Computing a database of rigorous Maass forms

David Lowry-Duda

Abstract. We announce a database of rigorously computed Maass forms on congruence subgroups $\Gamma_0(N)$ and briefly describe the methods of computation.

1. Computation of Maass Forms

Let $\Gamma_0(N)$ be a congruence subgroup in $\mathrm{SL}(2,\mathbb{Z})$. Each matrix $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}(2,\mathbb{R})$ acts on the upper halfplane $\mathcal{H}$ in the standard way, given by $\gamma z = \frac{az+b}{cz+d}$. The quotient space $\Gamma_0(N)\backslash\mathcal{H}$ is a noncompact Riemann surface. In the hyperbolic metric on $\mathcal{H}$, the weight k Laplace-Beltrami operator, Δ_k, takes the form

$$\Delta_k = y^2\Big(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\Big) - iky\frac{\partial}{\partial x}.$$

Let χ denote a Dirichlet character of modulus N. Then Maass cuspforms of weight k and character χ on $\Gamma_0(N)$ are eigenfunctions f of Δ_k that satisfy

- $\Delta_k f + \lambda f = 0$,
- $f(\gamma z) = \chi(\gamma)(cz+d)^k f(z)$ for all $\gamma = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \Gamma_0(N)$, and
- $f \in L^2(\Gamma_0(N)\backslash\mathcal{H})$.

Maass cuspforms form the discrete component of the spectral resolution of Δ_k and therefore form building blocks for all $L^2(\Gamma_0(N)\backslash\mathcal{H})$. The problem of giving numerical examples of Maass cuspforms has been widely considered since the 1970s. Despite this, it is difficult to give any nontrivial example of a Maass cuspform.

Each Maass form of weight k and character χ on $\Gamma_0(N)$ has a Fourier expansion of the form

$$f(z) = \sum_{n\neq 0} a(n)\sqrt{y}K_{ir}(2\pi|n|y)e^{2\pi i n x} \tag{1}$$

where $z = x+iy$, $K_\nu(y)$ is the modified Bessel function of the second kind, and the eigenvalue λ of f is given by $\lambda = \frac{1}{4} + r^2$ for $r \in \mathbb{R}$. This r is called the *spectral parameter* of f. To describe a Maass form, it's necessary to describe the eigenvalue λ (or equivalently the spectral parameter r) and the coefficients $a(n)$. In practice, most data associated to a generic Maass form is conjectured to be transcendental and algebraically independent of standard constants; hence one gives approximations to the eigenvalue and coefficients to specify a Maass form.

2020 *Mathematics Subject Classification.* Primary 11-04, 11Y99; Secondary 11F03, 11-11.

This work was supported by the Simons Collaboration in Arithmetic Geometry, Number Theory, and Computation via the Simons Foundation grant 546235.

20 Jun 2025 20:16:01 PDT
250131-LowryDuda Version 2 - Submitted to LuCaNT

In [**The05**], H. Then gives an extensive list of references to earlier computations of Maass forms. Until 2006, methods to generate numerical examples were fundamentally *heuristic*, meaning that there is no guarantee that a claimed approximation is actually close to a true Maass form.

Booker, Strömbergsson and Venkatesh [**BSV06**] gave the first method of computing *rigorous* Maass forms on $\mathrm{SL}(2,\mathbb{Z})$. Stated more precisely, they gave a method that gives explicit intervals containing the eigenvalue and coefficients, and hence bounds for how close a putative Maass form is to a true Maass form. There are now three algorithms to generate rigorous Maass cuspforms:

(1) *Quasimode construction*, a generalization of [**BSV06**] to general level and character by Child [**Chi22**].
(2) A rigorous implementation of the *Selberg trace formula* due to Seymour-Howell [**SH22**],
(3) A rigorous version of *Hejhal's algorithm* due to Seymour-Howell and Lowry-Duda, to be described in forthcoming work [**LDSH**].

It is notable that Booker (and to a lesser extent, Strömbergsson) helped develop each of these algorithms.

In this note, we describe how an initial database of 35416 rigorous Maass cuspforms (all of weight 0, on congruence subgroups $\Gamma_0(N)$ with N squarefree) was computed using these algorithms and inserted into the LMFDB [**LMF19**]. We briefly describe the three algorithms and how they interact in §2. In §3, we detail the data now available. And in §4, we give additional comments on the current and future status of this database.

2. Algorithms for Rigorously Computing Maass Forms

We say that a Maass cuspform f_j is computed rigorously if there is an explicit interval $I_{j,\lambda}$ for the eigenvalue and explicit intervals $I_{j,n}$ for the coefficients such that there is exactly one true Maass form

$$f_j(z) = \sum_{n\neq 0} a_j(n)\sqrt{y}K_{ir_j}(2\pi|n|y)e^{2\pi i n x}$$

with $\lambda_j = \frac{1}{4} + r_j^2 \in I_{j,\lambda}$ and $a_j(n) \in I_{j,n}$ for all n. In this note, we impose the additional restriction that there can be no missing eigenvalues, or rather that every eigenvalue up to λ_j must also have been produced.

We now briefly describe each of the three methods of computation.

Remark 1. A folklore conjecture states that the vector space of Maass forms on $\mathrm{SL}(2,\mathbb{Z})$ with a given eigenvalue is one-dimensional. This would imply that a sufficiently tight interval $I_{j,\lambda}$ would be sufficient to uniquely specify a Maass form on $\mathrm{SL}(2,\mathbb{Z})$. This is true for all of the Maass forms in the initial database. In general, however, there are spaces with nontrivial character in which multiple Maass forms have the same eigenvalue. In this case, both $I_{j,\lambda}$ and a finite number of $I_{j,n}$ are necessary to uniquely identify the form f_j.

2.1. Rigorous Trace Formula. With extreme simplification, the Selberg trace formula in [**SH22**, Theorem 3.1] describes sums over coefficients weighted by "nice" analytic test functions F, taking the form

$$\sum_{j>0} F(r_j)a_j(n) = \text{(complicated but explicit)}.$$

This sum is over all Maass forms f_j on $\Gamma_0(N)$. The omitted right hand side is in terms of data associated to conjugacy classes of the group $\Gamma_0(N)$ and the Fourier transform $\widehat{F}$.

One significant difficulty in this approach comes from the hyperbolic contribution. These terms include special values $L(1, (d/\cdot))$ for d of the form $(t^2 - 4n)$ for $t \in \mathbb{Z}$, and hence involve computing class numbers and regulators of quadratic fields by the class number formula. Choosing test functions F with the property that $\widehat{F}$ has compact support allows one to compute data for only finitely many quadratic fields. By carefully combining different choices of these test functions, it's possible to isolate intervals for individual eigenvalues and coefficients.

Remark 2. This relationship between class numbers and Maass forms can also be used in reverse. With ample Maass form data, one can compute *more* class numbers. See [**BBD$^+$24**] for more in this direction.

The most important benefit of this algorithm is that it can *guarantee that all Maass forms have been found* in an eigenvalue range. In practice, one can try to use the trace formula to give initial approximations to Maass forms and use other methods to refine these approximations.

2.2. Quasimode construction. In [**Chi22**], Child extends the quasimode construction technique in [**BSV06**] to general level. If $\widetilde{f}_j$ is a putative Maass form and $(\Delta - \lambda)\widetilde{f}_j$ has small L^2 norm, then a spectral resolution shows that $\widetilde{f}_j$ is close to a true eigenfunction. Child (and BSV) show that it is sufficient to obtain strong bounds along the boundary of the fundamental domain. This method can be used to certify heuristic approximations to Maass forms.

Quasimode construction requires *extremely precise* approximations (i.e. hundreds or thousands of digits of precision). This works particularly well in level one, where heuristic approximations are the easiest to generate. The techniques readily generalize, but unfortunately quasimode construction cannot guarantee that every eigenvalue in an interval has been found.

2.3. Rigorous Hejhal. Hejhal's algorithm [**Hej99**] is one of the better-known algorithms for producing heuristic approximations to Maass forms. Strömberg described how to adapt Hejhal's heuristic algorithm for general level multiplier systems in [**Str05**].

The fundamental idea is to use automorphy to construct linear systems of equations for the coefficients. Rapid decay from the Bessel functions in (1) shows that the truncation $\widetilde{f}_j$ to the first M coefficients is close to f_j. Taking an appropriate linear combination of $\widetilde{f}_j$ at points $z_j = x_j + iY$ along a fixed horocycle, one obtains equations of the form

$$a_j(n)\sqrt{Y}K_{ir_j}(2\pi|n|Y) = \frac{1}{2Q}\sum_{j=1-Q}^{Q}\widetilde{f}(z_j)e(-nx_j) + (\text{truncation error}).$$

If the horocycle points are chosen so that each z_j is outside the fundamental domain, then we compute the pullbacks z_j^* to the fundamental domain and insert those instead. This introduces nonlinear mixing into the system, heuristically improving the condition number of the system of equations. Solving this system would give an approximation to a Maass form.

20 Jun 2025 20:16:01 PDT
250131-LowryDuda Version 2 - Submitted to LuCaNT

The challenge is that the spectral parameter r_j is also unknown. Instead, one guesses values of r_j and obtains heuristic coefficients for a form that would have eigenvalue $\frac{1}{4} + r_j^2$. Based on the behavior of these heuristic guesses, one can try to adjust the values r_j to try to find eigenvalues yielding coefficients with behaviors that behave more like actual coefficients.

The rigorous implementation of Hejhal's algorithm makes the truncation and other errors explicit and tracks how an initial approximation (e.g. coming from the trace formula), guaranteed to some initial precision, behaves under iteration of Hejhal's algorithm. When the initial approximation is sufficiently accurate, rigorously applying Hejhal's algorithm refines and produces provably better approximations. Unfortunately, this requires strong, rigorous initial approximations; and if the initial approximations are not sufficiently strong, this algorithm may fail to improve the precision.

3. Data Computed

We rigorously computed 35416 Maass cuspforms on $\Gamma_0(N)$ across squarefree N from 1 to 105. For each Maass form, we compute the eigenvalue, the first 1000 coefficients, and a portrait (constructed using similar methods as [**LD22**]). As an example, we consider the Maass form with the smallest eigenvalue on $\Gamma_0(15)$ This has the following data

`Level`	15
`Weight`	0
`Character`	15.1
`Symmetry`	odd
`Fricke Sign`	-1
`Spectral Parameter`	$1.518429336024165880367467266 26 \pm 6 \cdot 10^{-12}$
`Spectral Index`	1

In addition, the first 1000 coefficients are stored as pairs of (`center`, `error`) pairs.

The level, weight, character, and spectral parameter are defined as in §1. Here, the character is specified by its Conrey label. The `Spectral Index` gives the index of the eigenvalue among the sorted list of all eigenvalues of Maass forms of that level, weight, and character, starting at 1.

Each Maass form is also an eigenfunction under the reflection operator $J(z) = -\overline{z}$ and either has "even" symmetry (when $f_j(-\overline{z}) = f_j(z)$) or "odd" symmetry (when $f_j(-\overline{z}) = -f_j(z)$). Further, each Maass form is an eigenfunction under the Fricke involution. The `Symmetry` and `Fricke Sign` pieces of data give the eigenvalues under these operators, respectively.

Remark 3. When available, we give additional heuristic digits of precision for spectral parameters. For example, the spectral parameter in this example is specified to more than 12 digits. We expect several of these digits to be accurate, but cannot yet guarantee this.

Remark 4. In principle the symmetry type and Fricke sign can be computed from the rest of the Maass form data. But for 15423 of the 35416 Maass forms, the coefficients aren't computed with enough precision to identify the Fricke sign.

These pieces of data form the label, uniquely identifying Maass forms with a fixed eigenvalue. The full label takes the form

$$\text{Label} = \text{Level.Weight.ConreyIndex.SpecIndex.Disambig.}$$

The final term in the label is designed to disambiguate between the finitely many forms of the same level, weight, character, and eigenvalue: it gives the lexicographical index of the sequence $\{(\mathrm{Re}(a_j(n)), \mathrm{Im}(a_j(n)))\}_{n\geq 1}$ among those Maass forms. The form above has the label 15.0.1.1.1.

In the initial database, the weight is 0, the character is trivial, and the disambiguation index is always 1. Thus forms can also be specified by `Level.SpecIndex`, and the form above has short label 15.1. Both links work in the LMFDB.

In total, the database consists of approximately 4.954 GB of Maass form data.

Level 1 is distinguished, because both trace formula methods and verification methods work much better there. There are 2202 Maass forms of level 1 in the database, each verified using the quasimode construction above. For each level $2 \leq n \leq 105$, we computed as many Maass forms as we could while guaranteeing that no eigenvalues were omitted.

4. Comments on Database Construction

Finally, we conclude with several small comments.

(1) For 15423 forms, the precision from the trace formula isn't high enough for the current rigorous implementation of Hejhal's algorithm to refine. For these forms, the database includes reasonable estimates on the eigenvalue but very poor estimates for the coefficients. Different methods to handle these problems are in development. The plots for the forms with low-quality coefficients are thus informal.

(2) By restricting to squarefree level, we omit Maass forms coming from induced representations of Hecke characters. These are Maass cuspforms that are *explicitly computable* (see e.g. [**Maa49**]). Algorithms to compute these efficiently have been recently implemented in PARI/GP by Molin and Page [**MP22**]. We also miss more complicated (and potentially interesting) Artin representations.

Forthcoming work of Booker, Bober, Knightly, Krishnamurthy, Lee, Lowry-Duda, and Seymour-Howell seeks to work out an explicit, computable trace formula for general level and weight.

(3) The database currently omits all L-functions of Maass forms. In principle, these can be constructed from current data. To do this rigorously, it would suffice to adjust existing algorithms and implementations to handle interval arithmetic (as the individual coefficients are specified by intervals instead of algebraic numbers). This is attainable, but was not considered a priority.

Acknowledgements

We thank Andrew Booker, Min Lee, Brendan Hassett, Andrei Seymour-Howell, and Drew Sutherland for guidance and support throughout the computation. We also thank John Voight for several suggestions on how to improve the initial form of the database, as well as David Roe, who helped improve the database following these suggestions. This task was easier thanks to the work of Edgar Costa and David Roe, and the entire process they helped create around contributing to the LMFDB [**CR21**].

References

[BBD+24] Andrew R Booker, Ce Bian, Austin Docherty, Michael Jacobson, and Andrei Seymour-Howell, *Unconditional computation of the class groups of real quadratic fields*, LuCaNT: LMFDB, computation, and number theory, American Mathematical Society, 2024. (Cited on page 3)

[BSV06] Andrew R Booker, Andreas Strömbergsson, and Akshay Venkatesh, *Effective computation of Maass cusp forms*, International mathematics research notices (2006). (Cited on page 2, 3)

[Chi22] Kieran Child, *Twist-minimal trace formula for holomorphic cusp forms*, Res. Number Theory **8** (2022), no. 1, Paper No. 11, 27. MR 4356511 (Cited on page 2, 3)

[CR21] Edgar Costa and David Roe, *Zen and the art of database maintenance*, Arithmetic Geometry, Number Theory, and Computation, Springer, 2021, pp. 277–292. (Cited on page 5)

[Hej99] Dennis A Hejhal, *On eigenfunctions of the Laplacian for Hecke triangle groups*, Emerging applications of number theory, Springer, 1999, pp. 291–315. (Cited on page 3)

[LD22] David Lowry-Duda, *Visualizing modular forms*, Simons Symposia on Arithmetic Geometry, Number Theory, and Computation (2022), `https://arxiv.org/abs/2002.05234`. (Cited on page 4)

[LDSH] David Lowry-Duda and Andrei Seymour-Howell, *A rigorous implementation of Hejhal's algorithm*, Forthcoming. (Cited on page 2)

[LMF19] The LMFDB Collaboration, *The L-functions and modular forms database*, `http://www.lmfdb.org`, 2019, [Online; accessed 30 October 2019]. (Cited on page 2)

[Maa49] Hans Maass, *Automorphe funktionen und indefinite quadratische formen*, Automorphe Funktionen und indefinite quadratische Formen, Springer, 1949, pp. 3–42. (Cited on page 5)

[MP22] Pascal Molin and Aurel Page, *Computing groups of Hecke characters*, Research in Number Theory **8** (2022), no. 4, 91. (Cited on page 5)

[SH22] Andrei Seymour-Howell, *Rigorous computation of Maass cusp forms of squarefree level*, Research in Number Theory **8** (2022), no. 4, 83. (Cited on page 2)

[Str05] Fredrik Strömberg, *Computational aspects of Maass waveforms*, Ph.D. thesis, Uppsala Universitet, 2005. (Cited on page 3)

[The05] Holger Then, *Maass cusp forms for large eigenvalues*, Mathematics of computation **74** (2005), no. 249, 363–381. (Cited on page 2)

ICERM, 121 South Main Street, Box E, 11th Floor, Providence, RI, 02903
Email address: `david@lowryduda.com`