The L-functions and modular forms database project

John Cremona

THE UNIVERSITY OF WARWICK

Second CoDiMa Training School in Computational Discrete Mathematics
ICMS Edinburgh, 18 October 2016

EPSRC
Engineering and Physical Sciences Research Council

Overview

1. What is the LMFDB?
2. What are L-functions and why are they important?
3. Why the database? Why the website? How do they work?
4. How does the project operate in practice?

What is the LMFDB?

Since the early days of using computers in number theory, computations and tables have played an important part in experimentation, for the purpose of formulating, proving (and disproving) conjectures.

Until the WWW tables were hard to use (let alone to make) as they were only available in printed form, or on microfiche!

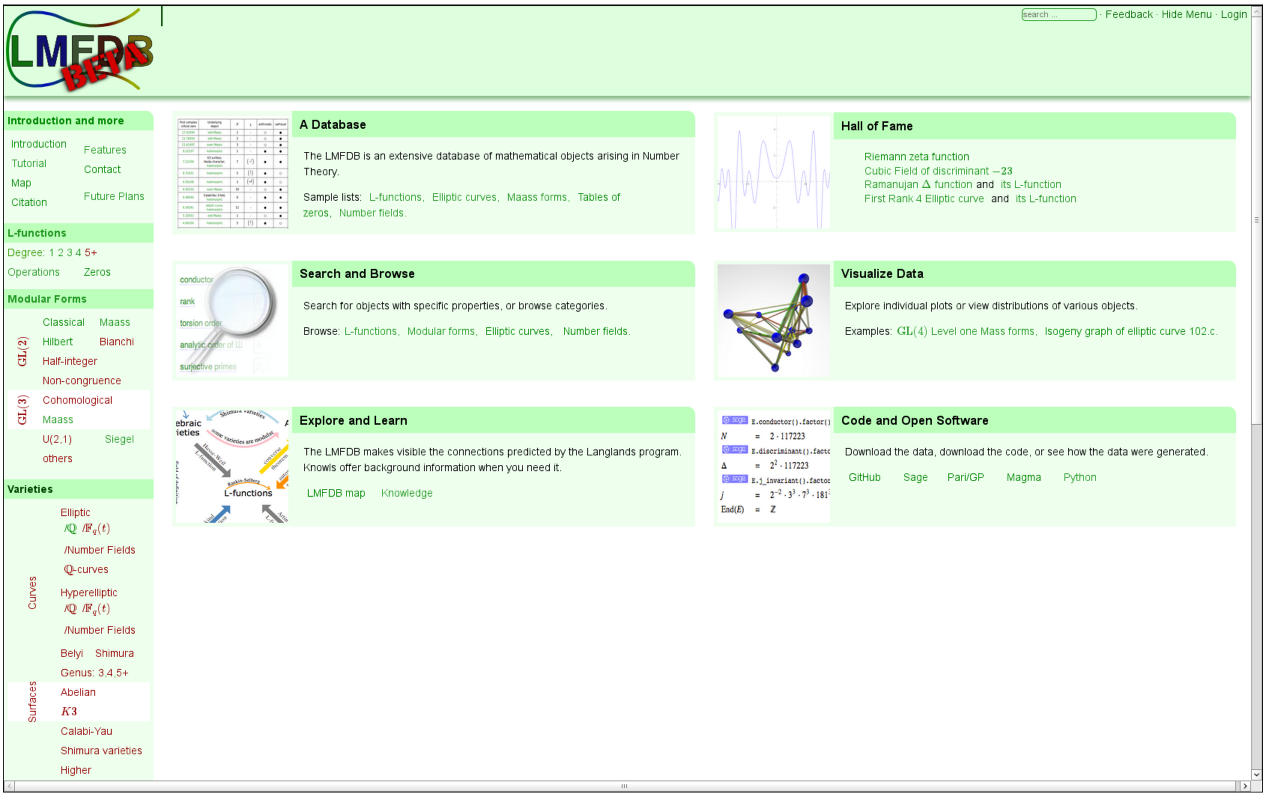
Example: the 1976 Antwerp IV tables of elliptic curves.

Even since the WWW, tables and databases have been scattered: to use them you had to know who to ask, download data, and deal with a wide variety of formats.

In some areas of number theory, such as elliptic curves, life is now much easier: Packages such as SAGE, MAGMA and PARI/GP contain elliptic curve databases (sometimes as optional add-ons as they are large); the internet makes accessing even "printed" tables much easier.

Now . . .

This is the LMFDB!



What are L-functions?

L-functions are at the heart of the LMFDB. What are they?

The simplest L-function is the **Riemann zeta function** $\zeta(s)$. This

- is a *complex analytic function* (apart from a pole at $s = 1$);
- has a *Dirichlet series* expansion over positive integers: $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ (valid when $\Re(s) > 1$);
- has an *Euler product* expansion over primes p: $\zeta(s) = \prod_p (1 - p^{-s})^{-1}$ (also when $\Re(s) > 1$);
- satisfies a *functional equation*: $\xi(s) = \pi^{-s/2}\Gamma(s/2)\zeta(s) = \xi(1 - s)$;
- has links to the *distribution of primes*.

L-functions: a definition

The definition of an L-function encapsulates these properties: it is a **complex function** with a **Dirichlet series** and an **Euler product** expansion which satisfies a **functional equation**.
There are other more technical axioms (by Selberg): see `http://www.lmfdb.org/knowledge/show/lfunction`.

Some of these properties are not yet proved for the L-functions in the database: this can be very hard!
For example, Wiles's proof of the modularity of elliptic curves over $\mathbb{Q}$ showed that elliptic curve L-functions really are L-functions in this sense; this is not yet known for elliptic curves defined over other fields.

Other expected properties of L-functions are not even known for $\zeta(s)$! For example, the *Riemann hypothesis*. . .

The Riemann Hypothesis

The Riemann Hypothesis states that all the “non-trivial” zeros of $\zeta(s)$ (excluding those coming trivially from poles of $\Gamma(s)$) are on the line $\Re(s) = 1/2$.

This was (part of) Hilbert’s 8th problem and is also one of the Clay Mathematics Institute Millennium Prize Problems, so a million dollars awaits the person who solves it.

What can a database say in relation to this problem? It can of course give the object a web page (`http://www.lmfdb.org/L/Riemann/`) which shows basic facts about it, and its graph along the critical line $1/2 + it$ to “show” the first few zeroes.
It can also store all the zeroes which have so far been explicitly computed: there are more than 10^{11} (**one hundred billion**) of them at `http://www.lmfdb.org/zeros/zeta/`, all computed to 100-bit precision (by David Platt of Bristol).

Degrees of L-functions

The Euler product for a general L-function has the form

$$L(s) = \prod_p L_p(s) = \prod_p P_p(p^{-s})^{-1}$$

where each $P_p(t)$ is a *polynomial*. These polynomials all have the same degree d, except for a finite number indexed by primes dividing an integer called the *level* of the L-function (denoted N) where the degree is smaller. The zeros of these polynomials are also restricted in a way depending on another parameter, the weight.

For $\zeta(s)$ we have $P_p(t) = 1 - t$ for all primes p; the degree is $d = 1$, and the level is $N = 1$.

L-functions of degree 1

There are other L-functions of degree 1 (and higher level N) which have been studied since the 19th century: **Dirichlet L-functions**. Their Dirichlet coefficients a_n are given by the values of a *Dirichlet character* $a_n = \chi(n)$, meaning that they are *multiplicative* and *periodic* with period N.

e.g. ($N = 4$) $L(\chi, s) = 1^{-s} - 3^{-s} + 5^{-s} - 7^{-s} + - \dots$
Dirichlet used these L-functions to prove his celebrated theorem about primes in arithmetic progressions: for any integer $N \geq 1$ and any a there are infinitely many primes $p \equiv a \pmod{N}$, provided that a and N are coprime.

This is a *complete* list of all degree 1 L-functions!

Where do L-functions come from?

A wide variety of mathematical objects have L-functions: algebraic number fields, algebraic varieties (including curves). There is a general term “motive” for objects which have L-functions.

In many cases, while we know how to define the L-function of a more complicated object, it has not yet proved that it actually satisfied the defining axioms for “L-function”. Even for elliptic curves over $\mathbb{Q}$, this would have been true until the mid-1990s; for elliptic curves over real quadratic fields such as $\mathbb{Q}(\sqrt{2})$ it was true until 2013!
These elliptic curves are now known to be modular [Siksek, Freitas and Le Hung: *Elliptic Curves over Real Quadratic Fields are Modular*, Inventiones Mathematicae (2015)]

L-functions of number fields

A number field is a finite extension of the rational field $\mathbb{Q}$, such as $\mathbb{Q}(\sqrt{2})$ or $\mathbb{Q}(i)$. Every number field K has an L-function called its **Dedekind zeta function** $\zeta_K(s)$, defined in a similar way to $\zeta(s) = \zeta_{\mathbb{Q}}(s)$ and with similar analytic properties.

Just as the analytic properties of $\zeta(s)$ imply number-theoretical statements about (for example) the distribution of primes, from the analytic properties of $\zeta_K(s)$ we can deduce statements about prime factorizations in the field K. For example, taking $K = \mathbb{Q}(e^{2\pi i/m})$ we can prove Dirichlet's Theorem on primes in arithmetic progressions this way.

Just as some properties of $\zeta(s)$ are not yet proved (RH) the same is true for $\zeta_K(s)$: the Generalized Riemann Hypothesis (GRH).

L-functions of curves

Algebraic curves have L-functions, whose degree depends on both the degree of the field over which the curve is defined, and the genus of the curve. So an elliptic curve over $\mathbb{Q}$ (which has genus 1) has a degree 2 L-function, elliptic curves over fields of degree d have L-functions of degree $2d$, and so on.

It is known how all degree 2 L-functions arise: apart from products of two degree 1 L-functions they all come from elliptic curves over $\mathbb{Q}$, or from (a special kind of) *modular form*. The insight of Weil, Taniyama, Shimura and others in the 1960s and 1970s was to realize that these two sources actually produce **the same** L-functions!
This is behind the famous theorem that “every elliptic curve is modular” (over $\mathbb{Q}$) from which Fermat’s Last Theorem was a consequence.

Higher degree L-functions

Even for degrees 3 and 4, we do not yet know all the sources of L-functions, and not all the connections have been proved.

I mentioned the recent result that elliptic curves defined over real quadratic fields (e.g. $\mathbb{Q}(\sqrt{5})$) are modular. This means that two sources of L-functions: one the one hand, elliptic curves over a field such as $\mathbb{Q}(\sqrt{5})$, and on the other hand *Hilbert modular forms* over the same field, actually produce the same L-functions.

By contrast, over *imaginary* quadratic fields (e.g. $\mathbb{Q}(\sqrt{-1})$) we conjecture, but cannot prove in general, that elliptic curves have L-functions also attached to a different kind of modular form, **Bianchi modular forms**. These can be computed and work is in progress in entering many examples into the LMFDB, even though they are not all known to "be modular" and hence have genuine L-functions.

Showing connections through the LMFDB

The LMFDB shows these connections between different objects with the same L-function, by linking its databases of (for example) Elliptic curves over real quadratic fields, and Hilbert modular forms over the same field. This is work in progress.

A difficulty we have encountered is rather typical of such a large project where many different individuals are providing data is to maintain *consistency of labelling* of objects. Over $\mathbb{Q}(\sqrt{5})$, the Hilbert modular forms were computed (in Magma) by John Voight, while the elliptic curves were computed (in Sage) by Alyson Deines. They use essentially the same naming convention, but the labels were not the same, and the latter were adapted to match.

What is the LMFDB, really?

It is

1. a database
2. a website

Both are currently hosted on machines installed at Warwick, funded by EPSRC (for development) and at Google, funded by a CompX grant from Dartmouth College (for the production site). The website was officially launched in May 2016 after several years of being beta-tested.

The LMFDB is also

3. a group of mathematicians who collaborate to make the first two happen

The database

We are using the open-source database software `mongodb`. This currently holds nearly 1TB of data and indices. The database consists of around 35 individual databases containing classes of mathematical objects (e.g. `elliptic_curves`, `number_fields`, `SL2Zsubgroups`) and other data used by the website such as `knowledge` which holds the contents of “knowls”. The data is indexed in various ways for faster searching. And, of course, backed up regularly. Each constituent database contains “collections” of records, and the records hold the data in a flexible format: additional data fields can be added later, for example.

Sample database entry

Example: the database "number_fields" contains just one collection "fields" for which a typical entry is


```
{u'_id': ObjectId('4cb80fdb5009fb52db0946b6'),
 u'class_group': u'',
 u'class_number': 1,
 u'coeffs': u'1,0,-1,1',
 u'degree': 3,
 u'disc_abs_key': u'00123',
 u'disc_sign': -1,
 u'galois': {u'n': 3, u't': 2},
 u'label': u'3.1.23.1',
 u'ramps': [u'23'],
 u'signature': u'1,1',
 u'unitsGmodule': [[3, 1]]}
```

Pros and cons

Using off-the-shelf software has plenty of advantages but will never be perfect for a mathematical project.
Like most mathematicians, even with a lot of computational experience I know almost nothing about databases–and I knew nothing at all before I joined the LMFDB.

`mongodb` data consists of strings or integers or floating point values, with strings as keys. Values can also be lists of these, **but** a serious deficiency for number-theoretic data is that the integers cannot be larger than C long ints, i.e. 2^{32}. This means that many fields which hold mathematical integers have to be stored as strings .
Similarly, rational number cannot be stored as such, or even as a list [*numerator*,*denominator*] if these could be large, so instead they are stored as strings.

Pros and cons

One good thing about `mongodb` for our project is that it has a good Python front-end `pymongo`, and we are using Python for everything else (the web server and the underlying mathematics, via Sage).

This makes it easy to learn to use. This is important since we want the barriers to new people joining our project to be as low as possible.

The website

The LMFDB website is many things:

- a shop window for the data
- a way to visualize the data and the connections between different objects
- a way to browse types of object
- a way to search for objects with specified properties
- a repository of knowledge through its “knowledge database”
- a source of data for downloading for further work

Trying to cater for several different audiences at once is not easy to get right!

Website software

The LMFDB website software is written in Python (see `https://github.com/LMFDB/lmfdb`) using the `flask` library to run the webserver using `html` templates, which in turn use the `Jinja` templating language.

Again, anyone contributing to the project who wants to do more than just donate data has to learn something about these.

This is not so hard: code written by beginners is reviewed by (slightly more) experienced peers, and subject to automated testing. But not enough!

Technical support (lack of)

The project would benefit greatly from having technical support staff. Our EPSRC grant does not provide this, so we are currently relying on “charitable contributions” of time. We would not be where we are now without the enormous contributions of one person: Harald Schilly, a doctoral student and freelance software developer from Vienna, who knows more than the rest of us put together about Python, `mongodb`, `flask`, and the rest.

We are about to hire a Research Software Engineer at Warwick, funded by the EU H2020 project OpenDreamKit (www.opendreamkit.org) to work on improving the interface between the LMFDB and Sage.

Homepages

Every object in the LMFDB has its own homepage.
These are created on the fly from templates.

Each homepage gives a view of the object (depending on its nature), highlighting its most important properties, with "breadcrumbs" to show its position in the whole.

A "related objects" box provides links between . . . related objects. For example, from the pages of an elliptic curve, or a number field, or a modular form there are links to the associated L-function.

Where possible, on the home page of an object it is possible to see code which will re-create the object in a standard number-theoretical package (Sage, Pari/GP or Magma) and work with it there. In this way, the LMFDB can be used by students learning a subject who wish to work out their own examples.

Searching and browsing

For each class of objects in the LMFDB there is a “Browse and Search” page.

The Browse section is intended to be usable by people who know nothing of the underlying theory but want to browse through examples without having to type anything or have technical knowledge.

The Search section is more for experts looking for a specific object (possibly by its label) or for an object with certain properties: *“a number field with Galois group C_5 ramified only at $p = 5$”*, or *“an elliptic curve with rank 2 and non-trivial Tate-Shafarevich group”*, or *“a classical modular form of weight 12 and level 12”*.

Knowledge and knowls

The “knowledge” aspect of the LMFDB exists in the first place as a glossary of technical terms used on the web pages, so the pages themselves do not get cluttered up, and there is consistency between pages on basic definitions.

The mechanism which serves these is the “knowl”, invented at an LMFDB workshop and implemented by Harald Schilly. The text expands within the page and can be dismissed, without any need for “pop-ups” or new pages.

These can be used anywhere on the web (for instructions see the knowl about knowls!)– for example, I use them on my own web page of preprints and publications to display abstracts of papers (`http://homepages.warwick.ac.uk/staff/J.E.Cremona/papers/index.html`).

The content of knowls can be edited by any project member (who has a login), and is itself stored in the database.

The LMFDB as a collaborative project

The LMFDB was first conceived at an AIM workshop in 2007, funded by an NSF FRG grant 20018-2012 and by EPSRC 2013-2019. It holds regular workshops, funded by these and other grants, which are run along the lines of AIM workshops: few talks and a lot of hard work.

The AIM connection remains strong: both Brian Conrey (Director of AIM) and David Farmer (Director of Programs at AIM) are number theorists who have been intimately connected with the project from the start; BC is a co-investigator on the EPSRC grant and DF a "project partner".
Other partners are Fernando Rodriguez-Villegas (ICTP), William Stein (Washington) and Mike Rubinstein (Waterloo).
We have Editorial and Managements Boards, but essentially all decisions are made by consensus at workshops.

Collaboration

Since the LMFDB encompasses such a wide range of mathematics, it is essential to have an equally wide range of mathematical expertise involved in the project.

Many of the collaborators listed at `http://www.lmfdb.org/acknowledgment#individuals` have contributed not by coding for the website but by providing the data (without which the project would be nothing!).

More contributors are always welcome. The best way to start is to come to a workshop: there are several of these each year. Longer activities are also important: in 2011 a semester-long programme at MSRI on *“Arithmetic Statistics”* included much LMFDB activity, as did semester-long programme on *“Computational Aspects of the Langlands Program”* at ICERM in 2015, for which all eight members of the Organizing Committee were LMFDB contributors.

Thank you!