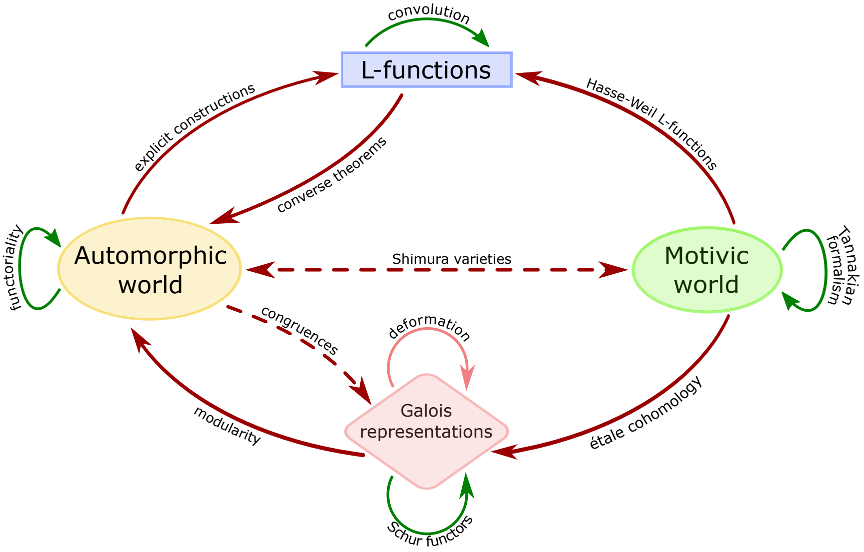
Crowdsourcing Mathematical Databases

Andrew V. Sutherland
Massachusetts Institute of Technology

Big Data 2026

September 3, 2026

The LMFDB universe



Motivation and desiderata

- Number theory has long been, in part, an experimental science.
  Data is a source of conjectures and counterexamples that lead to theorems.
- Exhaustive enumeration allows one to prove theorems and exposes holes (both in theory and in implementations) by finding all the special cases.
- The database should be easily accessible and comprehensible to as broad an audience as possible, serving both novices and experts.
- All data should have a clear and citable provenance: how it was computed, by whom, to what precision, and under what assumptions, if any.
- Search, aggregation, visualization, and API access tools are needed to maximize the utility of the data.

History and organization

- The LMFDB was first conceived at an AIM workshop in 2007.
  The first production release was launched in 2016.

- More than 100 researchers have contributed data, code, or both.
  All LMFDB software and the technology stack that supports it are open source.

- We have received substantial financial support from EPSRC, NSF, the Simons Foundation, and Renaissance Philanthropy.

- The LMFDB is overseen by a board of managing and associate editors.

- The LMFDB is currently hosted on servers running on Google Cloud Platform.
  A full mirror and development environment are hosted at MIT.

- In 2018 we migrated the entire database from MongoDB to PostgreSQL.
  In 2026 we added an MCP server to facilitate interactions with AI agents.

Technical specs

- About 4 TB of static data available for download, plus 1.2 TB of searchable data with 0.5 TB of indices on 2+ billion rows in 200+ PostgreSQL tables.
- There are two load-balanced web servers and a database server on GCP. A mirror of the development database serves as a staging area for updates.
- We use the Flask web framework on Python 3 via SageMath. All database access occurs via a custom interface layer built on psycopg2.
- Software development and version control are coordinated via GitHub. Our repository is public and the code is available under a GPLv2+ license.
- All software and data changes are reviewed on the development site before they are put into production.

Some challenges

- Data sets whose objects vary widely in complexity.
- Schema flexibility versus performance.
- Permanently labeling mathematical objects.
- Completeness and scope.
- On-demand computation.
- Rigor, reliability, reproducibility.
- Generating new data at scale.

The inverse Galois problem

Each irreducible polynomial $f \in \mathbb{Q}[x]$ defines a **number field** $K = \mathbb{Q}[x]/(f(x))$.
The **splitting field** L of f is the minimal extension of $\mathbb{Q} \subseteq \mathbb{C}$ containing all roots of f.
The **Galois group** G of f is $\mathrm{Aut}_{\mathbb{Q}}(L)$, which permutes the roots of f and can be identified with a transitive subgroup of S_d, where $d = \deg(f)$.

The **inverse Galois problem** (over $\mathbb{Q}$) asks whether every finite group G arises as the Galois group of a number field.

More precisely, given a positive integer d, one can ask which transitive permutation groups of degree d arise as Galois groups of degree-d number fields.

More explicitly, given a transitive permutation group G of degree d, one can ask for an explicit monic polynomial $f \in \mathbb{Z}[x]$ of degree d with Galois group G.

The IGP24 contest

Prior to this year, the inverse Galois problem over $\mathbb{Q}$ was known for $d \leq 23$ with one exception: the Mathieu group $M_2 3$. This exception was eliminated on August 9, 2026 by Huang–Jackson–Lee–Poonen–Pries–Zhang.

On June 15, 2026, we (John Jones, Jennifer Paulhus, David Roe, Terry Tao, and I), in collaboration with the Foundation for Science and AI Research (SAIR), launched a contest aimed at the inverse Galois problem over $\mathbb{Q}$ in degree 24.

When the contest started, the LMFDB contained number fields that realized only 79 of the 25,000 possible Galois groups of a degree-24 number field. Other large databases of number fields (such as the Klüners–Malle database) contained extensive data in degrees $d < 24$, but no examples in degree 24.

When phase one ended on August 15, 2026, all 25,000 groups had been realized (this occurred on August 8; the data was made public on August 23).

Takeaways and final thoughts

- Modern computer algebra systems (in this case, Magma) are very powerful and can play a crucial role in verification infrastructure.
- Layering complexity can lower the barrier to entry.
- Competition/evaluation/benchmarking is an attractive force.
- Scoring functions matter; this is the most important thing to get right.
- A diverse collection of (AI+human) teams outperforms any one team.
  The minimal number of teams needed to cover all 25,000 groups was 18.
- Future plans. . .