

Zuse Institute Berlin

ZIB

Takustr. 7
14195 Berlin
Germany

HAGEN CHRAPARY, WOLFGANG DALITZ, WINFRIED NEUN AND
WOLFRAM SPERBER

**Design, concepts, and state of the art
of the swMATH service**

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Design, concepts, and state of the art of the swMATH service

Hagen Chrapary, Wolfgang Dalitz, Winfried Neun and Wolfram Sperber

Mathematics Subject Classification (2010). Primary 68T30; Secondary 68T35.

Keywords. software information, software citation, mathematical research data.

Abstract. In this paper, the concepts and design for an efficient information service for mathematical software and further mathematical research data are presented. The publication-based approach and the Web-based approach are the main building blocks of the service and will be discussed. Heuristic methods are used for identification, extraction, and ranking of information about software and other mathematical research data. The methods provide not only information about the research data but also link software and mathematical research data to the scientific context.

1. Introduction

Research data is a buzzword in the current discussion of scientific infrastructure. It is a very general term and frame for research data of all kinds, data from experiments, simulations, software, etc. Some types, especially research objects, models, experiments, simulations, software, researchers, conferences, etc. are present in many scientific fields but the data, their meaning and their descriptions vary considerably. So, mathematical objects or models are described by mathematicians in a different way than by engineers. Here we confine ourselves to special types of mathematical research data.

The origin of our activities in mathematical research data is mathematical software. Mathematical software plays an increasing role for the application of mathematical methods and for mathematical research which is used in various ways, e.g., for implementation and verification of algorithms, proving of theorems, simulations, for dynamic mathematical services as Online Encyclopedia of Integer Sequences (OEIS) [1], etc. There are further unique characteristics of (mathematical) software:

- Software is written in a formal language, not in a natural language. Therefore, software is outside of the classical publication process. Information about software, e.g., manuals or documentations are mostly given in accompanying documents and not in the software itself.
- Software has a life cycle. Often, software has versions, releases, bugfixes, etc.
- Software depends on its environment (hardware, operating system, special programming languages, other software).
- License conditions for software differ.
- Software is widely distributed.
- Evaluation of software is sometimes complicated.

As a result, information services about mathematical software are isolated, not standardized, widely distributed, and provide less information for the user. In the following, we describe the concepts

and the state of the art of the swMATH service [2]. Most of the existing information services for mathematical software and research data are maintained manually. This is expensive and has led to a lot of non-standardized information services for single mathematical subjects. It is the intention of the swMATH project to develop a concept for a comprehensive and homogeneous information service on all mathematical software.

Therefore we have tried to develop machine-based concepts for evaluating and combining the existing information about a software given in publications and on Web sites, repositories, portals and directories.

2. The first design principle: The publication-based approach

Publications cite software for two reasons: they describe the software and their mathematical background or they use a software for solving a mathematical problem. Both classes deliver different information about publications. swMATH analyzes the data of the database zbMATH [3] and ioport [4] instead of the fulltexts of publications. This means only peer-reviewed publications (journal articles or books) are considered and described by a standardized metadata scheme. For the identification of mathematical software especially the title, the review or abstract, and the references are analyzed. Of course, zbMATH data do not completely cover the spectrum of mathematical software. Often mathematical software has a background in specific applications and the publications citing the software are outside of the scope of zbMATH. Secondly, there is a time-lag between a publication and its appearance in zbMATH.

2.1. Identification of software references and software citations

The identification of software references in citations is non-trivial because references to software are often incomplete. The current practice of software citations has some deficits:

- *Indirect citation*
Instead of the software, a publication about the software is cited.
- *Missed typing*
The cited resource is not declared as software. Moreover, the borderline between software, services, languages and environments is fluent and a distinction can be difficult.
- *lack of information*
The citation contains not more than the name of the software, data about the version are missing.
- *Proprietary citation standards*
Software companies, repositories, and publishers give different recommendations for citing of software.

We have analyzed the citation practice in mathematics for the optimization software SCIP. 180 publications were detected citing SCIP: 179 in the database zbMATH [3] and 1 in the database ioport [4]. The references were found in different metadata fields. All publications refer to SCIP in the field ‘Software’. SCIP is also contained in the field ‘References’ in 84 publications, 15 times in ‘Review/Abstract/Summary’, 1 times in ‘Title’, 1 times in ‘Keywords’.

But only the field ‘References’ contains more information than the name of the software. The citations for SCIP refer 64 times to a certain article (which has a DOI as persistent identifier) and 22

Mention Type	Count	Percentage
Cite to publication	84	46,67 %
Cite to user manual	2	1.11 %
Cite to website	19	10.56 %
Cite to other sources	3	1.67 %

TABLE 1. Cited resources of SCIP

times to the dissertation of the author of the publication. The SCIP homepage, including its URL, is cited 19 times, the other references go to a SCIP workshop, reports, other packages and user manuals.

Table 1 illustrates the citation praxis for software in detail (some publications cite more than one source, e.g., a publication plus the homepage)

Some publications contain more than one of these sources. Only two references provide also data about the versions. The current recommendation from SCIP, a referencing to a ZIB report of the current version 3.2 was not found inside the zBMATH entries.

Generally, the situation of software citations in mathematics is very ambiguous. There is some software, e.g., SageMath, where the citations in publications often contain also data about the versions used. Software providers, repositories, publishing houses, etc. have defined different policies for citing. The increasing importance of scientific software reflected in an increasing number of citations requires a standardization of software citations.

Currently, the Software Citation Working Group of the FORCE11 initiative has published a concept for software citations, the ‘Software Citation Principles’ [5]. The references to software should be typed as software, possibly extended by a finer classification (source code, package, etc.), and contain also information about the versions or alternatively the update date and download date. But, software code itself is not convenient as target of a citation because the software code is typically under dynamic development. What is also needed are persistent identifiers (for more information on persistent identifiers for software see the discussion below) and persistent information about a software, e.g., in the form of metadata. This is a non-trivial task because a widely accepted metadata standard for software does not exist until now. The missing metadata standard and DOIs are strong reasons that often publications about a software are referenced instead of software itself. Further difficulties are the maintenance and actuality of software information. A manual maintaining and updating of software information is too expensive and not realistic for the whole area of mathematical software. But, software information services, e.g., specialized portals or repositories, are very well able to provide comprehensive and persistent information about software (especially about the content of a software) even if the software code is removed. This requires the development of new concepts and methods for an adequate infrastructure. This process is under work. The FORCE11 initiative plans in a next step to develop concepts and implementation rules for software citations. This could be, e.g., a BIBLATEX implementation for what we prepare a proposal. The swMATH activities which are described in the following in detail are an example and a possible way to build up an efficient infrastructure for mathematical software information.

Number of citations	Number of swMATH entries
1	4588
2 - 5	4606
6 - 10	1668
10 - 10	1194
20 - 50	994
50 - 100	350
100 - 200	217
200 - 500	113
500 - 1000	31
1000 ++	11

TABLE 2. The distribution of software citations in swMATH

Despite all the difficulties, the analysis of software citations in zbMATH turned out to be a successful approach for the identification of mathematical software in publications. In a first step, software citations in the zbMATH data must be detected. Therefore, some heuristic methods have been developed. These methods are text-based: phrases which contain characteristic terms as 'package', 'software', 'module', etc. are extracted. Also the increasing list of the swMATH entries is reused to identify software citations in the zbMATH data.

As a result we get a list of mathematical software (and related mathematical research data because the method can not exactly distinguish software and other types, e.g., languages and environments, benchmarks, or services) and a second related list of publications citing a software. The publications provide especially important information about the content of a software and are used for swMATH in various way, see below.


We have observed that the number of publications is much higher than the number of software packages, swMATH lists currently (November 2016) more than 15,500 software packages (and related mathematical research data) which are cited in 128,000 publications. The number of all software citations in zbMATH is greater than 213,000. The number of software citations varies strongly, some universal and popular software as 'Mathematica' or 'Maple' have thousands of citations, specialized and/or new software packages have only a few citations, for the distribution of citations to software see Table 2. A large number of citations allows a ranking of the information about a software.

2.2. Computer Algebra Systems (CASs) and Symb. Computation software in swMATH

At a Computer Algebra conference it is highly interesting how swMATH deals with the CASs or Symbolic Computation software and how the results look like.

swMATH provides information for CASs (ca. 200 entries, depending if your search for 'computer algebra system' or 'CAS', see Figure 1) and Symbolic Computation software (ca. 100 entries). Since Symbolic Computation is the more general term, it is astonishing to see more CAS entries. It seems to be caused by the more modern term CAS, but also it can be observed that symbolic computation software is not used in a reference for a CAS.

[About & Contact](#) [Feedback](#) [Contribute](#) [Help](#) [zbMATH](#)


 [Advanced search](#) [Browse](#)

Results 1 to 20 of **254** Sort by:

Mathematica Referenced in 3782 articles [[sw00554](#)]
 deployed individual or enterprise solutions. **Computer algebra** system...

Maple Referenced in 3743 articles [[sw00545](#)]
 math, graphics, images, sound, and diagrams. **Computer algebra** system...

Magma Referenced in 1554 articles [[sw00540](#)]
Computer algebra system (CAS). Magma is a large, well-supported software package designed for **computations** ... structures such as groups, rings, fields, modules, **algebras**, schemes, curves, graphs, designs, codes and many ... **computational** research in those areas of mathematics which are **algebraic** in nature. The overview provides ... language. Magma is distributed by the **Computational Algebra** Group at the University of Sydney...

GAP Referenced in 1579 articles [[sw00320](#)]
 system for **computational** discrete **algebra**, with particular emphasis on **Computational** Group Theory. GAP provides ... library of thousands of functions implementing **algebraic** algorithms written in the GAP language as well ... groups and their representations, rings, vector spaces, **algebras**, combinatorial structures, and more. The system, including ... extend it for your special use. **Computer algebra** system...

SINGULAR Referenced in 879 articles [[sw00866](#)]
 SINGULAR is a **Computer Algebra** system (CAS) for polynomial **computations** in commutative **algebra**, **algebraic** geometry ... singularity theory. SINGULAR's main **computational** objects are ideals and modules over a large variety ... field (e.g., finite fields, the rationals, floats, **algebraic** extensions, transcendental extensions), or localizations thereof ... SINGULAR features fast and general implementations for **computing** Groebner and standard bases, including e.g. Buchberger...

Macaulay2 Referenced in 910 articles [[sw00537](#)]
 supporting research in **algebraic** geometry and commutative **algebra**, whose creation has been funded ... since 1992. Macaulay2 includes core algorithms for **computing** Gröbner bases and graded or multi-graded ... integral closure of rings, and more. **Computer algebra** system...

REDUCE Referenced in 622 articles [[sw00789](#)]
 interactive system for general **algebraic computations** of interest to mathematicians, scientists and engineers. **Computer algebra**...

SageMath Referenced in 686 articles [[sw00825](#)]
 research and teaching in **algebra**, geometry, number theory, cryptography, numerical **computation**, and related areas. Both ... Maple, Mathematica, Magma, and MATLAB. **Computer algebra** system...

FIGURE 1. List of CAS Software

Another observation is that older software which is integrated as module in a bigger and comprehensive software system is later on still referenced as an independent system. Some systems simply disappeared for commercial reasons. This leads to extra entries which may be misleading.

Typical examples are MuPAD or Derive. There should be some background knowledge saying that these systems are no longer available separately.

There are also some other problems. The Lambert W function was one of the topics of the invited talks at ACA 2016. If you look for the Lambert W function in swMATH (by searching for the string 'Lambert'), the first hit (Figure 2) provides a lot of very unstructured information, not what you expect to get about the research work of Lambert.

The screenshot shows the swMATH interface for the LambertW package. At the top, there are navigation links: About & Contact, Feedback, Contribute, Help, and zbMATH. Below these is the swMATH logo and a search bar with options for Search, Advanced search, and Browse. The search results for 'LambertW' are displayed, including a description of the package, its URL, manual, authors, license, current version, and dependencies. A word cloud of keywords is shown, with 'Lambert W' and 'family of skewed distributions' being prominent. Below the word cloud, there are references in zbMATH, a search bar for articles, and an MSC classification section.

LambertW

LambertW: Analyze and Gaussianize skewed, heavy-tailed data. The Lambert W framework is a new generalized way to analyze skewed, heavy-tailed data. Lambert W random variables (RV) are based on an input/output framework where the input is a RV X with distribution $F(x)$, and the output $Y = \text{func}(X)$ has similar properties as X (but slightly skewed or heavy-tailed). Then this transformed RV Y has a Lambert $W \times F$ distribution - for details see References. This package contains functions to perform a Lambert W analysis of skewed and heavy-tailed data: data can be simulated, parameters can be estimated from real world data, quantiles can be computed, and results plotted/printed in a 'nice' way. Probably the most important function is 'Gaussianize', which works the same way as the R function 'scale' but actually makes your data Gaussian. An optional modular toolkit implementation allows users to define their own Lambert $W \times$ 'my favorite distribution' and use it for their analysis. (Source: <http://cran.r-project.org/web/packages>)

URL: cran.r-project.org/web/packages/LambertW/
 Manual: cran.r-project.org/web/packages/LambertW/manual/
 Authors: Georg M. Goerg
 Licence: GPL (≥ 2)
 Current version: 0.2.9.9
 Dependencies: R; moments, gsl, MASS, nortest, maxLik

Add information on this software.

Related software:
 R
 cran

Keywords for this software

exact likelihood ratio test
 family of skewed distributions
 latent variables
 Lambert W
 transformation of random variables
 likelihood based inference
 logarithmic Lambert W random
 stylized factor of asset returns
 skewness value at risk RCH

References in zbMATH (referenced in 2 articles, 1 standard article)

Showing results 1 to 2 of 2. Sorted by year (citations) 20

1. Witkovský, Viktor; Wimmer, Gejza; Duby, Tomy: Logarithmic Lambert $W \times$ random variables for the family of chi-squared distributions and their applications (2015)
2. Goerg, Georg M.: Lambert W random variables -- a new family of generalized skewed distributions with applications to risk estimation (2011)

Article statistics & filter:

Search for articles


MSC classification / top

- Top MSC classes
 - 00 Probability theory and...
 - 02 Statistics
 - 05 Numerical analysis

FIGURE 2. The swMATH page of LambertW

More relevant information is provided on the swMATH page on LambertWDDE (Figure 3). Here some work must be done in the future.

[About & Contact](#)
[Feedback](#)
[Contribute](#)
[Help](#)
[zbMATH](#)



LambertWDEE

Analysis and control of time delay systems using the LambertWDEE Toolbox This chapter provides an overview of the Lambert W function approach. The approach has been developed for analysis and control of linear time-invariant time delay systems with a single known delay. A solution in the time-domain is given in terms of an infinite series, with the important characteristic that truncating the series provides a dominant solution in terms of the rightmost eigenvalues. A solution via the Lambert W function approach is first presented for systems of order one, then extended to higher order systems using the matrix Lambert W function. Free and forced solutions are used to investigate key properties of time-delay systems, such as stability, controllability and observability. Through eigenvalue assignment, feedback controllers and state-observers are designed. All of these can be achieved using the Lambert W function-based framework. The use of the MATLAB-based open source software in the LambertWDEE Toolbox is also introduced using numerical examples.

URL: rd.springer.com/chapte...
Authors: Yi, Sun; Duan, Shiming; Nelson, Patrick W.; Uiso, A. Gallip

[Add information on this software.](#)

Related software:
 DDE-BIFTOOL
 Matlab
 TRACE-DDE

Keywords for this software

observability

Lambert W function

linear time-invariant time delay systems

stability

controllability eigenvalue assignment

References in zbMATH (referenced in 1 article)

Showing result 1 of 1. Sorted by year (citations) 20

1. Yi, Sun; Duan, Shiming; Nelson, Patrick W.; Uiso, A. Gallip: Analysis and control of time delay systems using the LambertWDEE Toolbox (2014)

Article statistics & filter:

Search for articles

MSC classification / top
 Top MSC classes
 93 Systems theory; control

FIGURE 3. The swMATH page of LambertWDEE

2.3. swMATH Features: Citation profile

A first informative statistical information of a software is its citation profile. Therefore we create the citation graph presenting the annual numbers of publications citing a software. An example of a citation profile is presented in Figure 4.

The graph is meaningful to characterize the development stage and reflects the distribution and the acceptance of the software, especially it provides information on

- *Dissemination and acceptance*

A large number of publications indicates that the software has been widely used but a small number of publications can have different reasons: the software is specialized, or written in

Chart: cumulative / absolute



FIGURE 4. The citation graph of the software 'SageMath'

a special language, or is not widely distributed, or requires special hard- or software, or is high-priced, etc.

- *Life cycle*

The life cycle of software covers the different phases: planning, analysis, design, implementation, maintenance up to the substitution by an successor or an other software. The number of publications per year reflects indirectly the phases. Publications describing a software are typical for the planning, analysis, and (re-)design and (re-)implementation phases, publications using a software for the maintenance phase. A strongly decreasing number of publications is a signal for the substitution of the software.

- *Quality*

A high number of usage cases in publications citing a software can be also considered as an indicator for the quality of the software.

2.4. swMATH Features: Facets for the content information and classification

For publications there exists a metadata standard defining the most important information about it. For software such a standard is missing because of the unique features of software mentioned in the introduction. Comprehensive information about software contains

- metadata about
 - content
 - programming languages and environment
 - technical parameters
 - versioning
 - licenses and usability conditions
 - use cases and examples
 - developers (persons, institutions, companies)
 - contacts
- and links to

- documentations and manuals
- access (e.g., software code if free available)
- related information (publications, algorithms, data, repositories)

As said before, software can play a different role in publications. It can be the main content of a publication or used as a tool for solving a problem. This distinction is important for the analysis of the information about software. Therefore we distinguish between two classes of publications. The first class covers all publications which are focused on software: these publications we call ‘*standard publications*’. All other publications are so-called ‘*user publications*’. At least, a standard publication could be identified for more than 95 % of the swMATH entries.

Two remarks:

1. A major part of these standard publications is outside the scope of zbMATH, e.g., reports or articles in non-mathematical journals.
2. It is possible that there exist more than one standard publication, e.g., a new extended version of a software can be presented in a publication, but the mass of publications are user publications.

The information of standard and user publications is complementary: Standard publications provide a lot of information about the content of the software and its mathematical background, e.g., the mathematical model and algorithms. User publications name possible applications areas and present results which have been achieved by using a software. The dichotomic classification into standard and user publications can be done semi-automatically. Typically, standard publications contain the name of the software and some characteristic terms in the title, the publishing date (at least of the first standard publication) is before user publications, and also the classification codes reflect the type of publication. The appearance of the software name in the title is a strong indicator for a standard publication. Only four entries in the hit list for searching ‘computer algebra’ has not the name of the software in the title of the standard publication.

The zbMATH data are analyzed depending on the type of publication. The publication-based approach suggests to use the same metadata for the content analysis as in zbMATH, namely a short description (review, abstract, or summary), keywords, and classification.

- *Content description*

In a first approach we undertake the review or abstract or summary from standard publications (if there are more than one standard publication the most recent is used).

- *Keywords*

The keywords of all standard and user publications are aggregated, ranked and visualized in the keyword cloud. Ranking is done by the frequencies of the keywords. We could imagine a splitting in mathematical and user keywords. The problem with this is that the zbMATH keywords do not distinct between mathematical keywords and such ones for (non-mathematical) application areas.

- *Classification: MSC profile*

All zbMATH data are classified by the Mathematical Subject Classification 2010 (MSC2010) [8]. The MSC is focused on the classification of mathematical publications not for mathematical software. As consequence, mathematical software is concentrated to some MSC classes, especially the subclasses XX-04 ‘Explicit machine computation and programs (not the theory

of computation or programming)’ of the top classes of the MSC and 65-xx ‘Numerical analysis’, 68-xx ‘Computer science’, and 90Cxx ‘Operations research, mathematical programming’. Some subclasses, e.g., 68W30 ‘Symbolic computation and algebraic computation’, is a general class covering hundreds of software packages. But not only the granularity of the MSC is a problem, also the granularity of mathematical software differs. Some general software, e.g., ‘Mathematica’, ‘Maple’, ‘Mathlab’, ‘SageMATH’ cover a broad spectrum of mathematical subjects, other software has been developed for the solution of a special problem.

Moreover, the most classifications are not unique and are assigned to more than one MSC class. Nevertheless, the MSC codes of the publications are worthwhile information about a software. Therefore, the swMATH page of a software delivers a ranked list of the MSC codes of the publications citing a software. Basing on the frequencies of the MSC codes, a detailed MSC profile for each software is provided.

The MSC codes are also evaluated for the similarity analysis, see the discussion below. An analysis of the granularity of software based on the MSC codes and the number of citations appears to be possible but has not been done until now.

The most frequent MSC codes of the publications citing a software give an impression about the mathematical focus of the software and its applications. A large number of MSC codes with mathematical focus suggests that the software can be used for solving different mathematical problems (‘universal’ software), a large number of MSC codes in application areas implicates a broad dissemination and acceptance of the software.

2.5. Similar (related) software

Similar software is an important information for the user, e.g., to get information about alternative computation tools. To determine similar software, publications are used as weighting factor, see Figure 5. If different software packages are often cited in common, they are classified as related software. More weighting factors for the similarity metric are possible, especially the MSC classification and the keyword cloud. Experiments which compare also the MSC codes have not significantly improved the results but have led to long computing times.

Related software:

Magma
 GAP
 SINGULAR
 Sage-Combinat
 OEIS
 Macaulay2
 Mathematica
 PARI/GP
 Maple
 MuPAD
 Show more...

FIGURE 5. Related software for SageMATH

Remark: We use the weaker term 'related' instead of 'similar' which covers also broader-narrower relations between software, e.g., special packages or modules of a software.

2.6. Software developers

Software developers play another role than the authors of publications. Large software developer teams are often 'anonymous', summarized as 'developer team', and are under permanent change. Publications have a higher rating and reputation than software development. This is one of the reasons that software is often accompanied by standard publications. The assumption seems to be legitimate that the authors of the standard publications are closely related to the software development team. If no further information about the developers of a software is available, the authors of the standard publications are treated as possible contact persons and are searchable in the software developer index.

3. The second design principle: The Web-based approach

The publication-based approach is the first step in the swMATH workflow and provides a lot of information about mathematical software, especially about the content. But, details, e.g., versions, technical data or installations guides are missing. This kind of information can be found on the websites of a software, on repositories, or on portals which provide information about and access to software for a special subject. Therefore we have developed some concepts for capturing further information about software which are presented in the next chapter. In principle we are faced with the same tasks as in the publication-based approach: (1) identification of the relevant information and (2) extraction and processing of information of a software. But instead of publications we have to do it for the information in the Web. In an additional third step, the information from the Web is combined with publications.

- Websites identification and linking

The identification of websites of a software is done by a websearch. The information from the publication-based approach is an excellent starting point of the Websearch. A text search using the name of the software in combination with characteristic terms such as 'software' or 'package' is a simple but almost certain method to find the websites of a software. swMATH found for nearly $\frac{2}{3}$ of the swMATH entries— in absolute numbers 10.000 Web sites—URLs of the entries documented in swMATH. The URLs of the websites are self a first important information about software. Additionally, also some known repositories on mathematical software, e.g., the CRAN archive [12] for statistical software, and portals, e.g., the webpages on mathematical software of SIGSAM [9], the Computer Algebra Fachgruppe [10], or Wikipedia [11], which contain detailed information on software were identified and linked.

- Analysis of websites of software

There are two main difficulties:

- The dynamic character of websites

Most websites change dynamically and provide often only information about the most recent version of the software and not for earlier versions. Instead of the websites alone we can check also webarchives, especially the Wayback Machine [13] for information. The provider of the Wayback Machine, the Internet Archive, is a non-profit activity and

is being interested in cooperation with the community. Its main activity is the periodic archiving of websites. Therefore selected lists of websites are scanned and stored periodically. The URLs of the websites contained in swMATH are used as a seed list for the Internet Archive. All swMATH entries which have been scanned by the Wayback Machine are linked to it. The linked pages show the date of the scans, the absolute number of scans and provide links to the scanned websites. In other words, the Wayback Machine provides a lot information about the time changing content of the scanned URL, for software especially about the different versions.

- The heterogeneous content and structure of websites
The information on the websites is manifold. It contains tutorials and documentations, possibly the source code, information about necessary hard- and software, software developers and contact information, licenses and usability conditions, etc. The websites have different content and structure which complicate the automatic analysis. A first automatic method for analysis and classification of websites of a software has been developed by our cooperation partner Helge Holzmann from L3S, see [14]. Therefore, the websites of the different software versions found in the Wayback Machine are stored in microarchives. The archives contain the original websites and a secondary standardized structure. For the latter, the formats of the objects of the websites are analyzed.
- Linking to publications
Research results presented in publications are achieved with a special version of the software. Reproducing and verification of the research results requires information about the versions of the software and data used. Therefore, a linking from the publication to the used version of the software is essential. For the assignment of a software version to a publication, a heuristic method has been developed, see [14]: if a software version from the publication year the publication exists in the Internet Archive, the publication is linked to this version.

Figure 6 shows both links on the ‘Singular’ swMATH page to Wayback Machine and a microarchive.

3.1. A further problem for software information: Persistent identifiers

The SCPs [5] point to the relevance of persistent identifiers, e.g., DOIs, of the objects. But persistent identifiers should not refer to the software code which is changing or can be moved. Instead, persistent identifiers should be introduced for stable information about software. Portals and repositories are potential candidates to provide persistent information about software. The persistent identifiers should refer to both the software version used and a family of software which should be defined as the set of all versions of a software under the same label. As said before, the publication-based approach is content-centered and provides mainly non-technical information which is valid for the whole family of software under the same name. So, swMATH has introduced persistent identifiers for software families. In principle, these identifiers could be extended and completed by a supplementary suffix for the version (the version can be declared explicitly or described by the upload and download date). Software repositories which are equipped with version managing and control systems could provide also persistent identifiers for single versions, which can be used for information about the versions in swMATH in the future.

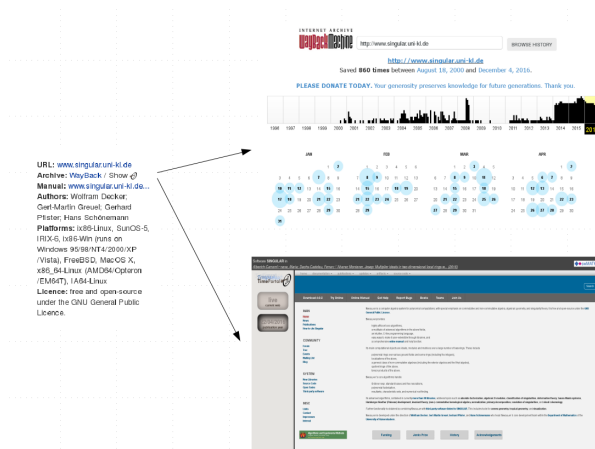


FIGURE 6. Linking of swMATH with Web Archive and Tempas

3.2. SWID Tags

SWID Tags [6] are another well-established system of metadata for the identification of software.

They have been developed earlier by ISO/IEC JTC 1, SC 7. These tags are mostly used in commercial environments for Software asset management [7]. The usage of these tags is defined, supported by tools and maintained by NIST and used by big players in the software market.

The main purpose of SWID tags is not the finding of software based on references from outside, like from publications and their keywords, but to make management of software easier and more secure e.g. by checking licenses, versions etc. These tags support big software firms and the administration of big installations.

The model for SWID tags usage is more likely a user (or administrator) in a big software pool rather than an individual researcher looking for (new) software solutions for mathematical problems.

4. The scope of swMATH and mathematical research data

swMATH is a portal not a repository. Software repositories like CRAN are software archives based on a common language and environment cover special subjects and provide tools for the search and the access to the software. Portals confine themselves to meta-information and linking. swMATH was planned and designed as a general portal and search engine for mathematical software.

Today, swMATH is an information service with the focus on mathematical software but it contains also other mathematical research data, especially languages and environments, repositories, portals, journals specialized to mathematical software, services and databases, benchmarks and test data collections, etc. The integration of other types of mathematical research data besides publications makes sense for different reasons:

- A classification scheme for mathematical software and research data is missing. Such services as OEIS are both data and integrated software. Such source should be inside swMATH.

- Sometimes the references in zbMATH data refer to software and other research data in the same context but the comment citation practice does not support an automatic filtering of different types of research data.
- Broadly accepted information services for special types of mathematical research data are missing.
- Other types of mathematical research data become more and more important, e.g., mathematical models which have been part of publications until now.
- Mathematical research data are strongly linked and should be presented in their context.

Different types of mathematical research data should be assigned to corresponding classes allowing specific views. But up to now, no classification scheme for mathematical research data exists. The MSC is primarily a scheme for classifying the mathematical subjects of a publication and contains only the special classes XX-04 for software. For data, the classification code XX-07 can be used which exists for some subject top classes of the MSC. For the futures, there are two ways: an extension of the MSC by other types or alternatively the development of an separate type scheme for mathematical research data. Machine learning methods could be used for the type classification of mathematical research data.

5. Conclusions

The permanent growth of usage shows the needs and the benefits of swMATH. The concepts and design developed by swMATH for a lean and machine-based maintenance of the service have been presented in the paper. Therefore, swMATH analyzes context information given by the mathematical literature to identify and extract information about mathematical software. This information can be enriched by special webresources, especially webarchives. The swMATH project is a first important step for the reproducing and evaluation of computing research results achieved by using mathematical software. Therefore it is not sufficient to develop information services for mathematical software only, information services for all types of mathematical research data must be developed and the information about the resources must be presented in their context.

References

- [1] The On-Line Encyclopedia of Integer Sequences, <https://oeis.org/>
- [2] swMATH, <http://www.swmath.org>
- [3] zbMATH, <http://www.zbmath.org>
- [4] io-port, <https://www.zentralblatt-math.org/ioport/>
- [5] Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group.(2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86
- [6] SWID (Software identification tags) defined in ISO/IEC 19770-2, http://www.iso.org/iso/catalogue_detail?csnumber=65666
- [7] Asset Description Metadata Schema, <http://joinup.ec.europa.eu/asset/adms/description>
- [8] Mathematical Subject Classification 2010 <http://www.msc2010.org>

- [9] The ACM Special Interest Group on Symbolic and Algebraic Manipulation, Computer Algebra Software, <http://www.sigsam.org/Resources/Software.html>
- [10] Fachgruppe Computeralgebra, Computeralgebrasysteme, <http://www.fachgruppe-computeralgebra.de/systeme/>
- [11] Wikipedia, List of computer algebra systems, https://en.wikipedia.org/wiki/List_of_computer_algebra_systems
- [12] The Comprehensive R Archive Network, <https://cran.r-project.org/>
- [13] Wayback Machine of the Web Archive, <http://web.archive.org>
- [14] Helge Holzmann, Wolfram Sperber, Mila Runnwerth: Archiving Software Surrogates on the Web for Future Reference, in Research and Advanced Technology for Digital Libraries; LNCS 9819, TPD 2016: 215-226

Hagen Chrapary
FIZ Karlsruhe
Dept. Mathematics and Computer Science
Franklinstr. 11
D-10587 Berlin
and
Zuse Institute Berlin (ZIB)
Takustr. 7
D-14195 Berlin
e-mail: hagen@zbmath.org

Wolfgang Dalitz
Zuse Institute Berlin (ZIB)
Takustr. 7
D-14195 Berlin
e-mail: dalitz@zib.de

Winfried Neun
Zuse Institute Berlin (ZIB)
Takustr. 7
D-14195 Berlin
e-mail: neun@zib.de

Wolfram Sperber
FIZ Karlsruhe
Dept. Mathematics and Computer Science
Franklinstr. 11
D-10587 Berlin
e-mail: wolfram@zbmath.org