Chapter 10

API solutions at zbMATH Open

Matteo Petrera, Fabian Müller, Moritz Schubotz, and Olaf Teschke

1 Introduction

Since January 2021 zbMATH Open[1] is open for public access. For the mathematics community this means open access to the available literature from anywhere in the world without subscription or authentication. Additionally, we are spending efforts to connect the open data of zbMATH Open with other information systems, collaborative platforms, and funding agencies, as outlined in [3, 6]. We expect that our commitment in disseminating mathematics research literature will considerably increase both the range of our target audience and the visibility of mathematics.

Recently we developed Application Programming Interface (API) solutions to facilitate and optimize the open access to mathematical research data. The main purpose of this contribution is to provide some details about these developments, thus extending our previous publications [5, 6].

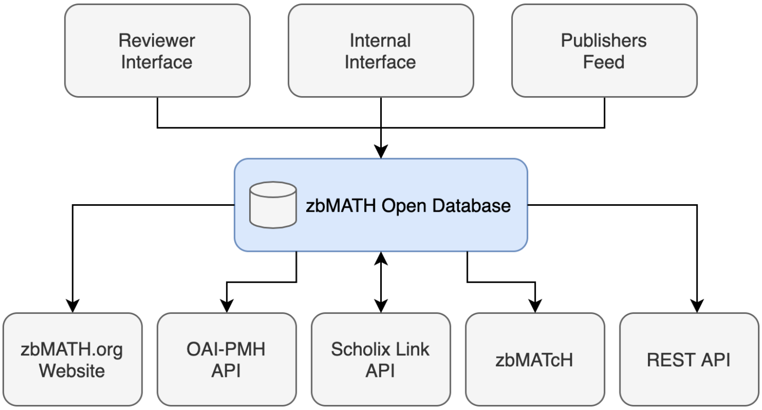


Figure 1. Overview of the zbMATH Open database and its associated data flows.

It is worth to sketch in Figure 1 a conceptual overview of our services in order to illustrate the current and future state of zbMATH Open. This will help the reader in understanding the overall structure of both zbMATH Open and the paper itself. In

[1] https://zbmath.org

Figure 1 we represent the data entering and leaving our database. The boxes called 'Reviewer Interface', 'Internal Interface', 'Publishers Feed', and 'zbMATH.org Website' show well-established components of zbMATH Open and are outside the scope of this paper. The box called 'OAI-PMH API' refers to the harvesting API that was released in April 2021. This will be shortly discussed in Section 5 and we refer to [6] for further and more technical details. The box 'Scholix Link API' refers to an API that is in the staging phase and will be deployed very soon. This will be discussed in Section 2, but we mention that a preliminary version of it has been already presented in [5]. Section 4 is devoted to the 'zbMATH Citation Matcher' (labelled 'zbMATcH' in Figure 1), that consists of an HTML interface designed for manual use, as well as an API. A part of the zbMATH Citation Matcher is a MathOverflow endpoint, discussed in Section 3. Finally, an open 'REST API' is currently in the planning stage. Some information about it will be provided in Section 6.

The motivation behind the recent implementation of APIs at zbMATH Open is twofold. On the one hand, we want to provide the community with efficient tools to benefit from the open access to our data. On the other hand, we wish to expose the dynamic interaction between our bibliographic data and those coming from other digital resources. Both motivations offer an opening for potential research opportunities, both on our part and on the part of any institutions interested in our data.

The potential users of our API endpoints may be clustered into five categories:

(1) Bibliographic consumers (MathOverflow, Wikimedia, arXiv, etc.) displaying references to scientific publications;

(2) Aggregators (research data infrastructures, OpenAIRE, SemanticScholar, etc.) extracting data to be then standardized for specific data models;

(3) Archives (research/software digital archives, etc.) typically interested to digitally preserve optimised and standardised flows of data;

(4) Search engines, e.g., Google, implementing the OpenSearch standard;

(5) Researchers interested in the literature for research purposes.

Let us remark that, before zbMATH became an open access web service, the main category of users interested in our product was represented by (5), namely researchers needing access to the literature. It is clear that with zbMATH becoming open the target audience has expanded incredibly.

To conclude, our main goals for the future can be summarised as follows:

- To be a modern and open reference tool for research data in mathematics;
- To promote a functional connection with external information systems of research data;
- To maximise the visibility and discoverability of research in mathematics.

2 Scholix Link API

The Scholix Link API is currently in the staging phase and it will be deployed very soon. A beta version of it is available for public testing[2] and we recently presented it in [5] in occasion of the DISCO2021 workshop at the ACM/IEEE Joint Conference on Digital Libraries.

The main purpose of this API is to document the interconnections (more specifically, *links*) between zbMATH Open and external platforms (called *partners*) which display and use documents indexed in the zbMATH Open database. Potential partners are (see Figure 2):

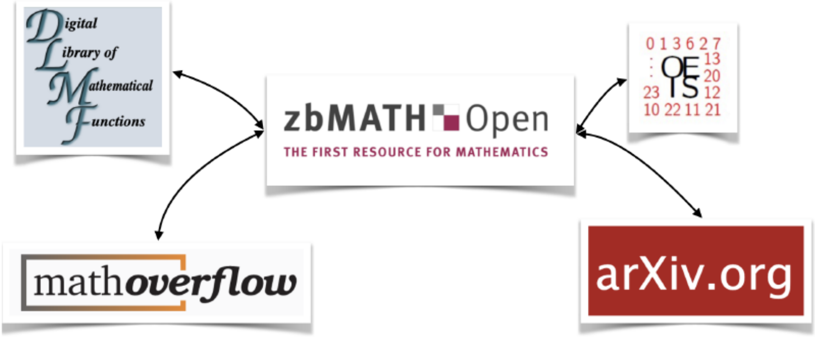

Digital Library of Mathematical Functions
zbMATH Open
THE FIRST RESOURCE FOR MATHEMATICS
OEIS
mathoverflow
arXiv.org

Figure 2. zbMATH Open and some of its partners.

- MathOverflow.[3] This is a question-and-answer platform for mathematics that is part of the StackExchange Network.[4] In a previous collaboration, zbMATH Open and MathOverflow added the possibility to cite entries of zbMATH Open in a MathOverflow post directly; see [4] and Section 3;
- arXiv.[5] arXiv is one of the most used open-access repositories of electronic preprints in mathematics. Roughly 250,000 zbMATH Open records contain links to specific arXiv preprints that were added manually, matched algorithmically, or provided by the publishers;
- Online Encyclopedia of Integer Sequences.[6] This a renowned online database of sequences of numbers launched in November 2010. It currently contains more

[2] https://zblink.formulasearchengine.com/links_api

[3] https://mathoverflow.net

[4] https://stackexchange.com

[5] https://arxiv.org

[6] https://oeis.org

than 340,000 sequences, each of them with its own list of metadata: first terms of the sequence, formulas for generating the sequence, references to books, articles, and scholarly links where the sequences have appeared, etc.;

- Digital Library of Mathematical Functions.[7] Please see Section 2.1 for further details.

Search engines or researchers from mathematics or the field of bibliometric research can use this API to present and use the search results. Furthermore, the source code of our API has been released in the form of a public Python package,[8] so that any interested user can use it for similar purposes in any context where the interconnection between bibliographic data and links has to be studied and documented. In this way, we hope to serve the needs of a wide range of users.

2.1 DLMF as a zbMATH Open partner

Among the possible platforms that interact with zbMATH Open, we selected the Digital Library of Mathematical Functions (DLMF) as a first partner. In addition to being an important reference tool for mathematicians, DLMF offers a relatively small bibliographic catalog and therefore has been very well suited for testing our API.

DLMF is a well-established web resource that enlarges and translates the classical 'Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables,'[9] edited by M. Abramowitz and I. A. Stegun in 1964, into a modern and functional digital library. As the title of the original book inspiring this web service suggests, DLMF is a digital handbook about theoretical and computational aspects of special functions. Its primary purpose is to provide a modern reference tool for researchers in mathematics, physical sciences, and engineering. It contains hundreds of definitions and theorems, presented with a standardised notation, together with tables, figures, and references to peer-reviewed papers and books. It was published online in May 2010 and is continuously maintained, reviewed, and updated ever since. Indeed, the field of special functions still receives great attention from the mathematics community, and new contributions enrich the contents of the library year by year.

DLMF presents its contents in 36 chapters, and the bibliography currently consists of almost 3,000 references[10], out of which about 75% are directly linked to zbMATH Open.[11]

[7] https://dlmf.nist.gov

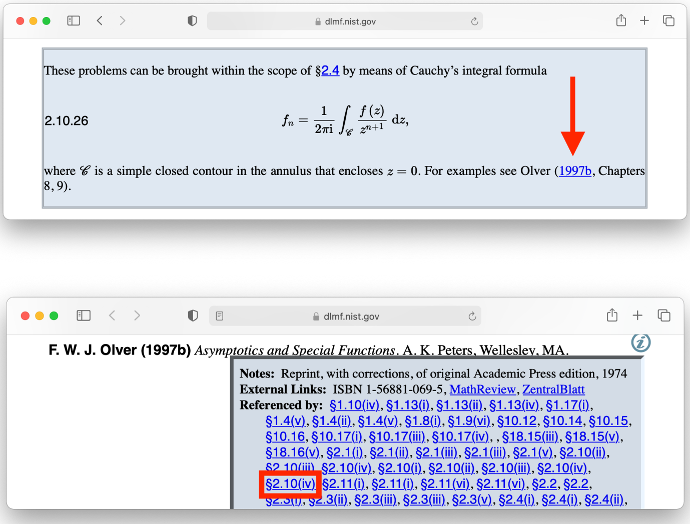
[8] https://github.com/zbMATHOpen/linksApi

[9] https://zbmath.org/0171.38503

[10] https://dlmf.nist.gov/bib

[11] The remaining 25% of publications not linked to zbMATH Open refer to documents not indexed in the zbMATH Open database.

Before providing more details about our Scholix Link API, let us mention a few details about the links' structure we are interested in. Each reference in the DLMF bibliography may be cited many times in the DLMF pages. Each of these instances carries its own link to zbMATH Open. For example, the book 'Asymptotics and special functions' by F. W. J. Olver (Reprint, 1997; Zbl 0982.41018)[12] is referenced 332 times. Each citation uniquely defines a link to zbMATH Open. An example of one of these links is: https://dlmf.nist.gov/2.10#iv.p2 (see Figure 3). In this case, Olver's book is referenced in Part 2 of § 2.10 (iv) with title 'Taylor and Laurent Coefficients: Darboux's Method.' In Figure 3, we also see that the § 2.10 (iv) is cited 3 times. Each instance corresponds to a link that points to a different destination site in the DLMF library. The highlighted § 2.10 (iv) points to what we see in the first screenshot of Figure 3.



These problems can be brought within the scope of §2.4 by means of Cauchy's integral formula

$$f_n = \frac{1}{2\pi \mathrm{i}} \int_{\mathscr{C}} \frac{f(z)}{z^{n+1}} \, \mathrm{d}z, \tag{2.10.26}$$

where $\mathscr{C}$ is a simple closed contour in the annulus that encloses $z = 0$. For examples see Olver (1997b, Chapters 8, 9).

F. W. J. Olver (1997b) *Asymptotics and Special Functions*. A. K. Peters, Wellesley, MA.

Notes: Reprint, with corrections, of original Academic Press edition, 1974
External Links: ISBN 1-56881-069-5, MathReview, ZentralBlatt
Referenced by: §1.10(iv), §1.13(i), §1.13(ii), §1.13(iv), §1.17(i), §1.4(v), §1.4(ii), §1.4(v), §1.8(i), §1.9(vi), §10.12, §10.14, §10.15, §10.16, §10.17(i), §10.17(iii), §10.17(iv), , §18.15(iii), §18.15(v), §18.16(v), §2.1(i), §2.1(ii), §2.1(iii), §2.1(iii), §2.1(v), §2.10(ii), §2.10(iii), §2.10(iv), §2.10(i), §2.10(ii), §2.10(iii), §2.10(iv), §2.10(iv), §2.11(i), §2.11(i), §2.11(vi), §2.11(vi), §2.2, §2.2, §2.3(i), §2.3(ii), §2.3(iii), §2.3(iii), §2.3(v), §2.4(i), §2.4(i), §2.4(ii),

Figure 3. A reference in DLMF, available at https://dlmf.nist.gov/bib/O (below), and a link to it, https://dlmf.nist.gov/2.10#iv.p2 (above).

The underlying dataset of the API has been generated by scraping the DLMF bibliography. For this purpose we developed an auxiliary Python open package.[13] This package is supposed to work for any zbMATH Open partner hosted in the Scholix Link API and has two main functionalities:

[12] https://zbmath.org/0982.41018
[13] https://github.com/zbMATHOpen/Update_Links

(1) Initialise the database of the API with data of a given partner. For those partners for which datasets need to be created from scratch, we included the corresponding scraping scripts;

(2) Update the initial database, thus add new links, delete links that no longer exist and edit links that have been changed.

In the case of DLMF, our auxiliary package creates a dataset containing about 2,000 references (indexed at zbMATH Open) and almost 7,000 distinct links. In this framework, the links are objects belonging to the *source* (of a given partner; DLMF in the present case), and records of zbMATH Open are objects belonging to the *target*.

2.2 Endpoints and response body

The current version of the API offers twelve endpoints:

- `GET /link`. It retrieves links for given zbMATH Open objects.
- `DELETE /link/item`. It deletes a link from the database.
- `POST /link/item`. It creates a new link related to a zbMATH Open object.
- `GET /link/item`. It checks existing relations between a given link and a given zbMATH Open object.
- `PATCH /link/item`. It edits an existing link.
- `GET /link/item/{doc_id}`. It retrieves links for a given zbMATH Open object.
- `GET /partner`. It retrieves data of a given zbMATH Open partner.
- `PUT /partner`. It edits data of a given zbMATH Open partner.
- `POST /partner`. It creates a new partner related to zbMATH Open.
- `GET /source`. It produces a list of all links of a given zbMATH Open partner.
- `GET /statistics/msc`. It shows the occurrence of primary MSC codes[14] (2-digit level) of zbMATH Open objects in the set of links of a given partner.
- `GET /statistics/year`. It shows the occurrence of years of publication of zbMATH Open objects in the set of links of a given partner.

Our JSON response body is modeled on the Scholix metadata schema.[15] This also explains the reason of the name ‘Scholix Link API’ for this service. The models used to pack the data are explicitly reported in the API web interface. It is worth recalling that Scholix is a well-established framework to exchange information between data

[14] Mathematics Subject Classification Scheme 2020, https://msc2020.org

[15] https://github.com/scholix/schema/releases/tag/3.0

and literature links. The schema's architecture is designed to allow for bulk exchange of link information, which contains all necessary data to keep track of bibliographic parameters identifying scholarly links.

2.3 Analysis of DLMF data

Based on our available DLMF dataset, it is possible to draw some conclusions:

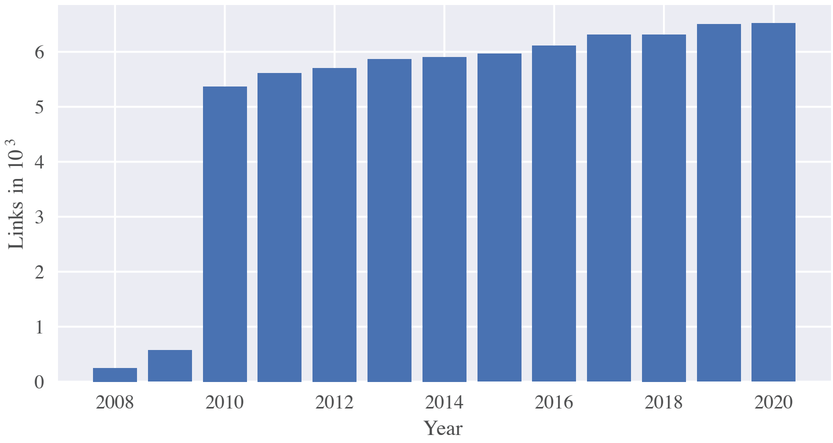


Figure 4. Growth of the links between DLMF and zbMATH Open.

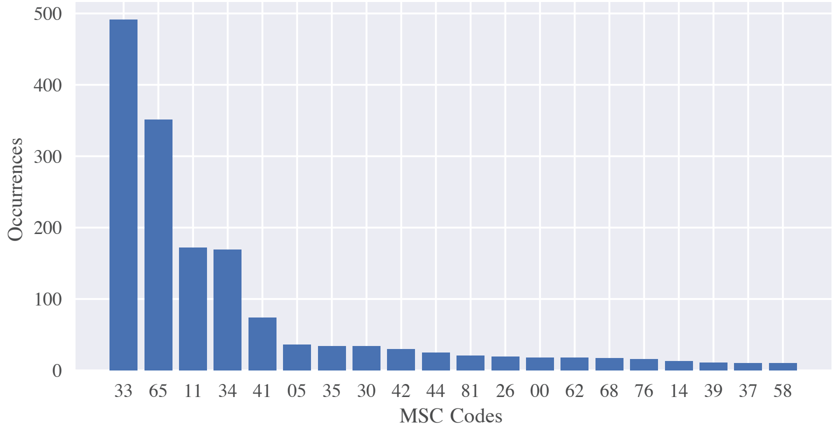


Figure 5. Distribution of primary 2-digit MSC codes in the DLMF dataset.

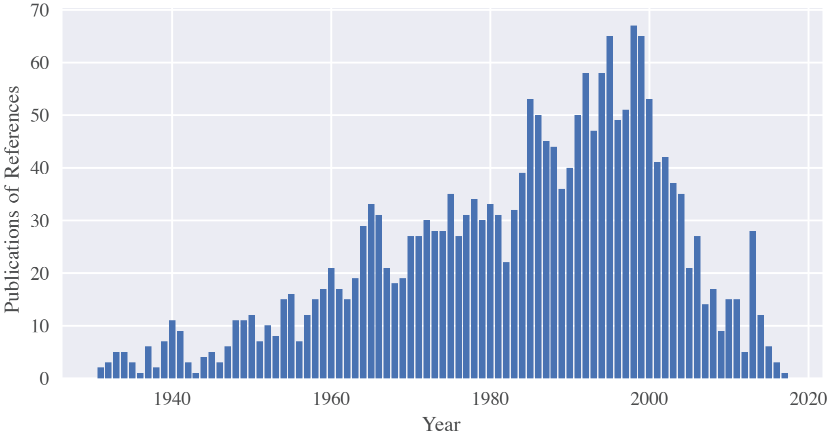


Figure 6. Distribution of years of publication of references in the DLMF dataset.

- In the JSON response body of our `GET` methods, one can see that each link is equipped with a publication date. This date refers to the date the link itself has been added in the DLMF bibliography. We scraped the historical bibliography between 2008 and 2020 and found the growth numbers depicted in Figure 4. Clearly, the growth of population of references changed drastically in 2010, the year when DLMF started officially.
- The two statistics routes show results concerning the distribution of primary MSC codes (2-digit level) and years of publication of the references in the current dataset. As one may expect, the most frequently cited primary MSC codes are (see Figure 5 for more details):

  - 33 (Special functions), with 491 documents;
  - 65 (Numerical analysis), with 351 documents;
  - 11 (Number theory), with 172 documents.

  A byproduct of this simple analysis confirms the consistency of our MSC tagging system over time.
  On the other hand, the most frequent years of publication of the cited references in the dataset are (see Figure 6 for more details):

  - 1998, with 67 documents;
  - 1999, with 65 documents;
  - 1995, with 65 documents.

Looking at Figure 6 we could infer that the DLMF bibliography suffers from a delay in updating its references. More precisely, the fact that the maximum peak is centered at the end of the 90s makes us think of some kind of difficulty in identifying relevant references referring to the last twenty years.

- The references in the current DLMF dataset which have the most citations are:
  - F. W. J. Olver, Asymptotics and special functions. Wellesley, MA: A K Peters (1997; Zbl 0982.41018)[16]: 332 citations;
  - M. Abramowitz (ed.) and I. A. Stegun (ed.), Handbook of mathematical functions with formulas, graphs and mathematical tables. Washington: U.S. Department of Commerce. (1964; Zbl 0171.38503)[17]: 118 citations;
  - A. Erdélyi et al., Higher transcendental functions. Vol. I. New York: McGraw-Hill Book Co. (1953; Zbl 0051.30303)[18]: 110 citations.

In Figure 7 one can see the references, identified by Zbl code, with more than 50 citations.

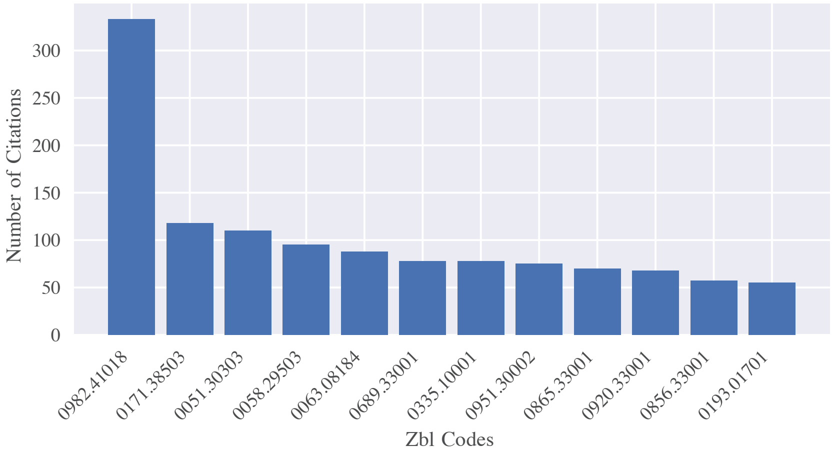


Figure 7. References (identified by Zbl code) in the DLMF dataset cited more than 50 times.

2.4 Usage

The Scholix Link API with its first partner DLMF represents a tool that can be used in various ways and contains many features that help the research process. Here, we

[16] https://zbmath.org/0982.41018
[17] https://zbmath.org/0171.38503
[18] https://zbmath.org/0051.30303

present concrete usage instances where a user of either DLMF or zbMATH Open can benefit from the service:

- A DLMF user can access all bibliographic resources indexed at zbMATH Open relating to a specific topic of interest. This may help to get a consistent overview of the scientific development of the topic itself.
- A researcher interested in a publication indexed at zbMATH Open can use our API to verify if and possibly where that publication is cited in DLMF. A search of this type can also be very diversified thanks to the filters that our routes offer. For example, one might be interested in identifying which DLMF links are related to a particular MSC code or a particular author. This means that a targeted use of our API can allow a detailed bibliographic search that otherwise would not be possible.
- A researcher more interested in the history of mathematics can use our API to trace the bibliography related to a certain topic covered in DLMF and observe the historical development of the topic itself in terms of the literature related to it. Such research can be very rich and diverse. It is sufficient to think that in the field of special functions there are classical topics, such as the ‘gamma function’ or ‘elliptic integrals’, which have a long history behind them.

When other partners will be included in our API, the covered spectrum will expand considerably, thus providing the user with a complete and flexible bibliographic searching tool.

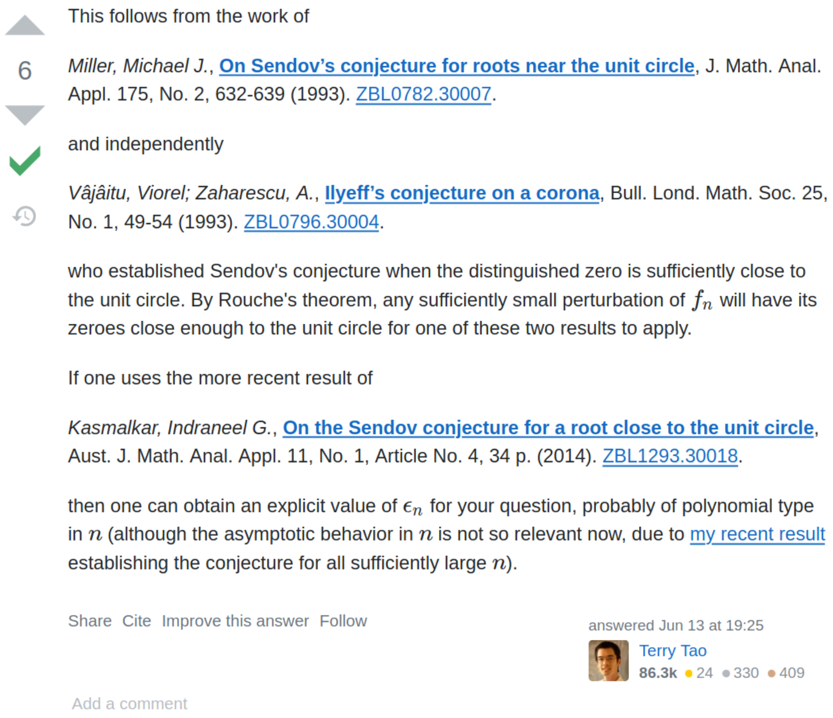
3 MathOverflow endpoint

Over four years ago, a fruitful collaboration between MathOverflow and zbMATH has led to the establishment of a new button labelled “Insert Citation” on the MathOverflow website[19]. The button appears when adding any question or answer and enables users to insert a properly formatted citation to a research article or book. The user can enter a few words of the title or names of some authors and will be presented with a short list of matching papers. If they click on one, a citation to the document will be inserted into the MathOverflow post. This citation includes a link to the respective zbMATH Open entry. The user-facing side of this process is described in [2], here we will focus more on the technical implementation.

To make this suggestion process possible, the well-known MathOverflow user Scott Morrison[20] added some client-side code which calls the MathOverflow API on

[19] https://mathoverflow.net

[20] https://mathoverflow.net/users/3



This follows from the work of

Miller, Michael J., **On Sendov’s conjecture for roots near the unit circle**, J. Math. Anal. Appl. 175, No. 2, 632-639 (1993). ZBL0782.30007.

and independently

Vâjâitu, Viorel; Zaharescu, A., **Ilyeff’s conjecture on a corona**, Bull. Lond. Math. Soc. 25, No. 1, 49-54 (1993). ZBL0796.30004.

who established Sendov's conjecture when the distinguished zero is sufficiently close to the unit circle. By Rouche's theorem, any sufficiently small perturbation of f_n will have its zeroes close enough to the unit circle for one of these two results to apply.

If one uses the more recent result of

Kasmalkar, Indraneel G., **On the Sendov conjecture for a root close to the unit circle**, Aust. J. Math. Anal. Appl. 11, No. 1, Article No. 4, 34 p. (2014). ZBL1293.30018.

then one can obtain an explicit value of ϵ_n for your question, probably of polynomial type in n (although the asymptotic behavior in n is not so relevant now, due to my recent result establishing the conjecture for all sufficiently large n).

Figure 8. The MathOverflow user Terry Tao citing some articles using the “Insert Citation” feature (post available at https://mathoverflow.net/a/395248).

the side of zbMATH Open and presents the results to the user in a readable format. The API is actually a part of the citation matcher described in more detail in Section 4. The text entered by the user is matched against an Elasticsearch[21] index, which contains all data from the following fields:

- document title;
- original title (in the case of non-English literature);
- author names;
- journal source;
- pagination;
- year of publication.

[21] https://elastic.co

Matching is then done using a standard TF/IDF algorithm:[22] A document is scored higher the more often a given term appears in it (TF, or term frequency). However, the more documents a given term appears in (IDF, or inverse document frequency), the lower its impact in boosting the score is. Thus less weight is given to common terms that appear in a lot of documents. The documents are ranked by the resulting score in descending order, and the top three are returned. The data is exchanged between the browser and the backend using a previously agreed JSON format. More details are presented in Section 4 below.

4 Citation matching

4.1 History

For services within the mathematical infrastructure community, it is often beneficial to be able to interconnect resources by adding links to external services. This includes links to article fulltexts (via DOIs or directly at open-access locations), or to arXiv preprints, but also to reviews at zbMATH Open. The ability to find such links easily is beneficial to publishers, providers of repositories and other services, and thus ultimately to mathematicians using such services. Six years ago zbMATH has therefore started to offer an automated link-finding service called the “zbMATH Citation Matcher”[23] (affectionately labelled “zbMATcH”). It consists of an HTML interface designed for manual use, as well as an API meant for automated access via script. A detailed documentation for the latter is available upon request.

4.2 Algorithm

Like the MathOverflow search described in Section 3, the Citation Matcher works by searching for the terms supplied by the user inside an Elasticsearch index. Here, however, the search is done in a more structured fashion, with dedicated fields for title, author, etc.

Both the HTML interface and the machine API accept input as an unstructured citation text as well as input split up into the respective fields. Thus one can search directly for a citation string like, e.g., “X. Chen, Rational curves on K3 surfaces, J. Algebraic Geom. 8 (1999), 245–278”, or manually enter each relevant part of the citation into the respective input field. In the latter case, matching is done directly, while in the former, the citation string has to be split up and tagged correctly first.

For splitting and tagging the citation string, we use the open source machine learning software GROBID.[24] It takes as input an unstructured string and returns

[22] https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[23] https://zbmath.org/citationmatching/

[24] https://github.com/kermitt2/grobid

an XML-encoded response that, according to its best guess, tags which part of the string is an author name, title, publication year, etc. For many commonly used citation formats this works quite well, though in more exotic cases it can give wrong answers. Moreover, the models that the software is shipped with have not been trained specifically on citations as used commonly within mathematical journals, so any formats or citation styles that are specific to the mathematical community can lead to less accurate results. It would be ideal to use a custom model that has been trained on citation data specific to mathematics. However, doing so would need a large corpus of manually tagged citation strings from mathematical publications, hence this approach has not been implemented yet due to lack of resources.

Once the unstructured data has been split up and tagged, a search request can be made to the Elasticsearch index. The index used is filled with the data from zbMATH Open and contains the most important fields for searching citation data. As in the MathOverflow case, a TD/IDF score is computed for each document, and results are ranked by this score in descending order. The topmost results are returned if their score exceeds a certain threshold (set to 5.0 by default, but adjustable in the API).

How high this threshold score should be is a non-trivial question, especially when using the results of the API to add links to zbMATH Open automatically, i.e., without human supervision: The lower the threshold score, the more frequently a result is returned, but the proportion of false matches will also go up. If on the other hand one sets it too high, the retrieved results will be more likely to be correct, however it will also be more often the case that a citation is not matched even though it is contained in zbMATH Open. In automated applications, a somewhat conservative threshold score of 8.0 is recommended to keep the number of errors of the first kind to an acceptably low level.

4.3 Future developments

Even though the zbMATcH algorithm has worked well for several years, it does have its share of problems and things left to be desired.

- Chief of these is that the TF/IDF score computed by Elasticsearch is only designed to rank the results of a single query amongst each other, not to compare results of different queries. Of course, this is exactly what we are doing if we postulate a global threshold score and only accept results when their score exceeds this threshold.
- Secondly, using the current algorithm it is not easy to add new features or criteria to the matching, or to measure the effect such features have if they are introduced.
- And finally, the current implementation is not able to take into account additional information besides the content of the citation string. For example, most citation

strings do not materialise out of thin air, but are instead contained in the references section of some publication. Now it is quite unlikely that the cited article has a publication year that lies in the future of the citing document (it can happen occasionally, for example due to preprints, reprints, or delays in publication, but in general it is quite safe to assume that cited articles have been published prior to the ones citing them). Hence, if one has the additional information of the publication year of the citing document, one can take that information into account when scoring candidate results of the matching process. Other ways of incorporation additional metainformation are also possible, e.g., coauthor networks or MSC classification (articles are more likely to cite within their own field, and we have data on which MSC classes are frequently cited from which others).

We are therefore implementing a new algorithm that is designed to overcome these weaknesses and provide better matching results while being at the same time more modular and hence easier to modify and evaluate. Figure 9 gives an overview of the structure of the new algorithm. As before, GROBID is used to structure text citations, however it is still possible to query using structured data directly.

What follows is a list of modules called *candidate generators*. Their task is to loosely select a set of documents, according to appropriate criteria, which have at least a non-negligible chance of matching the input citation. Their purpose is mainly to ensure efficiency, so that the matching score does not need to be computed for every single document contained in the database, but rather for this preselected set only. Several ways of generating candidates are conceivable, for example an Elasticsearch query as in the old algorithm, or a selection just by the numbers occurring in the citation string. The latter is helpful in particular for citation styles that do not contain a title. Moreover, collections of numbers, like volume, issue, page numbers, and publication year, tend to exhibit a high degree of uniqueness and hence of specificity. Using more than one candidate generator helps to ensure that no relevant target document is accidentally left out of the matching process.

Next comes the core of the new algorithm: The so-called *featurizers*, which for each candidate compute a numerical feature that encodes a certain degree of similarity to the input citation. There can be as many of these as needed. Each focuses on one specific property of the input/candidate pair. They can, for example, compare certain structured fields using appropriate methods (for example, allowing for LaTeX encoding in titles, or author names using initials only), or work with the raw citation text (e.g., by comparing substrings). If supplied, featurizers can also make use of metadata of the document containing the citation, for example by comparing publication years as explained above.

The output of this step is a list of numerical feature scores, which can then be fed into almost any kind of machine learning algorithm. Our example uses a support vector machine (SVM), which essentially computes a (weighted) linear combination

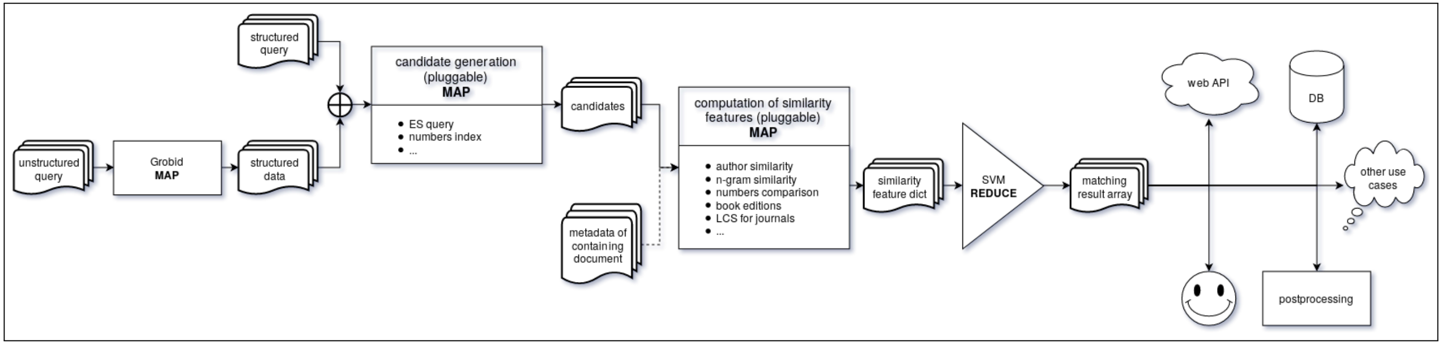


Figure 9. Schematic workflow of the new citation matching algorithm.

of the features. The weights have to be learned by training this algorithm, for which a set of so-called *gold data* is needed, i.e., citations where the correct matching document is known. For this, we use a set of references where a DOI is included (and the document referred to is part of the zbMATH Open corpus). By comparing the weights after the training is finished, it is even possible to determine to what extent each featurizer contributes to the final score.

As before, the final output of the algorithm is a list of scored candidates, ranked by score in descending order. The output fields of the new algorithm is a superset of the ones returned by the old, and likewise for the input. Hence existing tools can use the new one as a drop-in replacement without any changes. However, it is of course hoped that by making use of the new features the quality of the matching is greatly improved.

5 OAI-PMH API

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) protocol is widely used for metadata harvesting. With our OAI-PMH API,[25] one can harvest the entire zbMATH Open dataset or some specific subsets of it. In this section, we present an overview of the implemented endpoints and our custom extensions based on our previous publication [6].

5.1 Endpoints and response body

As required by the protocol, our API offers six endpoints:

- Endpoint 1 (“Identify”) helps aggregators and archives to discover the API fully unsupervised. Further it identifies the version of the OAI-PMH standard used;
- Endpoint 2 (“ListMetadataFormats”) lists the formats that we use to expose the data of zbMATH Open. We implemented two flavours, the standardized Dublin Core[26] metadata format (which is required by the standard) and a second format, that is closer to the internal data model of zbMATH Open. The content generated by zbMATH Open, such as reviews, classifications, software, or author disambiguation data are distributed under the CC-BY-SA 4.0 license.[27] This defines the license for the whole dataset, which also contains non-copyrighted bibliographic metadata and reference data derived from I4OSC (CC0). Note that the API does

[25] https://oai.zbmath.org

[26] https://dublincore.org

[27] https://creativecommons.org/licenses/by-sa/4.0

only provide a subset of the data in the zbMATH Open Web interface since in several cases third-party information, such as abstracts, cannot be made available under a suitable license through the API. In those cases we replaced the data with a placeholder string. We envision that for researchers dealing with different data providers, the Dublin Core format is more suitable. On the other hand, we expect that for people used to our website, our own format is more appealing;

- Endpoint 3 ("ListSets") lists the subsets of the zbMATH Open dataset, i.e., one set for each primary MSC label and one set for the articles originating from the 'Jahrbuch über die Fortschritte der Mathematik.'
- Endpoints 4 ("ListIdentifiers") and 5 ("ListRecords") list the current identifiers and records, respectively. This endpoint is intended to provide a dump of all public data contained in zbMATH Open;
- Endpoint 6 ("GetRecord") gets specific entries of zbMATH Open.

5.2 Extensions to the standard

While the endpoints defined in the OAI-PMH schema are useful for retrieving large fractions of the zbMATH Open dataset, the search capability for specific articles is limited. Therefore, we extended the OAI-PMH standard with custom endpoints without breaking the compatibility with the leading protocol. In particular, we have designed a simple query language that allows filtering based on the following properties: document type, year, document author, classification, keyword, document language, author variation, author reference, biographic reference, software, review type, review language, reviewer, serial publisher. All those fields can be combined with the boolean operators 'and', 'or' and 'not'. We chose the operators in a way that they are outside the alphabet for set names. By doing so no extra escaping or confusion between operators and sets is possible.

6 zbMATH Open REST API

As outlined in Section 5.2, we immediately realised that the OAI-PMH standard is not an optimal fit for all use case scenarios and does not optimally match the requirements of the five user groups outlined in Section 1. With the implemented extensions, we can retrieve more specific subsets of our dataset. However, we are still bound to the OAI-PHM metadata format and protocol. For example, the result format must be XML, which is hard to process for less experienced developers. Moreover, the search capabilities are very limited and write operations are not defined by the standard. In addition, defining metadata schema definitions in XML is connected with significant overhead and makes changes to the API more difficult. For example, correcting mistakes is not the purpose of that standard. Therefore, we plan to develop a custom

REST API tailored explicitly to the zbMATH Open dataset, providing different results formats, including JSON and XML. Eventually, we want to take our API development to the next abstraction level, making all information visible on the website machine readable. This would allow future user interfaces to run on top of the API without accessing internal data sources directly. This will facilitate the development of alternative frontends such as clients for mobile devices. By doing so, we follow the example of DataCite[28] and others that provide different APIs to access the bibliographic content. Eventually, different APIs that present the data in different formats for various purposes will contribute to the vision of interoperable research graphs [1].

While the OAI-PMH API is designed for harvesting data, not for updating data (note that also the current zbMATH Open website performs read-only access to the zbMATH Open database), we will in the future allow write operations via APIs. To ensure high data quality, we will require user authentication and double-check all incoming data before processing it further to ensure the reliability of zbMATH Open. This was also a crucial point in the development of the Scholix Link API and will be subject to discussion and interaction with the communities in order to find a good balance between high quality and high volume of data available at zbMATH Open.

7 Concluding remarks

The main purpose of this contribution is to provide a broad and complete scenario of the recent digital innovations in zbMATH Open.

Having made the data freely accessible has obviously offered a wide range of new ideas and resources in order to optimise the usability of our service. Some of these ideas have already been worked out, as discussed in this article, others will come soon.

We see two great challenges for the future: on the one hand to improve and solidify what we have already built in the recent past, on the other to frame our digital services in a universal scheme. The latter is undoubtedly the most difficult and exciting test for us. This scheme must contain in a functional and organic way all the various services discussed in this article in order to make zbMATH Open a solid tool for the community, avoiding the risk of offering disconnected and non-interoperating services.

[28] https://support.datacite.org/docs

References

[1] A. Aryani, M. Fenner, P. Manghi, A. Mannocci, and M. Stocker, Open science graphs must interoperate! In *ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, Lyon, France*, edited by L. Bellatreche, M. Bieliková, O. Boussaïd, B. Catania, J. Darmont, E. Demidova, F. Duchateau, M. Hall, T. Merčun, B. Novikov, C. Papatheodorou, T. Risse, O. Romero, L. Sautot, G. Talens, R. Wrembel, and M. Žumer, pp. 195–206, Communications in Computer and Information Science 1260, Springer Cham, 2020

[2] I. Beckenbach, zbMATH Open and community platforms. *This volume*, pp. 49–53, EMS Press, 2024

[3] K. Hulek and O. Teschke, The transition of zbMATH towards an open information platform for mathematics. *Eur. Math. Soc. Newsl.* **116** (2020), 44–47

[4] F. Müller, M. Schubotz, and O. Teschke, References to research literature in QA forums – a case study of zbMATH links from MathOverflow. *Eur. Math. Soc. Newsl.* **114** (2019), 50–52

[5] M. Petrera, D. Trautwein, I. Beckenbach, D. Ehsani, F. Müller, O. Teschke, B. Gipp, and M. Schubotz, zbMATH Open: API solutions and research challenges. In *DISCO 2021: Digital Infrastructures for Scholarly Content Objects 2021*, edited by W.-T. Balke, A. de Waard, Y. Fu, B. Hua, J. Schneider, N. Song, X. Wang, pp. 4–13, CEUR Workshop Proceedings 2976, CEUR-WS.org, 2021

[6] M. Schubotz and O. Teschke, zbMATH Open: Towards standardized machine interfaces to expose bibliographic metadata. *Eur. Math. Soc. Newsl.* **119** (2021), 50–53