The Journal of Open Source Software

hilbertmodgroup: Reduction algorithms and framework for Hilbert Modular Groups

Fredrik Strömberg[1]

1 University of Nottingham, School of Mathematics

DOI: 10.21105/joss.03996

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @tbirkandan
- @videlec

Submitted: 28 November 2021
Published: 07 April 2022

Summary

This package implements basic classes and a new reduction algorithm for Hilbert modular groups. The main improvement over previous algorithms is that this implementation works in theory for all Hilbert modular groups and in practice for a much wider range of examples. A more in-depth discussion of the theoretical background and details about the implementation can be found in Strömberg (2022).

A Brief Mathematical Background

One of the most important groups in Number Theory is the modular group, $\Gamma = \mathrm{PSL}_2(\mathbb{Z})$, consisting of fractional-linear transformations, $z \mapsto (az+b)/(cz+d)$ on the complex upper half-plane, $\mathbb{H}$, given by 2-by-2 matrices of determinant 1 and integer entries.

A *reduction* algorithm for the modular group is an algorithm that, for a given $z \in \mathbb{H}$, finds an element, $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \Gamma$ such that $Az = (az+b)/(cz+d)$ belongs to a specific set, a so-called "fundamental domain". This type of algorithm was first introduced in the context of binary quadratic forms in the 18th century by Lagrange, Gauss and others, with the main contribution published by Gauss (1966) in the famous *Disquisitiones Arithmeticae*.

A natural generalisation of the modular group over $\mathbb{Z}$ is given by the family of Hilbert modular groups, $\Gamma_K = \mathrm{PSL}_2(\mathcal{O}_K)$, where K is a totally real number field of degree n and $\mathcal{O}_K$ is its ring of integers. This group gives rise to an action on n copies of the complex upper half-plane

$$\mathbb{H}_K = \mathbb{H} \times \cdots \times \mathbb{H}.$$

A reduction algorithm for a Hilbert modular group Γ_K should work in the same way as before. Given $z \in \mathbb{H}_K$ the algorithm finds an element $A \in \Gamma_K$ such that Az belongs to a certain fundamental domain. The additional complexity in this case, when K is not equal to $\mathbb{Q}$, has both a theoretical and a practical part. The main theoretical problem arises when the number field K has class number greater than 1, in which case the corresponding Fundamental domain will have more than one point at "infinity". From a practical standpoint the main problem appears when the degree and discriminant of the number field increases, making it necessary to, for instance, locate integral points in higher-dimensional polytopes.

Statement of need

There have been several previous attempts at giving a reduction algorithm for Hilbert modular groups but they have all been limited in at least one of two ways: the number field either being restricted to degree 2, or the class number to be 1, or a combination of both. See for example the algorithms by Bouyer & Streng (2015) and Quinn & Verjovsky (2020).

Having access to the algorithm in this package, which is valid for any totally real number field, opens up for several new research directions and generalisations of previous research. Some of the direct applications to be pursued by the package author and collaborators lie in the field of explicit formulas and computational aspects of non-holomorphic Hilbert modular forms.

Implementation

The package `hilbertmodgroup` is mainly written in Python with some parts in Cython (Behnel et al., 2011). It is intended to run as a package inside SageMath (The Sage Developers, 2022) as it makes heavy use SageMath's implementation of number fields, which is in turn is in many cases using the backend from PARI/gp (The PARI Group, 2021).

Documentation and Examples

All functions are documented using docstrings with integrated doctests following the guide for SageMath development. In addition, the `/examples` directory contains
Jupyter notebooks illustrating the use of the package with a selection of fundamental examples, corresponding to examples presented in Strömberg (2022).

Acknowledgements

This work was partially supported by the Engineering and Physical Sciences Research Council [EP/N007360/1] and [EP/V026321/1].

References

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering*, *13*(2), 31–39. https://doi.org/10.1109/mcse.2010.118

Bouyer, F., & Streng, M. (2015). Examples of CM curves of genus two defined over the reflex field. *LMS Journal of Computational Mathematics*, *18*(1), 507–538. https://doi.org/10.1112/S1461157015000121

Gauss, C. F. (1966). *Disquisitiones arithmeticae* (p. xx+472). Yale University Press, New Haven, Conn.-London.

Quinn, J., & Verjovsky, A. (2020). Cusp shapes of Hilbert-Blumenthal surfaces. *Geometriae Dedicata*, *206*, 27–42. https://doi.org/10.1007/s10711-019-00474-w

Strömberg, F. (2022). A reduction algorithm for Hilbert modular groups. *Journal of Number Theory*. https://doi.org/10.1016/j.jnt.2022.02.011

The PARI Group. (2021). *PARI/GP version `2.13.1`*. https://pari.math.u-bordeaux.fr

The Sage Developers. (2022). *Sage Mathematics Software (Version 9.5)*. https://www.sagemath.org