZAINUDDIN & SAEID MOLLADAVOUDI

ABSTRACT

Spectral studies on the eigenfunctions of Laplace-Beltrami operator on a cusp manifold are known to contain both discrete and continuous eigenvalues. The discrete eigenfunctions are usually called Maass cusp forms where their eigenvalues are not known analytically. The aims of this report were to compute the eigenvalues $\lambda = r^2 + 1/4$ *for the modular group, PSL(2,***Z***) numerically and visualize the waveforms using GridMathematica. At the same time, we compared the performance of parallel programming (GridMathematica) and normal programming (Mathematica). This serves to show the feasibility and advantages of using the parallel version of commercially available software for complex computations of Maass cusp forms. In our computer search for 33 eigenvalues in the r-interval [9, 30.4], we found that the performance of the parallel programme is about six times faster than the normal programme.*

Keywords: GridMathematica; Maass cusp forms; modular group

ABSTRAK

Kajian spektrum pada fungsi eigen operator Laplace-Beltrami di atas permukaan berjuring diketahui mempunyai nilai eigen yang diskrit dan selanjar. Fungsi eigen diskrit biasanya dikenali sebagai fungsi bentuk juring Maass dengan nilai eigennya tidak diketahui secara analisis. Tujuan kertas ini adalah untuk mengira nilai eigen $\lambda = r^2 + 1/4$ *bagi kumpulan modul, PSL(2,***Z***) secara berangka dan menggambarkan gelombangnya dengan menggunakan GridMathematica. Pada masa yang sama, kami juga membandingkan prestasi pengaturcaraan selari (GridMatematica) dengan pengaturcaraan biasa (Mathematica). Ini bertujuan untuk menunjukkan kebolehlaksanaan dan kelebihan menggunakan perisian komersial versi selari untuk pengiraan kompleks fungsi bentuk juring Maass. Dalam carian komputer untuk 33 nilai eigen dalam selang-r [9, 30.4], didapati bahawa prestasi pengaturcaraan selari adalah lebih kurang enam kali ganda lebih laju daripada pengaturcaraan biasa.*

Kata kunci: Fungsi bentuk juring Maass; GridMathematica; kumpulan modul

INTRODUCTION

Computation of eigenfunctions of Laplace-Beltrami operators on cusp manifold has been studied in both physics and mathematics due to their spectra that being both discrete and continuous. Well-studied cusp manifolds are the hyperbolic surfaces H / Γ formed from the quotient of the upper half plane H equipped with the hyperbolic metric $ds^2 = y^{-2}(dx^2 + dy^2)$ by a discrete group of $PSL(2, \mathbf{R})$. One of the reasons that such systems are of interest is that classically they represent chaotic systems (Gutzwiller 1990) whose quantum properties are yet to be properly understood.

Our report was based on the well known modular surface $PSL(2, \mathbf{Z}) \setminus H$. Particularly we focused on the Maass cusp forms (MCF) on this surface which are eigenfunctions (the quantum bound states) of the discrete spectra. Computational work of MCF on $PSL(2, \mathbf{Z}) \setminus$ H is not new and has been done by researchers such as Hejhal and Rackner (1992), Stromberg (2005) and Then (2005). Their earlier numerical programmes are based on Cray-FORTRAN (Hejhal & Rackner 1992), FORTRAN (Stromberg 2005) and C programming (Then 2005). These numerical programmes while efficient are often complex and hard to understand for beginners.

Following this, Siddig and Zainuddin (2009) used a commercial software Mathematica for the computation of Maass waveforms. Mathematica has a wider user base and are easily accessible to beginners. They implemented Hejhal and Then's algorithm in Mathematica and developed two modules for finding even and odd eigenvalues separately. An advantage of using Mathematica is that there are many built-in functions including the K-Bessel function, BesselK[order, argument]. This is one of the main components and among the most time consuming part in computing eigenvalues. They also made a comparison between the built in K-Bessel function and the K-Bessel developed by Then (2005). They found out that both gave the same results to justify Mathematica usage for its ease. The only drawbacks of the built-in function are it is more time consuming and has limited range. At present, to the

best of our knowledge, there are only one Mathematica package that handles specifically the MCF computations and particularly there is none using GridMathematica. GridMathematica is a parallel version of Mathematica which is realizable on a cluster of workstations. This is particularly handy if one is interested in computing higher ranges of eigenvalues for the modular surface or eigenfunctions for more complex surfaces. It is in this view that this work is done; namely to facilitate future calculations by testing the programme on a standard example.

Materials and Methods

The Algorithm for Computation

An algorithm for computing MCF is built based on Hejhal's algorithm (Hejhal & Rackner 1992) and that of Then (2005). For completeness, we reproduce here the outlines of the algorithm. The algorithm is based on Fourier expansions and the use of implicit automorphy under the action of the modular group. The Fourier expansion of the MCF for the symmetrical fundamental domain is written as (Then 2005):

$$\psi(x+iy) = \sum_{n=1}^{\infty} a_n y^{1/2} K_{ir}(2\pi n y) cs(2\pi n x), \tag{1}$$

where $cs(x)$ represents $2\cos(x)$ for even Maass waveforms and $2\sin(x)$ for odd ones, a_n is the Fourier coefficient and $K_{ir}(x)$ is the modified Bessel function of the second kind whose order is connected with the eigenvalue λ by $\lambda = r^2 + 1/4$. It is to be noted here that λ is the true eigenvalue but then hereafter we will call r to be eigenvalue instead as commonly used in the literature (Then 2005). We then truncate the Fourier expansion to get:

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

where $M_0 = M(y_0)$ is the adopted truncation of number of terms.

Implementing the automorphy of $\psi(z)$ under the group $G = PSL(2, \mathbf{Z})$, we write $\psi(z) = \psi(z^*)$. Point z^* is the pullback of the point z into the fundamental domain of the modular surface. Applying this condition on (2), we will have:

$$\begin{aligned}\psi(x+iy) &= \psi(x^* + iy^*) \\ &= \sum_{n=1}^{M_0} a_n y^{*\frac{1}{2}} K_{ir}(2\pi n y^*) cs(2\pi n x^*) + [[\varepsilon]].\end{aligned} \tag{3}$$

Next, we solve (3) by using finite Fourier transform to give:

$$\begin{aligned}a_m y^{\frac{1}{2}} K_{ir}(2\pi m y) \;=\; & \frac{1}{2Q} \sum_{j=1}^{Q} \sum_{n=1}^{M_0} y^{*\frac{1}{2}} K_{ir}(2\pi n y^*) \\ & cs(2\pi n x_j^*) cs(-2\pi m x_j) + 2[[\varepsilon]],\end{aligned} \tag{4}$$

where Q is the selected number of equidistributed sampling points. Neglecting the error $2[[\varepsilon]]$ and taking $1 \le m \le M_0$, we have the system of equations:

$$\sum_{n=1}^{M_0} V_{mn}(r, y) a_n = 0, \quad m \ge 1, \tag{5}$$

where the matrix V_{mn} is given by:

$$\begin{aligned}V_{mn}(r, y) &= y^{\frac{1}{2}} K_{ir}(2\pi n y) \delta_{mn} \\ &- \frac{1}{2Q} \sum_{j=1}^{Q} \sum_{n=1}^{M_0} y^{*\frac{1}{2}} K_{ir}(2\pi n y^*) cs(2\pi n x_j^*) cs(-2\pi m x_j).\end{aligned} \tag{6}$$

We may introduce normalization in order to avoid the trivial solution by setting $a_1 = 1$ (Miyake 1989) to give:

$$\sum_{n=2}^{M_0} V_{mn}(r, y) a_n = -V_m(r, y), \text{ for } 2 \le m \le M_0. \tag{7}$$

By solving (7), we will get the Fourier coefficients. Details of the algorithm can be found in Booker et al (2006), Hejhal and Rackner (1992), Siddig and Zainuddin (2009) and Then (2005).

Note that the coefficients in (7) should be independent of y. To avoid repeated solving of (7) for such check, we employ a method described by Then (2005), that computes:

$$g_m = \sum_{n=1}^{M_0} V_{mn}(r, y^{\#2}) a_n^{\#1} \quad \text{for } 1 \le m \le M_0, \tag{8}$$

where $y^{\#2} = 0.9 y^{\#1}$. We then look for simultaneous changes of sign in g_m to determine the candidate interval for the eigenvalues.

GridMathematica Implementation

The computational work in this report is based on the programmes developed by Siddig and Zainuddin (2009) whose intent was to demonstrate that available commercial software Mathematica is capable to do the MCF computations. We further demonstrate here that use of GridMathematica or its parallel version can help make the computations less time-consuming. In this work, the GridMathematica is installed on a cluster of workstations which consist of 7 nodes (Dual processors of 1.7 GHz and 1 GB of memory) linked together by Rocks Cluster (version 5.2, Linux distribution). The earlier Mathematica programmes (Siddig & Zainuddin 2009) are optimised and modified so that they are workable in parallel computing environment.

In optimization, we modify the bisection routine which is used to find the root of a candidate interval to a certain point of accuracy. We limit our programme to run the g_m module, g[r,m], only once for a single interval. We use only g_m value for m=1 as this will give us a more accurate eigenvalue when compared with the other g_m values. Table 1 shows the performance of the maassodd module before and after optimization process. This way of optimization saved at least 79% of computing time. Next, we also

TABLE 1. Performance of maassodd module before and after optimization

Tasks	Maassodd module		Time saved	
	Computing time before optimization, s	Computing time after optimization, s	Second, s	Percentage, %
[9.532, 9.5341, 0.0003]	321.7	65.2	256.5	79.7
[12.172, 12.1741, 0.0003]	792.3	141.1	651.2	82.2
[14.358, 14.3601, 0.0003]	1745.0	361.6	1383.4	79.3
Total computing time	2859.0	567.9	2291.1	80.1

minimize the use of the g_m plot to find the candidate interval of eigenvalues. We found out that some fake intervals will eventually disappear when we rerun the candidate interval in a smaller step size.

For the programme to work in parallel computing environment (using star network topology), we need to distribute all functions in the programme to all available nodes on the cluster workstation using command Distri buteDefinitions[definition]. Then, we use the automatic parallelization command Parallelize[expression] to send all tasks to different nodes for evaluation. This command will automatically divide the number of tasks between each kernel before the evaluation starts. Let's take interval [9, 30.4] for example. Within our programme, we have Fourier coefficient module, f[r], pullback module, pulz[z], g_m module, g[r,m], precision module, prec[r] and maassodd module, maassodd[r_i, r_f ,dr]. Initially, we distribute them to other kernels using command written in this form:

DistributeDefinitions[f, pulz, g, prec, maassodd]

Next, to parallelize a list of task, the command is written in this form:

Parallelize[{maassodd [9,20, 0.1],
maassodd[19.9, 25, 0.1],
maassodd[24.9,30.4,0.1]}]

The number of tasks and the way of setting the interval are all dependent on the user. There are several other parallel computing commands besides Parallelize. Since our programme is written in the form of modules, parallelizing using command Parallelize is considered the best way.

After computing the eigenvalues, we check the corresponding Fourier coefficients to verify the authenticity of the eigenvalues. All Fourier coefficients must satisfy the Ramanujan-Petersson conjecture (Then 2005), i.e. $| a_p | \leq 2$ for all primes p. As for the visualization, we simply exploit the contour plot and density plot functions of GridMathematica.

RESULTS AND DISCUSSION

We have developed two modified and optimised modules for finding even and odd eigenvalues in GridMathematica. For a better successive rate of finding all the correct eigenvalues, we have to set the parameters M_0, Q and y sharply. One can run with different parameters to find the best results. We set M_0 according to Hejhal's observation (Hejhal & Rackner 1992):

$$M_0 = \frac{r + Ar^{1/3}}{2\pi y_0} \tag{9}$$

for some constant A. In practice, it turns out that A=8 is good enough (Siddig & Zainuddin 2009). For y, it must be $y \leq y_0 = \sqrt{3}/2$, thus we set it as $y = 0.5$ and $Q = M_0 + 1$. We run our modules for interval [9, 30.4] with different step sizes (0.1, 0.0336, 0.002 and 0.0003). These step sizes are chosen because the eigenvalues are distributed exponentially. This process eventually will provide a list of candidate intervals. The accuracy can be increased further by using the bisection module.

We have found 33 eigenvalues (Table 2) for interval [9, 30.4] and these results are very close to the Weyl's law prediction (Risager 2004). For $PSL(2, \mathbf{Z}) \setminus H$, area $|F| = \pi/3$, Weyl's law gives $N(\lambda) = N(0.25+30.4^2) = 34.6$. Besides, all Fourier coefficients of these eigenvalues also satisfy the Ramanujan-Petersson conjecture.

The computing times to locate the eigenvalues for both modules for interval [9, 30.4] are shown in Table 3. These results show that the performance of GridMathematica is at least 6 times faster than Mathematica. Figure 1 shows the parallel processes operating at 13 out of 14 kernels. The more kernels we use, the more computing time can be saved because more tasks can be evaluated simultaneously. One kernel is reserved mainly for the use of data gathering and some local Linux and GridMathematica processes.

Visualization of the MCF is shown in Figure 2. Figures 2(a) and 2(c) show the contour plot for r =29.546388124 and r=28.863394353, respectively. When the MCF is even, there are no nodal lines crossing the boundary of the fundamental domain. Nodal lines crossing each other only happen when the MCF is odd. Nodal lines are actually the points where $\psi = 0$. The result of the nodal lines agrees well with the results obtained by Hejhal and Rackner (1992). Figure 2(b) and 2(d) show the density plots for r =29.546388124 and r =28.863394353, respectively. These plots show us the probability function associated to ψ (i.e. $|\psi|^2$). The brighter regions show where higher probability distribution of the quantum particle can be

TABLE 2. *r*-values for the odd and even Maass cusp form for the interval [9, 30.4]

Maasseven r-values		Maasseven *r*-values
9.533695261	23.263711537	13.779751351
12.17300832	24.41971544	17.738563381
14.358509518	25.050854850	19.423481470
16.138073171	26.05691776	21.315795940
16.64425920	26.4469964	22.78590849
18.180917834	27.28438401	24.112352729
19.484713859	27.775920701	25.826243712
20.106694682	28.5102777	26.152085449
21.479057544	29.137587557	27.3327080
22.19467397	29.546388124	28.53074769
23.201396181	30.279048	28.863394353

TABLE 3. Computing time for parallel and normal processes for odd and even module

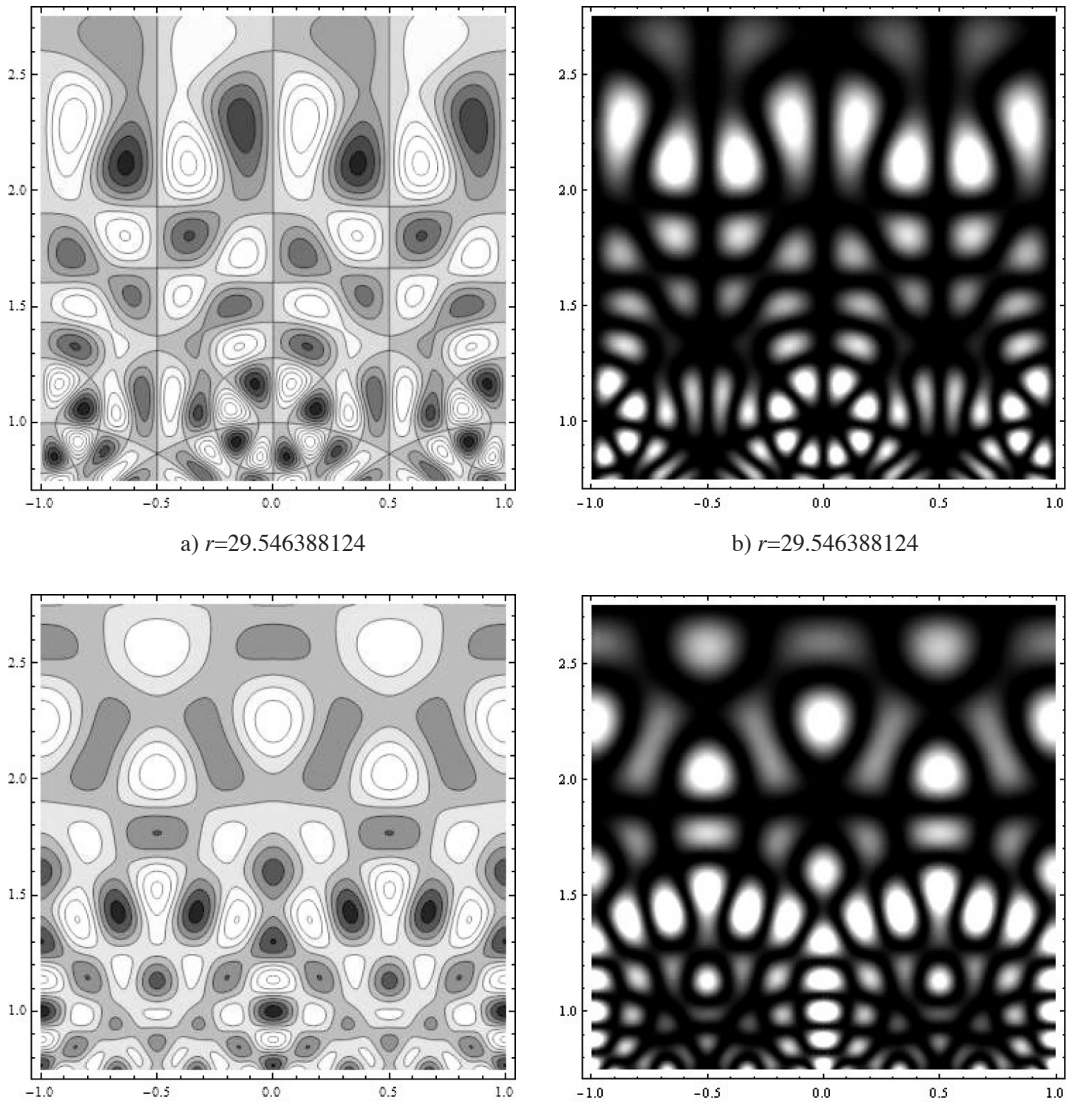
Interval [9,30.4]	Step 0.1	0.0336	0.002	0.0003	Bisection	Total time
Parallel Odd	7574.8 s (2.10 h)	4577.2 s (1.27 h)	30850.9 s (8.57 h)	28776.7 s (7.99 h)	12328.0 s (3.42 h)	84107.6 s (23.36 h)
Normal Odd	57898.4 s (16.08 h)	31119.6 s (8.64 h)	149335.5 s (41.48 h)	152450.2 s (42.35 h)	103601.8 s (28,78 h)	494405.5 s (137.33 h)
Parallel Even	7815.3 s (2.17 h)	3259.7 s (0.91 h)	14771.9 s (4.10 h)	15150.8 s (4.21 h)	9951.9 s (2.76 h)	50949.6 s (14.15 h)
Normal Even	57862.1 s (16.07 h)	17739.0 s (4.93 h)	70511.5 s (19.59 h)	81928.8 s (22.76 h)	55480.0 s (15.41 h)	283521.4 s (78.76 h)

Parallel Kernel Status

13 kernels running, idle

ID	Name	Host	Process	CPU	RAM	Version	Close	Act	Time	Elapsed: 12328.017s, sp
0	master	qcluster	8866	532.429	15M	7.0			528.877	
1	local	qcluster	9525	10422.907	7M	7.0	x		10422.700	
2	10.1.1.253	compute-0-1	7635	4227.411	7M	7.0	x		4227.210	
3	10.1.1.253	compute-0-1	7671	9775.337	7M	7.0	x		9775.130	
4	10.1.1.252	compute-0-2	7607	9937.712	7M	7.0	x		9937.500	
5	10.1.1.252	compute-0-2	7661	5860.568	7M	7.0	x		5860.370	
6	10.1.1.251	compute-0-3	7635	8766.439	7M	7.0	x		8766.240	
7	10.1.1.251	compute-0-3	7660	12323.780	7M	7.0	x		12323.600	
8	10.1.1.250	compute-0-4	7590	7017.266	7M	7.0	x		7017.060	
9	10.1.1.250	compute-0-4	7626	10145.113	7M	7.0	x		10144.900	
10	10.1.1.249	compute-0-5	7596	11234.409	7M	7.0	x		11234.200	
11	10.1.1.249	compute-0-5	7683	3591.369	7M	7.0	x		3591.170	
12	10.1.1.248	compute-0-6	7633	6878.546	7M	7.0	x		6878.340	
13	10.1.1.248	compute-0-6	7668	3229.237	7M	7.0	x		3229.040	

FIGURE 1. Parallel processes



a) *r*=29.546388124 b) *r*=29.546388124

c) r=28.863394353 d) r=28.863394353

FIGURE 2. Contour plot (a and c) and density plot (b and d) for *r*=29.546388124 and *r*=28.863394353, respectively. Figure (a) and (b) are for the odd waveforms while (c) and (d) are for the even waveforms

at the given energy level (eigenvalue). Plotting times for both eigenvalues using GridMathematica takes around 28 h while using Mathematica, it takes more than 3 days.

CONCLUSION

We have successfully computed and visualized the eigenvalues for modular group using GridMathematica. Performance of the parallel programming is about six times faster than the normal programming. Besides, we have also managed to visualize the MCF in a density plot giving the probability distribution of the quantum particle at the given energy levels. The computation for the modular group using GridMathematica will serve as the basis for further optimization and development of computation for cuspidal waveforms for higher energy eigenvalues as well as cuspidal waveforms on more complex surfaces.

ACKNOWLEDGEMENTS

This work was supported by the Malaysian Ministry of Higher Education, under fundamental research grant scheme, project number 01-10-07-286FR and 01-04-10-687FR.

REFERENCES

Booker, A.R., Strombergsson, A. & Venkatesh, A. 2006. Effective computation of Maass cusp forms. *International Mathematics Research Notices IMRN* pp. 1-34. doi: 10.1155/1MRN/2006/71281.

Gutzwiller, M.C. 1990. *Chaos in Classical and Quantum Mechanics*. New York: Springer-Verlag.

Hejhal, D.A. & Rackner, B.N. 1992. On the topography of Maass waveforms for PSL(2,Z). *Experimental Mathematics* 1(4): 275-305.

Miyake, T. 1989. *Modular Forms*. Berlin: Springer-Verlag.

Risager, M.S. 2004. Asymptotic densities of Maass newforms. *Journey of Number Theory* 109: 96-119.
Siddig, A.A.M. & Zainuddin, H. 2009. Computation of Maass cusp forms on modular group in Mathematica. *International Journal of Pure and Applied Mathematics* 54(2): 279-295.
Stromberg, F. 2005. Computational aspects of Maass waveforms, Ph.D. Thesis. University of Uppsala, Sweden (unpublished).
Then, H. 2005. Maass cusp forms for large eigenvalues. *Mathematics of Computation* 74(249): 368-381.

Chan Kar Tim* & Hishamuddin Zainuddin
Department of Physics
Faculty of Science
Universiti Putra Malaysia
43400 Serdang, Selangor
Malaysia

Saeid Molladavoudi & Hishamuddin Zainuddin
Laboratory of Computational Sciences and Mathematical Physics
Institute for Mathematical Research
Universiti Putra Malaysia
43400 Serdang, Selangor
Malaysia

*Corresponding author; email: ckartim3371@gmail.com

Received: 9 March 2012
Accepted: 20 October 2012