Creating a dynamic database of finite groups

Lewis Combes, John Jones, Jennifer Paulhus, David Roe, Manami Roy, and Sam Schiavone

ABSTRACT. A database of abstract groups has been added to the *L-functions and Modular Forms Database* (LMFDB), available at https://www.lmfdb.org/Groups/Abstract/. We discuss the functionality of the database and what makes it distinct from other available databases of abstract groups. We describe solutions to mathematical problems we encountered while creating the database, as well as connections between the abstract groups database and other collections of objects in the LMFDB.

1. Introduction

Finite groups have played a profound role in mathematics for close to two centuries and, almost since their inception, mathematicians have asked classification questions about them. The quest to classify finite simple groups took up most of the 20th century. In the last 30 years, as computational power became prominent, various databases of groups have been developed. The *ATLAS of Finite Groups* which contains information on interesting groups, notably character tables, began as a book [**CCN$^+$85**] and is now available online [**WWT$^+$**]. The computer algebra programs GAP [**GAP21, BEO02**] and Magma [**BCP97**] both have complete databases of groups of order up to 2000 (except for order 1024, for which there are almost 49.5 billion distinct isomorphism classes). These computer algebra programs also include other databases, such as almost simple and perfect groups (up to orders $1.6 \cdot 10^7$ and $2 \cdot 10^6$, respectively) and transitive groups up to degree 48 [**Hul**], [**BCH$^+$**]. The *GroupNames* project [**Dok**] includes groups up to order 500 (skipping orders 256 and 384), as well as additional information such as lattice of subgroups and character tables for the groups.

We have created an online database of over 540,000 finite groups, 275 million of their subgroups, and almost 40 million irreducible complex characters. The database can be found at: https://www.lmfdb.org/Groups/Abstract/.

1.1. Highlights of the database. As we note above, there are several other databases of groups already available, so why do we need yet another one? Some of the useful features of our database are the following.

- *The database is dynamically searchable.* We have precomputed and stored many invariants and properties of the groups in the collection, and created a search interface which allows users to search on these stored values. For instance, we can quickly determine the 8 groups G of order 1440 which satisfy the short exact sequence

$$1 \to D_{10} \to G \to F_9 \to 1,$$

with D_n the dihedral group of order $2n$ and F_q the group of linear affine transformations of the finite field $\mathbb{F}_q$; we can immediately determine that the smallest group to have an irreducible character of degree 7 is a group of order 56; in a matter of seconds we can find all 129,069 subgroups of groups of order between 1 and 255 which are non-normal, non-abelian, non-maximal, and non-trivial proper. The list of all such subgroups (a 23 megabyte file) can be downloaded in a few seconds. Each of these searches would be time-consuming and computationally intensive in computer algebra systems like Magma or GAP.

- *The database is free and easy to use.* Use of our database requires no prior knowledge of any programming language or computer algebra package, making it accessible to a broad audience, from students learning the basics of group theory to experts using groups in their research. While most of our computations were done in Magma, which is closed-source and requires purchase of a license, the data we have computed is now publicly available on the internet through the *L-functions and Modular Forms Database* (LMFDB).

- *The database fosters connections between other collections of objects in the LMFDB.* The LMFDB is a huge international collaboration to collect, curate, and connect computational mathematical work in number theory, particularly as it relates to the Langlands program. Groups are attached to many mathematical objects throughout the LMFDB, from Sato-Tate groups to automorphism groups of curves, and our database of groups facilitates more connections between these various collections. See Figure 3 for a diagram illustrating some of these connections.

- *The database aggregates groups drawn from a variety of sources.* The database is flexible in that it can accommodate groups of various types, such as polycyclic, permutation, and matrix groups, drawn from multiple sources. See Section 2 for a list of sources used to compute the data.

- *The database supports searches on subgroups.* For each group in the database, we have computed and stored information on its subgroups. Users can search for all groups containing a subgroup with a given property; for instance, one can easily find the 12 groups of order 96 that contain a normal subgroup isomorphic to A_4; see Section 3.3 for more details on the subgroup information stored.

1.2. Structure of the paper. We begin in Section 2 with information about sources for the groups currently included in the database, and how the data about them is generated. In Section 3 we describe the web interface and demonstrate a number of features of the database. In the process of designing the database and computing the corresponding data, we encountered a series of challenges which required mathematical solutions. The next two sections discuss these challenges: Section 4 describes the method by which we computed presentations for solvable groups and Section 5 details the procedures we used to deterministically label subgroups, conjugacy classes, and characters. In Section 6 we give examples of connections between the abstract groups database and other collections of objects in the LMFDB.

---

[1]A *knowl* is a feature of the LMFDB, where definitions and results are stored within webpages and can be revealed by clicking. The term was coined to reflect that each knowl contains a little bit of knowlege.

Mathematics for a workshop where the project took off. We are grateful to the referees for their comments, which have improved the paper.

2. Data overview

2.1. Sources of data. The database currently contains roughly 540,000 groups, 275 million subgroups and 40 million of their irreducible complex characters.

The code used to generate the data is written in Magma [**BCP97**] and Python [**VRD09**], and it can be found in the GitHub repository `https://github.com/roed314/FiniteGroups`. The database, as with the rest of the LMFDB, uses PostgreSQL as its database management system. Computations were carried out via a combination of an AMD Epyc 7713 server with 256 2.0GHz cores and 2TB of RAM, and a distributed computation on Google Compute Engine, with a substantial use of GNU parallel [**Tan11**] in both cases.

Source	Total	Solvable	Perm.	Matrix	OptimizedPC	MinPerm
Small Groups	257936	257500	257746	68042	257500	257746
Transitive Groups	235919	211279	235919	218	14499	161656
Intransitive Groups	5444	2739	5444	16	2330	5378
Classical Lie Type	2201	2	1509	2201	0	1509
CARAT	189	185	186	189	174	186
$\mathrm{GL}_n(\mathbb{F}_q)$ Subgroups	3018	2456	3000	3018	2397	3000
$\mathrm{GL}_2(\mathbb{Z}/N)$ Subgroups	29771	28819	25319	29771	24323	25254
Perfect	123	0	123	1	0	123
Chevalley	13	0	7	13	0	7
Sporadic	9	0	9	7	0	9
Small Group Auto.	283	283	283	0	73	111
Transitive Group Auto.	498	438	498	0	51	201
Auto. Groups of Curves	530	527	530	0	526	530

TABLE 1. Number of groups by source. See Section 2 for more information.

We compute and store data on groups from various sources. Counts of groups currently in the database from these different sources are given in Table 1. The "Total" column represents the total number of groups in the database from that source, excluding those already shown on a previous line. The "Solvable", "Perm.", and "Matrix" columns give the number of groups from each source that are solvable, the number for which we store a permutation representation, and the number for which we store a matrix representation (over $\mathbb{Z}$, $\mathbb{F}_p$, $\mathbb{F}_q$, or $\mathbb{Z}/N$), respectively. The "OptimizedPC" column counts how many groups from a given source include an optimized polycyclic presentation (see Section 4), and "MinPerm" gives the number of groups for which we store a minimal degree permutation representation.

Approximately half of the data originates from the Small Groups library in Magma [**BE99a**, **BE99b**, **BEO01**, **O'B90**, **BE01**, **O'B91**, **NOVL04**, **OVL05**, **DE05**, **DE12**, **DEP22**]. All groups of order up to 2000 are included, except those whose order is larger than 500 and divisible by 128. In the rest of the paper, these are the groups we will call "small groups". Most of the remaining groups come from the transitive

groups database [**Hul05**, **CH08**, **HR20**], from which we include all groups of degree up to 47, except those of degree 32, with orders between 512 and 40 billion. Note that our notion of equivalence in this database is abstract isomorphism rather than conjugacy within S_n; the work to divide the transitive groups into isomorphism classes is briefly described in Section 2.5. In addition to the databases of small groups and transitive groups, we use the following sources:

- classical Lie groups up to particular bounds, and Chevalley groups that don't already show up as classical Lie groups [**Tay87**, **RT98**, **HRT01**];
- additional intransitive groups which are subgroups of S_{15} not currently in the database, computed using the Magma Subgroup command;
- all integer matrix groups of dimension up to 6 from CARAT [**OPS08**];
- subgroups of $\mathrm{GL}_n(\mathbb{F}_q)$ computed via Magma for $n = 2$ ($q < 1000$), $n = 3$ ($q < 16$), $n = 4$ ($q < 7$) and $n = 5$ ($q = 2$);
- subgroups of $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$ for N up to 125 (skipping 80, 96, 104, 112 and 120);
- all perfect groups of order up to 50,000 from the corresponding database in Magma;
- the sporadic groups J_1, J_2, HS, J_3, McL, He, Ru, Co_3, and Co_2 with permutation and matrix representations taken from the Atlas of Sporadic Groups [**WWT$^+$**];
- automorphism groups of curves up to genus 48 [**Bre00**], and "large" automorphism groups (of order $> 4(g-1)$) of curves up to genus 101 [**Con10**].

Since the automorphism groups of small groups can be much larger than the groups themselves, we also include all automorphism groups of groups of order up to 255 and of transitive groups of degree up to 23.

2.2. Basic attributes computed for every group. For each group, we compute as many attributes as possible. Some use commands directly from Magma, such as determining if a group is abelian, while others require special functions we wrote (for example, determining if a group is metacyclic). We compute a "reasonable" presentation for the group (see Section 4) and a minimal degree permutation representation. We usually compute the lattice of subgroups up to automorphisms of the group, and up to conjugacy when possible (Sections 2.4 and 3.3), and conjugacy classes and character tables when feasible. We also determine special subgroups such as the center, commutator, Frattini, and Fitting subgroups, as well as various series for the group.

2.3. Challenging attributes computed for many groups. Many of our quantities of interest become more difficult to compute as the size or degree of the group increases. Our general approach is to try to compute everything for all groups, setting appropriate timeouts and omitting data if the computation does not finish or if we encounter errors in Magma. We have developed additional code for some problems where Magma's built-in methods were insufficient for our purposes; notable examples include computing quotients of permutation groups and isomorphism testing (discussed in Section 2.5).

2.4. Computing subgroup lattices. The subgroup lattice is an example of a particularly challenging attribute to compute. For each group, we attempt to compute the full lattice of subgroups, but there are many groups in our database where it is infeasible to compute subgroups up to conjugacy. For example, C_2^{10} has 229,755,605 subgroups, none of which are conjugate. However, it only has 11 classes of subgroups up to automorphism. We implemented the computation of the subgroup lattice up to automorphism and improved on Magma's built-in subgroup lattice methods up to conjugacy. In some cases, such as S_{47}, even working up to automorphism does not sufficiently reduce the quantity of subgroups, so we restrict our attention to normal subgroups and their complements, maximal subgroups, Sylow subgroups, and subgroups with small index and/or trivial core due to their importance in permutation representations.

Magma has built-in methods for computing both the subgroup lattice and the list of subgroups up to conjugacy (without inclusions). We found that the second was much faster than the first, and that we could recover the inclusions more quickly by computing a vector counting the intersections with each conjugacy class: if we have a group G and subgroups $H_1 \subseteq H_2$, then the vector of counts for H_2 dominates that for H_1, dramatically decreasing the number of calls to `IsConjugateSubgroup`. As noted above, when there are many subgroups up to conjugacy (e.g. abelian p-groups), we instead compute subgroups up to automorphism. The first algorithm for doing so uses the holomorph of G, i.e., the semidirect product $G \rtimes \mathrm{Aut}(G)$. Conjugacy within the holomorph translates to automorphism in the group itself, and a code snippet kindly provided by Derek Holt, using lifting through an elementary abelian series, allowed for the computation of representatives of subgroups up to automorphism in the solvable case without computing the list up to conjugacy.

However, the holomorph is implemented in Magma as a permutation group of degree equal to the order of G, and as the size of G increases, computations using the holomorph get bogged down. For non-solvable G and G with order at least 5000, we switch to a graph theoretic algorithm. First we compute a list of subgroups up to conjugacy, together with a list of automorphisms that generate the outer automorphism group of G. Taking subgroups up to conjugacy as vertices, we add an edge between H_1 and H_2 if there is an outer generator σ with $\sigma(H_1)$ conjugate to H_2. The components of the resulting graph give the subgroups up to automorphism,.

2.5. The isomorphism problem. When adding transitive permutation groups to our abstract groups database, we had to ensure that groups with multiple transitive permutation representations were only added once each. For small groups, Magma will compute its small group identification number. In a larger range, the LMFDB's transitive group section already had the needed information. However, there were multiple cases where neither approach sufficed.

Running isomorphism tests can be time consuming, so we used the following strategy. We took multiple isomorphism invariants of a group which are quick to compute and combined them into a hash. This could be precomputed for large numbers of groups. Groups with different hashes were clearly non-isomorphic. If groups produced identical hashes, we then had Magma perform a slower, but conclusive test for isomorphism. Our hash was highly effective in distinguishing non-isomorphic groups. Full details will be given in a forthcoming article by Roe [**Roe**].

3. Features of the database

The database is viewable through the LMFDB website at `https://www.lmfdb.org/Groups/Abstract/`. In this section we describe how to navigate the web interface, and highlight some particular features.

3.1. Features of the search interface. Groups can be searched through numerous query types, for example: the isomorphism class of agroup's automorphism group, commutator, center, abelianization, Frattini subgroup, etc.; the order of any of these and the group itself; many boolean properties such as nilpotency, simplicity, and solubility. With the search interface, it is easy to quickly answer questions such as *"What are the nilpotent groups of order 36?"* and *"Are there any groups of order 256 with abelianization $C_2 \times C_2$?"* In this way, the database provides a useful service to researchers looking for groups with particular properties, as well as students coming to grips with the basics of group theory. It is also possible to search for groups whose orders factor in a particular way. For example, groups whose order is of the form $p^3 q^2 r$ for primes p, q, r can be queried with the string `[3,2,1]`.

Groups are assigned a unique label in the database (see Section 5 for more details). These labels can be used for searching for specific groups, and match those in the GAP Small Groups database when the

ranges overlap. If the label is not known beforehand, several options exist for searching for a specific group. Many families of groups can be found using strings. For example, `GL(n,q)` gives the page for $\mathrm{GL}_n(\mathbb{F}_q)$, and the strings `D10` and `F5` give the two groups in the example from Section 1.1—a full list of supported "transliterations" can be found alongside the database in the `group.families` knowl. Groups defined in `Magma` can also be searched for using the output from the `GroupName` function.

There is also a curated list of some interesting groups that one can quickly access, and one can also view the page of a group picked randomly from the database.

3.2. Features of a group page. On individual group pages, information is organized roughly by topic. Constructions of the group via a presentation (see Section 4), and as direct, semidirect, and non-split products are given when possible, as well as representations as a matrix or permutation group. Homological information like the Schur multiplier and commutator length is provided. Special subgroups like the center, commutator, and socle are displayed. For many groups, we provide several subgroup diagrams; one can toggle whether subgroups are shown up to conjugacy or up to automorphism (see Section 3.3), and independently whether to show all such classes of subgroups or just normal subgroups. We also display the derived series, the chief series, and the upper and lower central series of the group. Groups for which the given group is a maximal subgroup or a maximal quotient are listed under the Supergroups section.

Groups are also represented pictorially, in line with other sections of the LMFDB. Each disc in a group picture represents a conjugacy class of elements, arranged in an annulus around the middle class (of the identity), with distance according to the number of prime factors in the class' order. The size of a disc is proportional to the size of the conjugacy class, and the color is chosen based on its order. Decisions on how to display a group in picture form are essentially a matter of taste; our conventions lead to some group pictures showing striking symmetries within the conjugacy classes, such as that of `480.60` in Figure 1.
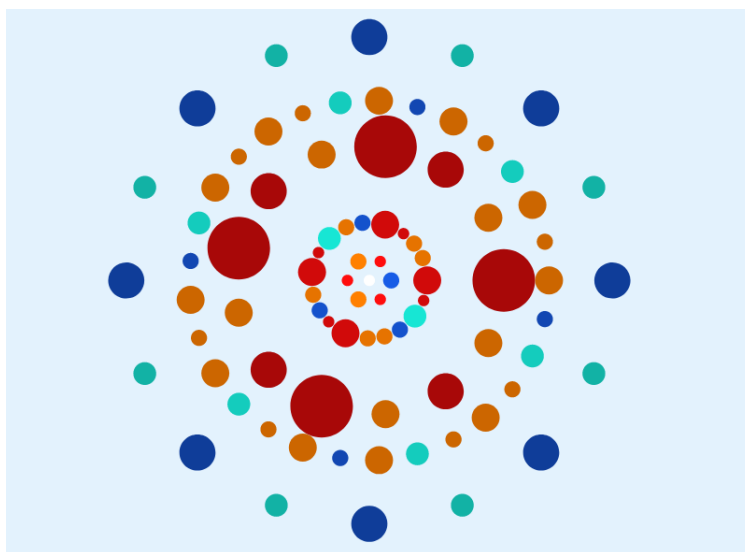

FIGURE 1. Group picture of `480.60`

For many groups, we provide character tables for both complex and rational characters. Rows correspond to irreducible characters, and columns correspond to either conjugacy classes (for the complex character table) or divisions (for the rational character table); for more details on conjugacy classes and divisions, see

Section 5.1. For ease of terminology, in this discussion we refer to columns as corresponding to "classes" to cover both cases simultaneously.

The headers for classes expand to show basic information such as the class's size and centralizer, as well as a representative for the class. As is standard, the row(s) above the table contain power map information on classes. For each prime p dividing the order of the group and each class represented by an element g, the entry above that column in the row labeled "pP" gives the label for the class of g^p. For each complex character, in the 2$^{\text{nd}}$ column of the table, we designate whether the representation is real, complex, or quaternionic.

REMARK 3.1. For some groups which lie outside the range of the database, we still generate webpages that display some basic information about the group by using on-the-fly calculations in GAP and SageMath. Two types of groups that are handled this way are groups in GAP's Small Groups database that are outside the LMFDB collection, such as 512.402873, and large abelian groups, such as $C_{12}^3 \times C_6$.

REMARK 3.2. Statistics about the abstract groups database can be found at https://www.lmfdb.org/Groups/Abstract/stats. We have computed many statistics based on the order factorization type of a group, i.e., whether the order is a prime, a product of two distinct primes, a power of one prime times another prime, etc. For instance, one can see the distribution of the nilpotency class of groups with order p^k, where p is a prime and $k \in \{3, \ldots, 6\}$.

3.3. Subgroups. One of the main features of our work is a companion database of subgroups. Subgroups have their own search functionality and labeling system (see Section 5.3). Given a group H, it is possible to search the database for all groups G containing H as a subgroup, subject to numerical constraints such as the index of H in G and the order of G. Cyclic subgroups, normal subgroups, and Sylow and Hall subgroups of a group can all be searched for. Each subgroup also has its own page. These pages give information about the ambient group and the quotient group structure (when normal). Related subgroups such as the centralizer, normalizer, normal closure and core of the subgroup are given.

The subgroups database allows us to display the subgroup lattice for many groups. This lattice is algorithmically very useful, allowing the computation of generating data about the group, such as the size of a smallest generating set (called the rank)[2] and how many such sets a group has, as well as expressions for the group as a semidirect product, its linear and permutation degrees, etc. The subgroup lattice can be viewed on a group page when it has fewer than 100 subgroup classes; see Figure 2 for an example. Inclusion is indicated with lines, and clicking a particular subgroup reveals information about it, e.g., whether it is maximal or solvable, its normalizer, its core. Subgroups are separated into levels vertically based on their orders or by the number of prime factors in their order (the user can toggle between these two options), and can be rearranged by clicking and dragging. When the diagram is too large to conveniently display on the page, the full diagram can often be viewed on a separate linked page.

In addition, we provide profiles of the subgroups up to automorphism and up to conjugation. For each order of subgroup, the profile lists the isomorphism types and multiplicities of classes of subgroups with that type. For example, for the elementary abelian group C_2^5, the profile for subgroups up to conjugation would list 155 conjugacy classes of subgroups of order 8 (all isomorphic to C_2^3), and one class of subgroups of the same order up to automorphism.

4. Computing presentations for solvable groups

As mentioned in Section 3.2, each group page contains various constructions of the group. In the case of solvable groups we give particularly nice presentations, exploiting the fact that solvable groups are polycyclic.

---

[2]Although this is now more easily computable with Magma's SmallestGeneratingSet command
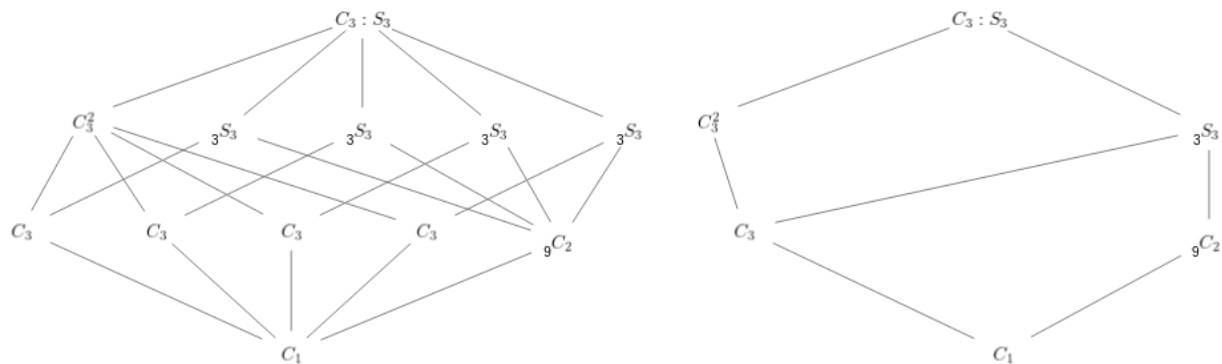
7

FIGURE 2. Subgroup diagrams of 18.4 up to conjugacy and autjugacy, as they appear on the LMFDB

A group G is *polycyclic* if it has a descending chain of subgroups

$$G = G_1 \geq G_2 \geq \cdots \geq G_{n+1} = 1$$

such that each G_{i+1} is normal in G_i, and the quotient G_i/G_{i+1} is cyclic. A polycyclic presentation is a special type of presentation for the group G that takes advantage of this descending chain of subgroups; for details, see [**HEO05**, Definition 8.7].

However, the polycyclic presentations in Magma use one generator for each prime dividing the order of the group (counted with multiplicity). So, for example, the cyclic group of order $2^2 \cdot 3 \cdot 5$ would have a presentation with 4 generators instead of the simpler one using a single generator. Here we describe a process for finding a more human-readable polycyclic presentation for a solvable group by using chains of subgroups. One can pass between polycyclic presentations and subnormal filtrations with cyclic quotients:

(1) given a polycyclic presentation with generators $\{g_1, \ldots, g_m\}$, the chain

$$1 \leq H_1 \leq \cdots \leq H_m = G$$

will be a filtration of G with cyclic quotients, where $H_i = \langle g_1, \ldots, g_i \rangle$;

(2) given a filtration of the form above, any choice of elements g_i generating H_i/H_{i-1} will give generators for a polycyclic presentation.

Define the *relative order* of g_i to be the order of the quotient H_i/H_{i-1}. We approach the search for a presentation by first finding a filtration and then arbitrarily choosing generators.

We first construct all minimal length chains of subgroups, each normal in the next, with cyclic relative quotients. We do this via a "top-down" approach, building them down from the top in layers until reaching the trivial subgroup. This process guarantees a polycyclic presentation with a minimal number of generators; we then compare all such minimal polycyclic presentations according to the following criteria.

(1) Maximize the number of generators with order equal to their relative order.
(2) Maximize the number of generators that commute with each other.
(3) Aim for relative orders that are non-increasing.
(4) Conjugacy relations should be "deeper" (within smaller groups in the filtration).

This process cannot feasibly be made canonical: at some points we make arbitrary choices, both for the filtration and for the choice of generators.

There are some groups for which the process of constructing and comparing all minimal length chains is too time-consuming. In these cases we instead use two other algorithms, both of which run much faster, but may not give as good a presentation. The first proceeds by greedily building the chains by starting with $H = G$ and iteratively choosing a random normal subgroup $K < H$ with H/K cyclic. The second algorithm simply refines the derived series by taking an abelian basis at each step in order to fill in the series to one with cyclic quotients.

It is important to pick the presentation at the beginning of our computations for each solvable group, as this presentation impacts certain attributes of the group, such as how we represent its elements.

5. Labels

A foundational principle of the LMFDB is to display every object with a unique label, an identifier that carries mathematically relevant information which can ideally be computed in a deterministic way. The primary role of the label is to serve as a permanent identifier for the object (so that it can be referenced in the mathematical literature, for example). Labels generally encode a sequence of invariants of an object, and provide a natural ordering. Conversely, a sequential ordering of all objects of a given type trivially allows one to label them, by their position in the sequence. We will talk of having labels and an ordering interchangeably. When sorting ordered tuples, we always mean lexicographically, and treat complex numbers as ordered pairs of real numbers for this purpose.

We label small groups as `N.i` corresponding to the GAP ID encoded as a string, where `N` is the order of the group and `i` distinguishes groups of the same order (as determined in GAP). If a group is not in the GAP Small Groups database, we replace `i` with an incrementing letter code, assigning labels to groups as they are added to our database.

The existence of automorphisms precludes having a definitive labeling *de novo*. Consequently, the labels for conjugacy classes, characters, and subgroups depend on fixing an ordered list of generators, which we do, and a specific realization of the group, i.e., as a permutation group, a polycyclic group, or as a matrix group. The realization is important since, in each case, one can easily construct an injective set map from the group to $\mathbb{Z}$, which then orders elements of the group. We then need to have a reproducible method for generating elements of a group. In the applications below, we found that picking pseudo-random group elements worked efficiently; for details on how we generate such elements, see `https://www.lmfdb.org/knowledge/show/group.pseudo_random_elements`.

5.1. Conjugacy classes and divisions. If G is a group and $g_1, g_2 \in G$, we say that g_1 and g_2 are in the same *division* if there exists $h \in G$ such that $h^{-1}g_1 h = g_2^i$ for some $i \in \mathbb{Z}$ such that $\gcd(i, |g_1|) = 1$. Divisions are unions of conjugacy classes, and rational-valued characters are constant on divisions. In fact, the irreducible rational-valued characters form a basis for the space of functions that are constant on divisions, as complex characters do for class functions. We say that a division is *maximal* if a representative generates a maximal cyclic subgroup of G.

Full details of how we label conjugacy classes and divisions are given at `https://www.lmfdb.org/knowledge/show/group.division_computing_labels`; here we describe some properties of the labeling.

Divisions are labeled with `nA` where `n` is a positive integer giving the order of a representative element and `A` is a string made up of capital letters. They are ordered first by the size of a conjugacy class within the division, and then by the number of conjugacy classes within a division.

Finding representatives for each division can be computationally expensive, so we focus first on maximal divisions. From there, we can compute labels and representatives for other classes as follows. Working through the maximal divisions in order, we use its chosen representative g and compute the classes of elements g^i

with $0 < i < |g|$ and $\gcd(i, |g|) > 1$. The order of the non-maximal divisions is simply the order in which they appear from this sequence (looping over divisions, and within that, powers of an element).

The label for a conjugacy class starts with the label of the division containing it. If there is only one class within the division, the two labels are the same. When there is more than one class in a division `nA`, the conjugacy classes are labeled `nAj` where `j` is an integer. The first class encountered will then be `nA1`. Let g be an element representing this class. We then consider the conjugacy classes $[g^{-1}]$, $[g^2]$, $[g^{-2}]$, ... These correspond to conjugacy class labels `nA-1`, `nA2`, `nA-2`, ... The label for the class is the first time it appears in this sequence. Conjugacy classes which land in the same division have the same prefix, and will be grouped together in the complex character table for the group.

5.2. Characters. When feasible, we give information on irreducible rational-valued and complex-valued characters. The group $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts on the complex characters, and the Galois orbits correspond to the rational characters, with each rational character being simply the sum of the complex characters in an orbit. We note that a rational-valued character may not arise from a rational-valued representation. The Schur index gives the smallest multiplier for a rational character so that the resulting character arises from a rational-valued representation.

Permutation representations of a group G can be realized using permutation matrices, and are thus sums of irreducible representations. Transitive permutation representations are classified by their degree n and T-number t [**CHM98**]. For a given irreducible representation, we can consider the "first" transitive permutation containing it to be the smallest pair (n, t).

The labels for rational characters are of the form `G.na` where `G` is the label for the group, `n` is the degree, and `a` is a lower-case letter which acts as a counter. The label for a complex character takes the form `G.nak` where `G.na` is the label for the corresponding rational character and `k` is positive integer which serves as a counter. In the character tables, characters are sorted by their labels viewed as a tuple (n, a) or (n, a, k). To completely determine the labels, we just need to know the ordering among characters with the same degree.

For rational characters, we sort by (d, m, n, t) where d is the degree (which is explicitly given in the label), m is the size of the Galois orbit of complex characters giving rise to it, and n and t refer to the smallest containing permutation representation. Any remaining ties are broken by sorting on the vector of character values using our ordering of conjugacy classes. Complex irreducible characters are given in the same order as their corresponding rational characters, with characters within a Galois orbit ordered by their vectors of values.

5.3. Subgroups. Labels for subgroups take one of four forms, depending on whether they are computed up to conjugacy or automorphism, and whether or not the deterministic process described below succeeds. (If it fails or we do not have all subgroups of a given order, we use a fallback nondeterministic label.) The deterministic labels take the form `N.i.m.a.c` (for subgroups up to conjugacy) or `N.i.m.a` (for subgroups up to automorphism). The piece `N.i` is the label of the ambient group G, `m` is the index of H in G, and `a` and `c` are alpha-numeric identifiers that distinguish H up to automorphism and conjugacy, respectively. To determine `a` and `c` we use the idea of Gassmann equivalence classes to order subgroups of the same index.

DEFINITION 5.1. Two subgroups H_1 and H_2 of G are *Gassmann equivalent* if they intersect each G-conjugacy class with the same cardinality. Equivalently, H_1 and H_2 have the same index m and the permutation representations $\pi_{H_1} : G \to S_m$ and $\pi_{H_2} : G \to S_m$ have the same character (which counts the number of cosets fixed by each conjugacy class).

For more on Gassmann equivalence and related notions see [**Sut21**], and for full details on the subgroup labeling scheme see https://lmfdb.org/knowledge/show/group.computing_subgroup_labels; here we describe some aspects of the labeling.

In order to label Gassmann classes, we first fix an ordering of the conjugacy classes of G as in Section 5.1. To each subgroup H we associate a vector of positive integers whose entries are the sizes of the intersections of H with each conjugacy class. We then sort the subgroups using these vectors. An analogous process works for Gassmann vectors up to automorphism, where we collect together G-conjugacy classes that are related by an automorphism of G.

If there are multiple subgroups up to conjugacy or automorphism within a given Gassmann class, we use the subgroup lattice to further order those remaining subgroups. We sort subgroups H in the same Gassmann class by considering the collection of supergroups of H; if this does not suffice to distinguish Gassmann equivalent subgroups, we resort to computing permutation representation and their associated characters.

Once we have an ordering inside the Gassmann classes, we assign a letter label to the Gassmann class and concatenate a number from the ordering of subgroups inside the Gassmann class to create an alpha-numeric value which we assign to a, and similarly, we assign an alpha=numeric value to c when we are able to work up to conjugacy. For example, the label of A_5 inside S_5 (up to conjugacy) is 120.34.2.a1.a1. As another example, there are two classes of subgroups (up to automorphism) of $\mathrm{SL}(3, 4)$ isomorphic to $\mathrm{GL}(2, 4)$ with labels 60480.a.336.a1 and 60480.a.336.a2—these are Gassmann-equivalent but not related by an automorphism—and four classes of subgroups (up to automorphism) of $\mathrm{SL}(3, 4)$ of order 8, each contained in a different Gassmann class: 60480.a.7560.a1 (Q_8), 60480.a.7560.b1 ($C_2 \times C_4$), 60480.a.7560.c1 (D_4), and 60480.a.7560.d1 (C_2^3).

In some cases, we store only a few subgroups of a given order (we may store the center of a group, for example, but not other subgroups of that order) or the code for computing the labels did not finish. In these cases, we append an arbitrary letter code to the N.i.m part of the label, with capital letters indicating that subgroups are stored up to conjugacy and lower case indicating subgroups up to automorphism.

6. Connections and examples

We anticipate that the group home pages will be a valuable tool for everyone from undergraduate students first learning abstract algebra to experts in the field. The layout of each page, the search features, and the knowls allow for exploration of many group-theoretic concepts. Students can easily find examples of groups with common features like simple or solvable groups, they can explore Sylow subgroups (and more generally subgroup lattices), and they can learn about advanced topics such as series or character theory. Students can also explore relationships between different attributes of groups. A quick search demonstrates, for example, the existence of non-trivial groups that are perfect but not simple or A-groups[3] that are not solvable.

For researchers, the abstract groups database is intimately connected with many other mathematical objects in the LMFDB. We highlight some of those connections below.

6.1. Connections with other objects. One of the integral aims of the LMFDB is demonstrating links between different mathematical objects. The LMFDB universe gives the big picture of various connections that are related to the *Langlands program.* There are many related objects, and links between them are displayed in the database. In this section we list other databases in the LMFDB which are linked with the database of Abstract Groups. We describe the various connections to other LMFDB pages with the group pages in Figure 3. Note that, at present, the boxes representing the LMFDB sections Belyi maps,

---

[3]A group G is an *A-group* if all its Sylow subgroups are abelian.

Hypergeometric motives over $\mathbb{Q}$, Lattices, and Modular curves in Figure 3 are currently only available on the beta version of the LMFDB (https://beta.lmfdb.org/).
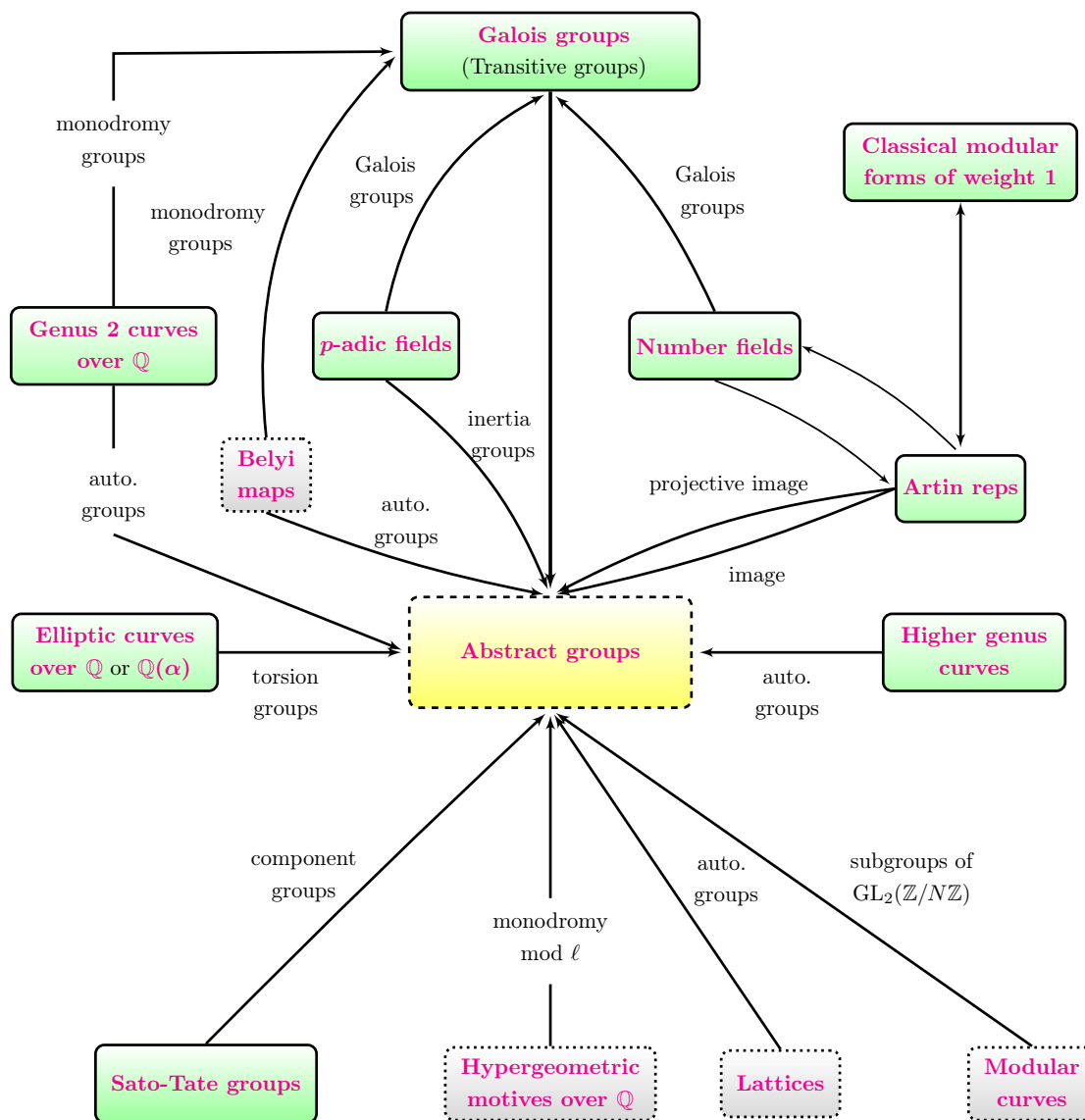

FIGURE 3. Diagram illustrating the connections between abstract group pages and other collections of objects in the LMFDB. Nodes in solid (green) and dotted (grey) rectangles are on the production and beta version of the LMFDB, respectively.

Since many objects in the LMFDB link to groups, having a built-in database of groups makes it easy for the user to find more information about the groups in question. Some groups may be familiar to the user, such as those which arise as automorphism groups of genus 2 curves; others are more complicated.

For instance, Artin representations connect to abstract groups via both their image and their projective image. One can also go from Artin representations to Galois groups (i.e., transitive permutation groups) by taking a stem field for the field cut out by the image of the representation to get a number field, and then taking the Galois group of its Galois closure. For a number field, one has the Artin representations of the Galois group of the normal closure of the field. The following example illustrates this connection further.

12

EXAMPLE 6.1. Consider the Artin representation with LMFDB label 4.5744.8t39.d.a, which has the smallest conductor of all 4-dimensional irreducible Artin representations with group https://www.lmfdb.org/Groups/Abstract/192.1493 [JR17]. This group is displayed as $C_2^3 : S_4$, but since the semidirect product notation may not uniquely identify a group, the connection to a specific group page lets the user obtain lots of information about this group. Similarly, the projective image of this group is $C_2^2 : S_4$, via which the user is referred to the page https://www.lmfdb.org/Groups/Abstract/96.227 for more details.

6.2. Applications via Galois theory. Perhaps the most celebrated application of groups is their usage in studying field extensions as a part of Galois theory. Below we give examples of how our database of finite groups, together with the Galois correspondence, can be used to study objects in other collections in the LMFDB.

EXAMPLE 6.2. Let $F = \mathbb{Q}(\alpha)$ be the number field with LMFDB label 5.1.35152.1, where α has minimal polynomial $f := x^5 - x^4 + 2x^3 - 4x^2 + x - 1$. Then F has Galois closure $L = \mathbb{Q}(\beta)$ with LMFDB label 20.0.3354518684571451850752.1, where β has minimal polynomial

$$x^{20} - 2x^{19} + 10x^{17} - 15x^{16} + 40x^{14} - 64x^{13} + 46x^{12} + 8x^{11} - 32x^{10} + 8x^9$$
$$+ 46x^8 - 64x^7 + 40x^6 - 15x^4 + 10x^3 - 2x + 1\,,$$

and $G := \mathrm{Gal}(L/\mathbb{Q}) \cong F_5$, the Frobenius group of order 20 (with LMFDB label 20.3). Examining the lattice of subgroups up to conjugacy given on the 20.3 homepage, we see that F_5 is solvable, so the roots of f can be expressed by radicals. Computing fixed fields, we obtain the subfield lattice shown in Figure 4, where K
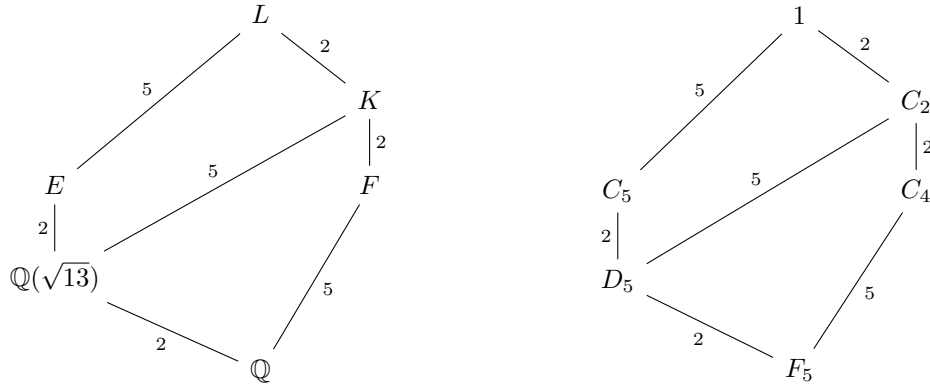

FIGURE 4. The subfield lattice of the number field with LMFDB label 20.0.3354518684571451850752.1 and the subgroup lattice of its Galois group F_5.

and E are the number fields given by

$$x^{10} - x^9 - 3x^8 + 5x^6 + x^5 - 5x^4 + 3x^2 - x - 1$$

and

$$x^4 + x^3 + 2x^2 - 4x + 3\,,$$

respectively. We observe that the commutator subgroup of G is C_5, the subgroup corresponding to E. Thus E is the largest abelian subfield of L, with Galois group $\mathrm{Gal}(E/\mathbb{Q}) \cong G^{\mathrm{ab}} \cong C_4$.

13

EXAMPLE 6.3. Let $K = \mathbb{Q}_2(\alpha)$ be the p-adic field with LMFDB label 2.4.6.7, where α has minimal polynomial $f := x^4 + 2x^3 + 2x^2 + 2$. Note that K is not Galois over $\mathbb{Q}_2$ and K has Galois closure $K^{\text{gal}} = \mathbb{Q}_2(\beta)$ with LMFDB label 2.12.18.59, where β has minimal polynomial

$$x^{12} - 2x^{11} + 6x^{10} + 4x^9 + 6x^8 + 12x^7 - 4x^6 - 8x^3 + 16x^2 - 8$$

and $G := \text{Gal}(K^{\text{gal}}/\mathbb{Q}_2) \cong A_4$, the alternating group of order 12 (with LMFDB label 12.3). The inertia group $I := I(K^{\text{gal}}/\mathbb{Q}_2)$ of K^{gal} is the abelian group C_2^2 of order 4 (with LMFDB label 4.2). The wild inertia group of K^{gal} (i.e., the unique 2-Sylow subgroup of I) in this case is equal to I.

Examining the lattice of subgroups up to conjugacy given on the 12.3 homepage, we obtain the subfield lattice of $K^{\text{gal}}/\mathbb{Q}_2$, shown in Figure 5. It is clear from the subgroup lattice of 12.3 that $\text{Gal}(K^{\text{gal}}/K) \cong C_3$ and the extension $K/\mathbb{Q}_2$ is primitive. Also note that A_4 is solvable of length 2, so we find two intermediate fields E and F of K^{gal} such that $\text{Gal}(K^{\text{gal}}/E) \cong C_2^2$ and $\text{Gal}(K^{\text{gal}}/F) \cong C_2$. Computing fixed fields, we obtain the subfield lattice shown in Figure 5, where E and F are the p-adic fields given by $x^3 - x + 1$ and $x^6 + x^2 - 1$, respectively. The LMFDB labels for E and F are 2.3.0.1 and 2.6.6.1, respectively. We observe that the commutator subgroup of G is C_2^2, the subgroup corresponding to E. Thus E is the largest abelian subfield of K^{gal}, with Galois group $\text{Gal}(E/\mathbb{Q}) \cong G^{\text{ab}} \cong C_3$.

This is an interesting example since K is the only degree 4 extension of $\mathbb{Q}_p$, for any p, which has Galois group $\text{Gal}(K^{\text{gal}}/\mathbb{Q}_2) \cong A_4$.
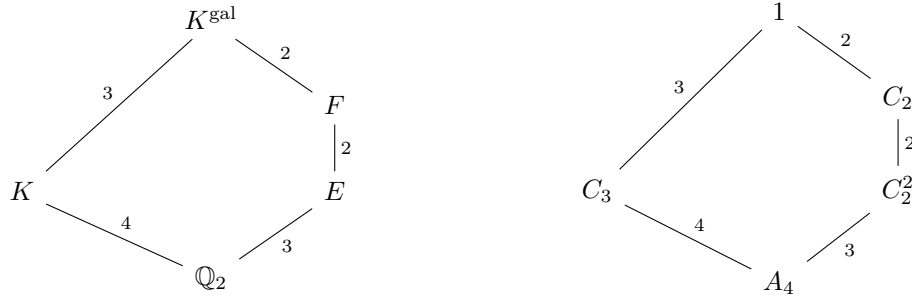

FIGURE 5. The subfield lattice of the Galois closure of the p-adic field with LMFDB label 2.4.6.7 and the subgroup lattice of its Galois group A_4.

EXAMPLE 6.4. Let G be the Sato-Tate group $J(O)$ with LMFDB label 1.4.F.48.48a. Let G^0 be the connected component of the identity and G/G^0 be the component group of G. As noted on its homepage, $J(O)$ has the largest component group ($C_2 \times S_4$, LMFDB label 48.48, with order 48) among Sato-Tate groups of abelian surfaces over number fields.

Let A be an abelian variety of dimension $g \leq 3$ defined over a number field k. Let K be the minimal extension of k over which all endomorphisms of A are defined, i.e., such that $\text{End}(A_K) = \text{End}(A_{\bar{k}})$. By [**FKRS12**, Proposition 2.17], then $\text{ST}_A / \text{ST}_A^0 \cong \text{Gal}(K/k)$, where ST_A is the Sato-Tate group of A.

Let $C : y^2 = x^6 - 5x^4 + 10x^3 - 5x^2 + 2x - 1$ considered over $\mathbb{Q}$, and let $A = \text{Jac}(C)$. As shown in [**FKRS12**], the variety A has Sato-Tate group $J(O)$, realizing this group over $\mathbb{Q}$, and the endomorphisms of A are defined over the number field $K = \mathbb{Q}(\sqrt{-2}, \sqrt{-11}, a, b)$ where

$$a^3 - 7a + 7 = 0 \qquad \text{and} \qquad b^4 + 4b^2 + 8b + 8 = 0 \,.$$

Applying the above proposition to this example, we have

$$C_2 \times S_4 \cong G/G^0 \cong \text{Gal}(K/k) \,.$$

14

However, as shown in [**FKRS12**, Table 4], the curve C can also be used to realize 24 other Sato-Tate groups by varying the base field. In other words, by taking a number field L with $k \subseteq L \subseteq K$ and considering the base change A_L, we can obtain other Sato-Tate groups. For instance, examining [**FKRS12**, Table 8] we see that $J(T)$ (LMFDB label `1.4.F.24.13a`) is the unique Sato-Tate group occurring in genus 2 with component group $C_2 \times A_4$. Computing the fixed field L of $C_2 \times A_4 \leq C_2 \times S_4$, we find that $L = \mathbb{Q}(\sqrt{-11})$. Thus the base change A_L has Sato-Tate group $J(T)$, realizing another of the 52 possible Sato-Tate groups.

With further calculation, one can use other base changes of A to realize other Sato-Tate groups whose component groups are subgroups of $C_2 \times S_4$ by considering the lattice of subgroups given on the homepage for $C_2 \times S_4$ and computing fixed fields.

Galois theory can also be applied in a geometric context. The equivalence of categories between function field extensions in one variable and non-singular projective curves (see [**Sta18**, Tag 0BXX], for instance) allows us to study non-constant morphisms of curves by examining the associated extension of function fields. We consider an example coming from the Families of higher genus curves with automorphisms collection, available at `https://www.lmfdb.org/HigherGenus/C/Aut/`.

EXAMPLE 6.5. Consider the refined passport with label `3.168-42.0.2-3-7`. It corresponds to a unique topological equivalence class of morphisms $X \to X/\operatorname{Aut}(X) \cong \mathbb{P}^1$, where X is the Klein quartic curve, a genus 3 curve with the largest possible number of automorphisms for its genus. (Such curves are known as *Hurwitz curves*.) It has automorphism group $\operatorname{Aut}(X) \cong \operatorname{PSL}_2(\mathbb{F}_7)$ (which has LMFDB label `168.42`), and all automorphisms are defined over the field $K = \mathbb{Q}(\zeta_7)$. By examining the lattice of subgroups of $\operatorname{PSL}_2(\mathbb{F}_7)$ given on its homepage and applying the Galois correspondence, we can find all intermediate covers Y with $X \to Y \to X/\operatorname{Aut}(X) \cong \mathbb{P}^1$.

Computationally, this can be accomplished using Magma's `CurveQuotient` command [**BCP97**]. Calling this on each subgroup of $\operatorname{PSL}_2(\mathbb{F}_7)$ in turn, we find three intermediate covers Y of genus 1, corresponding to the subgroups of $\operatorname{PSL}_2(\mathbb{F}_7)$ isomorphic to C_2, C_3, and C_4. (All other quotients by non-trivial subgroups of $\operatorname{Aut}(X)$ result in genus 0 curves.) Each of these curves can be equipped with the structure of an elliptic curve by taking as the origin of the group law the image of $(1:0:0)$ under the appropriate quotient map. With some further computation (see [**Elk99**, Equation 2.10]), one can show that each of these elliptic curves is isomorphic to the curve $E : y^2 = 4x^3 + 21x^2 + 28x$ over K. From this we observe that $\operatorname{Jac}(X)$ decomposes as E^3 up to isogeny.

6.3. Modular curves and subgroups of $\mathbf{GL_2(\mathbb{Z}/N\mathbb{Z})}$. The LMFDB currently has a database of modular curves, available at `https://www.lmfdb.org/ModularCurve/Q/`. We conclude this section by describing how properties of modular curves can be deduced from their corresponding subgroups of $\operatorname{GL}_2(\mathbb{Z}/N\mathbb{Z})$. For a more comprehensive exposition of modular curves and subgroups of $\operatorname{GL}_2(\widehat{\mathbb{Z}})$, see [**RSZB22**] or [**Zyw22**].

Let E be an elliptic curve over $\mathbb{Q}$ and $E[N]$ be its N-torsion subgroup for each $N \in \mathbb{Z}_{\geq 1}$. The absolute Galois group $G_{\mathbb{Q}} := \operatorname{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts on $E[N]$, and since $E[N] \cong (\mathbb{Z}/N\mathbb{Z})^2$ as abelian groups, we obtain a representation

$$\rho_{E,N} : G_{\mathbb{Q}} \to \operatorname{Aut}(E[N]) \cong \operatorname{GL}_2(\mathbb{Z}/N\mathbb{Z}).$$

By choosing compatible bases, we can take the inverse limit and package these together as a single representation

$$\rho_E : G_{\mathbb{Q}} \to \varprojlim_N \operatorname{GL}_2(\mathbb{Z}/N\mathbb{Z}) = \operatorname{GL}_2(\widehat{\mathbb{Z}}).$$

If E does not have complex multiplication, then Serre's Open Image Theorem [**Ser72**] implies that the image of ρ_E is an open subgroup of $\operatorname{GL}_2(\widehat{\mathbb{Z}})$, hence has finite index. Given an open subgroup H of $\operatorname{GL}_2(\widehat{\mathbb{Z}})$, one can

define the modular curve X_H whose K-points parametrize elliptic curves E/K such that $\text{img}(\rho_E) \subseteq H$ (up to conjugation). (For a precise definition of X_H, see [**RSZB22**, Section 2.3] or [**Zyw22**, Section 3].)

For each $N \in \mathbb{Z}_{\geq 1}$, let $\pi_N : \text{GL}_2(\widehat{\mathbb{Z}}) \to \text{GL}_2(\mathbb{Z}/N\mathbb{Z})$ be the projection map. Every open subgroup $H \leq \text{GL}_2(\widehat{\mathbb{Z}})$ contains $\ker(\pi_N)$ for some N, and the smallest such $N \in \mathbb{Z}_{\geq 1}$ is called the *level* of H. If H contains $\ker(\pi_N)$, then H can be recovered from $\pi_N(H)$ as $H = \pi_N^{-1}(\pi_N(H))$. Thus to store H on a computer, one can simply store generators for $\pi_N(H)$ where N is the level of H.

Although this is a database of modular curves, much of the geometric data is computed group theoretically. Let $H \leq \text{GL}_2(\widehat{\mathbb{Z}})$ be an open subgroup of level N. Letting $\Gamma_H := \pm\pi_N(H) \cap \text{SL}_2(\mathbb{Z}/N\mathbb{Z})$, then one can determine the number of elliptic points and cusps of X_H by studying the action of the matrices

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

on the coset space $\Gamma_H \backslash \text{SL}_2(\mathbb{Z}/N\mathbb{Z})$. This data can in turn be used to compute the genus of X_H. We further illustrate this connection in the example below.

EXAMPLE 6.6. Consider the modular curve $X_0(6)$ of level 6 with LMFDB label `6.12.0.a.1`. As a containment $H \subseteq H'$ of open subgroups of $\text{GL}_2(\widehat{\mathbb{Z}})$ induces a morphism $X_H \to X_{H'}$ of modular curves, we have morphisms $X_0(6) \to X_0(2)$ and $X_0(6) \to X_0(3)$. On the homepage for $X_0(6)$, it is further claimed that $X_0(6)$ is the fiber product of $X_0(2)$ and $X_0(3)$ over $X(1)$—we explain how this can be determined group theoretically.

Let H_2, H_3, H_6 be the subgroups of $\text{GL}_2(\mathbb{Z}/2\mathbb{Z})$, $\text{GL}_2(\mathbb{Z}/3\mathbb{Z})$, $\text{GL}_2(\mathbb{Z}/6\mathbb{Z})$ corresponding to $X_0(2)$ (with label `2.3.0.a.1`), $X_0(3)$ (with label `3.4.0.a.1`), and $X_0(6)$, respectively. Note that $\text{GL}_2(\mathbb{Z}/6\mathbb{Z})$ has label `288.851` and the subgroup $H_6 \cong C_2^2 \times S_3$ has subgroup label `288.851.12.c1.a1`. Taking the inverse image of H_2 and H_3 under the projection maps $\pi_{6,2} : \text{GL}_2(\mathbb{Z}/6\mathbb{Z}) \to \text{GL}_2(\mathbb{Z}/2\mathbb{Z})$, $\pi_{6,3} : \text{GL}_2(\mathbb{Z}/6\mathbb{Z}) \to \text{GL}_2(\mathbb{Z}/3\mathbb{Z})$, we obtain subgroups $\widetilde{H_2} := \pi_{6,2}^{-1}(H_2)$ with subgroup label `288.851.3.a1.a1` and $\widetilde{H_3} := \pi_{6,3}^{-1}(H_3)$ with subgroup label `288.851.4.b1.a1`.

Either by examining the (very large) subgroup lattice diagram of $\text{GL}_2(\mathbb{Z}/6\mathbb{Z})$ or by computing directly, we find that $\widetilde{H_2} \cap \widetilde{H_3} = H_6$ and $\langle \widetilde{H_2}, \widetilde{H_3} \rangle = \text{GL}_2(\mathbb{Z}/6\mathbb{Z})$. This shows that H_6 is the fiber product of $\widetilde{H_2}$ and $\widetilde{H_3}$ over $\text{GL}_2(\mathbb{Z}/6\mathbb{Z})$, and hence $X_0(6)$ is the fiber product of $X_0(2)$ and $X_0(3)$ over $X(1)$, as depicted in Figure 6.

$$
\begin{array}{ccc}
X_0(6) & \longrightarrow & X_0(2) \\
\downarrow & \ulcorner & \downarrow \\
X_0(3) & \longrightarrow & X(1)
\end{array}
\qquad\qquad
\begin{array}{ccc}
H_6 & \longrightarrow & \widetilde{H_2} \\
\downarrow & \ulcorner & \downarrow \\
\widetilde{H_3} & \longrightarrow & \text{GL}_2(\mathbb{Z}/6\mathbb{Z})
\end{array}
$$

FIGURE 6. The fiber product diagram for the modular curve $X_0(6)$, and the fiber product diagram of the corresponding subgroups of $\text{GL}_2(\mathbb{Z}/6\mathbb{Z})$.

References

[BCH$^+$] Greg Butler, John Cannon, Derek Holt, Alexander Hulpke, John McKay, and Gordon Royle. Database of transitive groups. https://magma.maths.usyd.edu.au/magma/handbook/text/781. 1

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993). 1, 3, 15

[BE99a] Hans Ulrich Besche and Bettina Eick. Construction of finite groups. *J. Symbolic Comput.*, 27(4):387–404, 1999. 3

[BE99b]   Hans Ulrich Besche and Bettina Eick. The groups of order at most 1000 except 512 and 768. *J. Symbolic Comput.*, 27(4):405–413, 1999. 3

[BE01]   Hans Ulrich Besche and Bettina Eick. The groups of order $q^n \cdot p$. *Comm. Algebra*, 29(4):1759–1772, 2001. 3

[BEO01]   Hans Ulrich Besche, Bettina Eick, and E. A. O'Brien. The groups of order at most 2000. *Electron. Res. Announc. Amer. Math. Soc.*, 7:1–4, 2001. 3

[BEO02]   Hans Ulrich Besche, Bettina Eick, and E. A. O'Brien. A millennium project: constructing small groups. *Internat. J. Algebra Comput.*, 12(5):623–644, 2002. 1

[Bre00]   Thomas Breuer. *Characters and automorphism groups of compact Riemann surfaces*, volume 280 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2000. 4

[CCN+85]   J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson. ATLAS *of finite groups.* Oxford University Press, Eynsham, 1985. Maximal subgroups and ordinary characters for simple groups, With computational assistance from J. G. Thackray. 1

[CH08]   John J. Cannon and Derek F. Holt. The transitive permutation groups of degree 32. *Experiment. Math.*, 17(3):307–314, 2008. 4

[CHM98]   John H. Conway, Alexander Hulpke, and John McKay. On transitive permutation groups. *LMS J. Comput. Math.*, 1:1–8 (electronic), 1998. 10

[Con10]   Marston Conder. Group actions on surfaces. https://www.math.auckland.ac.nz/~conder/BigSurfaceActions-Genus2to101-ByGenus.txt, 2010. 4

[DE05]   Heiko Dietrich and Bettina Eick. On the groups of cube-free order. *J. Algebra*, 292(1):122–137, 2005. 3

[DE12]   Heiko Dietrich and Bettina Eick. Addendum to "On the groups of cube-free order" [J. Algebra 292 (1) (2005) 122–137] [mr2166799]. *J. Algebra*, 367:247–248, 2012. 3

[DEP22]   Heiko Dietrich, Bettina Eick, and Xueyu Pan. Groups whose orders factorise into at most four primes. *J. Symbolic Comput.*, 108:23–40, 2022. 3

[Dok]   Tim Dokchitser. GroupNames. http://groupnames.org. 1

[Elk99]   Noam D. Elkies. The Klein quartic in number theory. In *The eightfold way*, volume 35 of *Math. Sci. Res. Inst. Publ.*, pages 51–101. Cambridge Univ. Press, Cambridge, 1999. 15

[FKRS12]   Francesc Fité, Kiran S. Kedlaya, Víctor Rotger, and Andrew V. Sutherland. Sato–Tate distributions and Galois endomorphism modules in genus 2. *Compositio Mathematica*, 148(5):1390–1442, 2012. 14, 15

[GAP21]   The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.11.1*, 2021. 1

[HEO05]   Derek F. Holt, Bettina Eick, and E. A. O'Brien. *Handbook of Computational Group Theory*. Discrete Mathematics and Its Applications. CRC Press, 2005. 8

[HR20]   Derek Holt and Gordon Royle. A census of small transitive groups and vertex-transitive graphs. *J. Symbolic Comput.*, 101:51–60, 2020. 4

[HRT01]   R. B. Howlett, L. J. Rylands, and D. E. Taylor. Matrix generators for exceptional groups of Lie type. *J. Symbolic Comput.*, 31(4):429–445, 2001. 4

[Hul]   Alexander Hulpke. Transitive groups library - a GAP package, version 3.0. https://www.gap-system.org/Packages/transgrp.html. 1

[Hul05]   Alexander Hulpke. Constructing transitive permutation groups. *J. Symbolic Comput.*, 39(1):1–30, 2005. 4

[JR17]   John W. Jones and David P. Roberts. Artin L-functions of small conductor. *Res. Number Theory*, 3:Paper No. 16, 33, 2017. 13

[NOVL04]   M. F. Newman, E. A. O'Brien, and M. R. Vaughan-Lee. Groups and nilpotent Lie rings whose order is the sixth power of a prime. *J. Algebra*, 278(1):383–401, 2004. 3

[O'B90]   E. A. O'Brien. The p-group generation algorithm. volume 9, pages 677–698. 1990. Computational group theory, Part 1. 3

[O'B91]   E. A. O'Brien. The groups of order 256. *J. Algebra*, 143(1):219–235, 1991. 3

[OPS08]   J. Opgenorth, W. Plesken, and T. Schulz. Carat, crystallographic algorithms and tables, Version 2.1b1. https://github.com/lbfm-rwth/carat/, Jul 2008. 4

[OVL05]   E. A. O'Brien and M. R. Vaughan-Lee. The groups with order p^7 for odd prime p. *J. Algebra*, 292(1):243–258, 2005. 3

[Roe]   David Roe. A fast nonisomorphism test for finite groups. In preparation. 5

[RSZB22]   Jeremy Rouse, Andrew V. Sutherland, and David Zureick-Brown. ℓ-adic images of Galois for elliptic curves over $\mathbb{Q}$ (and an appendix with John Voight). *Forum Math. Sigma*, 10:Paper No. e62, 63, 2022. With an appendix with John Voight. 15, 16

[RT98]   L. J. Rylands and D. E. Taylor. Matrix generators for the orthogonal groups. *J. Symbolic Comput.*, 25(3):351–360, 1998. 4

[Ser72]   Jean-Pierre Serre. Propriétés galoisiennes des points d'ordre fini des courbes elliptiques. *Invent. Math.*, 15(4):259–331, 1972. 15

[Sta18]   The Stacks Project Authors. *Stacks Project.* https://stacks.math.columbia.edu, 2018. 15

[Sut21]   Andrew V. Sutherland. Stronger arithmetic equivalence. *Discrete Anal.*, pages Paper No. 23, 23, 2021. 11

[Tan11]   O. Tange. GNU parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47, Feb 2011. 3

[Tay87]   Don Taylor. Pairs of generators for matrix groups. I. Cayley Bulletin 3, 1987. 4

[VRD09]   Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA, 2009. 3

[WWT⁺]   Robert Wilson, Peter Walsh, Jonathan Tripp, Ibrahim Suleiman, Richard Parker, Simon Norton, Simon Nickerson, Steve Linton, John Bray, and Rachel Abbott. Atlas of finite group representations - version 3. http://brauer.maths.qmul.ac.uk/Atlas/v3/. 1, 4

[Zyw22]   David Zywina. Explicit open images for elliptic curves over ℚ. https://arxiv.org/abs/2206.14959, 2022. 15, 16

Lewis Combes, School of Mathematics and Statistics, University of Sheffield, UK, S3 7RH
Email address: lmc577@proton.me

John Jones, Department of Mathematics and Statistical Sciences, P.O. Box 871804, Arizona State University, Tempe, AZ, 85287
Email address: jj@asu.edu

Jennifer Paulhus, Department of Mathematics and Statistics, Mount Holyoke College, South Hadley, MA, 01075
Email address: jpaulhus@mtholyoke.edu

David Roe, Massachusetts Institute of Technology, Department of Mathematics, 77 Massachusetts Ave., Bldg. 2-336 Cambridge, MA 02139, United States of America
Email address: roed@mit.edu

Manami Roy, Department of Mathematics, Lafayette College, 730 High St, Easton, PA 18042
Email address: royma@lafayette.edu

Sam Schiavone, Massachusetts Institute of Technology, Department of Mathematics, 77 Massachusetts Ave., Bldg. 2-336 Cambridge, MA 02139, United States of America
Email address: sam.schiavone@gmail.com