Some steps to improve software information

Albert Heinle
Department of Computer science
University of Waterloo
Waterloo, Canada, ON N2L 3G1
aheinle@uwaterloo.ca

Wolfram Koepf
Department of Mathematics
University Kassel
Heinrich-Plett-Str. 40, Kassel, Germany, D-34132 Kassel
koepf@mathematik.uni-kassel.de

Wolfram Sperber
Department of Mathematics and Computer Science
FIZ Karlsruhe
Franklinstr. 11, Berlin, Germany, D-10587
wolfram@zbmath.org

2016-12-29

Abstract

Mathematical software plays an increasing role in mathematics and key technologies. But to find and get information about software is expensive. Missing metadata standards, e.g., the citation of software, are one of the reasons. In the following some recent developmentsare described how the community can contribute to a better information system for mathematical software.

1 Introduction

Computer Algebra Systems (CASs) and their use in research are a critical part of modern mathematics and moreover for a broad class of applications. More generally, scientific software has established itself as an autonomous kind of scientific research. But the existing scientific infrastructure is focussed on articles and books and does not support the information about software optimally. Also, methods for the evaluation and quality control of scientific software, in particular computer algebra software, must be developed and the development of scientific software should be given the same credit and reputation as it is given to other research results.

Citations play an essential role for identifying resources, as they help to track and weight the development and the realization of ideas and theories. Furthermore, the evaluation of research results, improvement of visibility of the cited resources and proper credits to authors are the origin for developing information services. The first section describes recommendations for a better citation practice for software.

Scientific software development and software information are widely distributed which complicates the integration of scientific software into existing infrastructures in an adequate way. In section 2 we give a brief overview about existing information resources, their role, and some problems in scientific software information services in mathematics. At the top, portal and catalogues are a first contact point to find software and information about it. There exist a lot of portals, also for symbolic computation, but manual maintenance and updating the information is expensive. The swMATH service—for a brief description of this system see section 4—is an attempt to create a comprehensive information service on mathematical software in a (nearly) automatic way. Therefore the close relationships between software and publications are essentially used. But the context of mathematical software is broader, it covers also algorithms, data, languages, people, communities, institutions, etc. The idea of a social network for the symbolic computation community is introduced in section 5. It is based on semantic web technologies allowing to maintain the information in distributed resources. It is one of the aims of this paper to sensitize the symbolic computation community to the questions and problems of a suitable scientific infrastructure for this mathematical subject. We hope that it will lead to an intensive discussion on these addressed problems within the community.

2 Software Citations

Publications—as the classical products of scientific research—use citations for embedding the content of a publication into its scientific context. Citations of publications are—at least in principle—standardized. But this is not at all true for software citations up until now. This results from the character of software. Software is dynamic and has a life cycle, has often different versions, releases, or bug fixes, etc., which prompt questions of archiving, reproducibility, and sustainability. Software is written in a formal language and is accompanied by documentation, manuals, metadata files, etc. The line between software and other types of research, especially algorithms, languages, environments and services is fuzzy. Software is dependent on hardware, the operating system and other software; licenses and usability conditions for software are varying. A metadata standard for software is missing. Mathematical software is closely linked to models and mathematical objects, theories and algorithms.

With respect to the use of computer algebra systems and packages within these systems the missing citation standard leads to the absurd situation that in most cases citation is just "turned off", hence the author of the software is not cited at all. Users of such systems who get results that they use in their papers often are not aware of whom they should cite and how they should do this properly.

Recently, the increasing importance of software expresses itself in more software citations in scientific publications. But these citations often provide not enough information about the software used. Fig. 1 gives an example for the citation of the "Singular" software in a publication.

Böhm, Janko; Decker, Wolfram; Keicher, Simon; Ren, Yue

Current challenges in developing open source computer algebra systems. (English) Zbl 06585009

Kotsireas, Ilias S. (ed.) et al., Mathematical aspects of computer and information sciences. 6th international conference, MACIS 2015, Berlin, Germany, November 11–13, 2015. Revised selected papers. Cham: Springer (ISBN 978-3-319-32858-4/pbk; 978-3-319-32859-1/ebook). Lecture Notes in Computer Science 9582, 3-24 (2016).

Summary: This note is based on the plenary talk given by the second author at MACIS 2015, the Sixth International Conference on Mathematical Aspects of Computer and Information Sciences. Motivated by some of the work done within the Priority Programme SPP 1489 of the German Research Council DFG, we discuss a number of current challenges in the development of Open Source computer algebra systems. The main focus is on algebraic geometry and the system Singular.

Figure 1: An example of a typical software citation.

Such a citation practice for software is more or less typical not only in mathematics. Howison and Bullard [9] analyzed nearly 300 software references in biology, see Fig. 2.

Mention Type	Count (n=286)	Percentage
Cite to publication	105	37%
Cite to users manual	6	2%
Cite to name or website	15	5%
Instrument-like	53	19%
URL in text	13	5%
In-text name only	90	31%
Not even name	4	1%

Figure 2: Software citing in biology.

A lot of initiatives, being run by e.g., software companies, publishers, or repositories have discussed and developed proprietary recommendations for software citations. Mike Jackson has given in his blog [11] a detailed state of the art analysis and pitfalls of software citation and recommendations for a better citation practise. Currently, the FORCE 11 Software Citation Working group [18], an international initiative of more than 50 information experts from different scientific areas, has discussed basic concepts for software citation. As one result the group has published the Software Citation Principles (SCPs) [16]. They emphasize that software is a legitimate product of research and therefore must be citable. The six principles address the importance of software within research which should manifest that all relevant software will be cited, that software citations should facilitate giving credit and attribution to the developers and contributors of software, include methods for a unique identification, refer to persistent information about software, should facilitate access to the software, and provide accurate information about software (e.g., the version used). The SCPs define a general frame for software citations and moreover formulate principles for maintaining of information about software.

The SCPs do not discuss the realization of the principles. This is planned to be subject for a follow-up working group.

Exact information about software—together with the data used—is a necessary condition to evaluate and reproduce scientific results which were achieved by using mathematical software. Therefore, for the citation format, the following recommendation is given: "We recommend that all text citation styles support the following: a) a label indicating that this is software, e.g., [software], potentially with more information such as [Software: Source Code], [Software: Executable], or [Software: Container], and b) support for version information, e.g., Version 1.8.7" [16]. Each researcher who uses a software for research and publishes her or his results (e.g., in form of a paper, software or data file) is recommended to cite software according to this recommendation.

The mathematical community uses the TEX format for publishing. References are encoded in the (outdated) BibTEX format or, more up to date, in the BibLATEX format. Actually, neither BibTEX nor BibLATEX support a type "software". Up to now, the BibLATEX standard contains no special document type for software. Software must be typed as "misc". A pragmatic recommendation is that it should be added to the title if a citation refers to a software, together with detailed information about the software instance. This could be done in the following form: title [Software:special type (Source Code, Package, Executable, Library, Other)] [Version or release or URL and/or date of the update and /or date of the download]. This would be a first step to a better software citation practice and provides the required information for the human user.

A more rigorous and Semantic Web compatible solution would be an extension of BibLATEX standard. BibLATEX together with the back-end software Biber provides the opportunity to define new document types, e.g., software and the corresponding fields. A prototype for a TEX implementation is under development within the framework of the swMATH activities. It is planned to provide a template for TEX encoding of software citations.

The BibLATEX encoding allows also a simple transformation to other formats, e.g., JSON, which can be used for a machine-based semantic processing of software citations.

Comment: Also software which cites another software should contain the corresponding notations. This could be done by separate citation files which are encoded in the same form as for papers, e.g., in BibLATEX.

The SCPs recommend that "the software itself should be cited on the same base as any other research product", and should have a unique and persistent identifier, preferably a DOI. This does not mean that the DOI is assigned to the software code. Outdated software is often removed from the Web. Instead, "the software identifier should resolve to a persistent landing page that contain metadata and a link to the software itself, rather than directly to the source code files, repository, or executable", [16]. The problem of persistent identifiers and landing pages, especially of a DOI, is connected with additional efforts. Up to now, the existing landing pages, e.g., portals and software directories, provide only metadata about families of software which is offered under the same name, not about versions.

It seems to make sense that—similar to publications—persistent identifiers should be provided and maintained by special information services which integrate the information about software in a subject and make it available. The SCPs make clear that a citation standard would be very helpful for better software information but it is only a building block in a better digital information infrastructure for software and scientific information in general. That is why we continue with a brief description about Web resources which are relevant for mathematical software information.

3 The landscape of mathematical software information in the Web

The landscape of Web resources of mathematical software information is heterogeneous, widely distributed, and has different layers. Here is an incomplete list of the relevant resources:

- *Individual websites of a software*
  This is in some sense the basic layer of the software information infrastructure. Websites exist for many though not all software packages (from our experiences in the swMATH project we

estimate that nearly two thirds of mathematical software packages provide information on own websites). Typically, these websites contain a lot of detailed information about a software, documentation, manuals, tutorials, software code (if the software is free), the programming language used, contact information, usability and licences, hard- and software requirements, publication lists, etc.

- *Repositories*
  Software repositories as "The Comprehensive R Archive Network (CRAN)" [15] or "The Comprehensive Perl Archive Network (CPAN)" [14] provide and maintain metadata plus the source code of software collections. CRAN is a repository for statistical software written in the R language and presents standardized meta information, the version history, and links to the source code for nearly 10,000 packages.

- *Portals, directories and information services*
  Portals or directories of software provide lists of software, metadata, and links. "Fachgruppe Computeralgebra" [2] or "SIGSAM" [17] offers structured webpages for computer algebra systems. These services, like the Symbolic Data project [6], are not limited to information about software but also on conferences and workshops, researchers, and data. We will discuss this in more detail below.

  An informative list of computer algebra systems can also be found in Wikipedia [20].

System	Creator	Development started	First public release	Latest stable version	Latest stable release date	Cost (USD)	License	Notes
Axiom	Richard Jenks	1977	1993 and 2002[7]		August 2014[8]	Free	modified BSD license	General purpose CAS. Continuous Release using Docker Containers
Cadabra	Kasper Peeters	2001	2007	2.1.2	January 4, 2017	Free	GNU GPL	CAS for tensor field theory
CoCoA-4	The CoCoA Team	1987	1995	4.7.5	2009	Free for non-commercial use	own license	Specialized CAS for commutative algebra
CoCoA-5	Abbott, Bigatti, Lagorio	2000	2011	5.1.1	2014	Free	GNU GPL	Specialized CAS for commutative algebra
Derive	Soft Warehouse	1979	1988	6.1	November 2007	Discontinued	Proprietary	CAS designed for pocket calculators; it was discontinued in 2007
DataMelt (DMelt)	jWork.ORG (Sergei Chekanov)	2005	2015	1.5	May 14, 2016	$0 for academic usage, commercial license unknown	Proprietary	Java-based. Runs on the Java platform. Supports Python, Ruby, Groovy, Java and Octave.
Erable (aka ALGB)	Bernard Parisse, Mika Heiskanen, Claude-Nicolas Fiechter	1993	1993	4.20060919	April 21, 2009	Free	LGPL	CAS designed for Hewlett-Packard scientific graphing calculators of the HP 48/49/40/50 series; discontinued in 2009
Fermat	Robert H. Lewis	1986	1993	5.25	July 5, 2016	$70 if grant money available, otherwise $0	Proprietary	Specialized CAS for resultant computation and linear algebra with polynomial entries
FORM	J.A.M. Vermaseren	1984	1989	4.1	October 25, 2013[9]	Free	GNU GPL	CAS designed mainly for particle physics
FriCAS	Waldek Hebisch	2007	2007	1.3.0	August 31, 2016	Free	modified BSD license	Full-featured general purpose CAS. Especially strong at symbolic integration.
GAP	GAP Group	1986	1986	4.8.6	November 12, 2016	Free	GNU GPL[10]	Specialized CAS for group theory and combinatorics.
GiNaC	Christian Bauer, Alexander Frink, Richard B. Kreckel, et al.	1999	1999	1.7.1	October 2, 2016	Free	GNU GPL	Integrate symbolic computation into C++ programs; no high-level interface, but emphasis on interoperability.
KANT/KASH	KANT Group	?	?	3	2005/2008	Free for non-commercial use	own license	Specialized CAS for algebraic number theory
Macaulay2	Daniel Grayson and Michael Stillman	1992	1994	1.8	2015	Free	GNU GPL	Specialized CAS for algebraic geometry and commutative algebra
Macsyma	MIT Project MAC and Symbolics	1968	1978	2.4	1999	$500	Proprietary	The oldest general purpose CAS. Still alive as Maxima.
Magma	University of Sydney	~1990	1993	2.22-3	July 20, 2016	$1,440	Proprietary	General purpose CAS, originally specialized in group theory. Works with elements of algebraic structures rather than with non typed mathematical expressions
Maple	Symbolic Computation Group, University of Waterloo	1980	1984	2016	March 2, 2016	$2,275 (Commercial), $2,155 (Government), $1245 (Academic), $239 (Personal Edition), $99 (Student), $79 (Student, 12-Month term)[11]	Proprietary	One of the major general purpose CAS
Mathcad	Parametric Technology Corporation	1985	1985	15.0 M045	November 2015	$1,600 (Commercial), $105 (Student), Free (Express Edition)[12]	Proprietary	Numerical software with some CAS capabilities
Mathematica	Wolfram Research	1986	1988	11.0.1 (September 28, 2016) [±][13]	April 18, 2016	$2,495 (Professional), $1095 (Education), $295 (Personal),[14] $140 (Student), $69.95 (Student annual license),[15] free on Raspberry Pi hardware[16]	Proprietary	One of the major general purpose CAS

Figure 3: A snippet of the Wikipedia (I): list of computer algebra systems.

The lists mentioned here are manually maintained and updated, have different structure also for metadata and are weakly coordinated, see also the remarks about the Computer Algebra Social Network (CASN) below.

Functionality [edit]

Below is a summary of significantly developed *symbolic* functionality in each of the systems.

System	Formula editor	Arbitrary precision	Calculus		Solvers					Graph theory	Number theory	Quantifier elimination	Boolean algebra	Tensors	Probability	Control theory	Coding theory	Group theory
			Integration	Integral transforms	Equations	Inequalities	Diophantine equations	Differential equations	Recurrence relations									
Axiom	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Magma	No	Yes	No	No	Yes	No	Yes	No	No	Yes	Yes	No	No	No	?	?	Yes	Yes
Maple	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes
Mathcad	Yes	No	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No
Mathematica	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes[20]	Yes	Yes	No	Yes
Mathomatic	No	No	Yes	Yes	Yes	No	No	No	No	No	Yes	No	No	No	No	No	No	No
Symbolic Math Toolbox (MATLAB)	No	Yes	Yes	Yes	Yes	No	No	Yes	No	No	No	No	No	No	No	No	No	No
Maxima	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes
SageMath	No	Yes	Yes	Yes	Yes	Yes	Yes[A]	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
SymPy	No	Yes	Yes	Yes	Yes	Yes	Yes[21]	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	Yes
Wolfram Alpha	Pro version only	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	?	?	No	Yes
GAP	No	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Xcas/Giac	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes	?	No	?
Yacas	No	Yes	Yes	No	Yes	No	No	No	No	No	No	No	No	No	?	?	No	No

Figure 4: A snippet of Wikipedia (II): functionalities of computer algebra systems

- *Further resources*

  - Services
    Software can be available in different forms, e.g., as a service (cloud computing, Class Group Database [1, 13]).
  - Journals specialized in mathematical software
    Also the software journals are mentioned here because they play a pioneering role for the quality control and evaluation of software. There exists a number of journals specialized in mathematical software, e.g., the Journal of Software for Algebra and Geometry [12], where the peer reviewing also includes mathematical software.
  - Conferences (including proceedings) on CASs

Moreover, mathematical software must be considered in its context which is given by mathematical theories, algorithms, programming languages, applications, data, e.g., benchmarks and data formats, and also the developers and user communities of software. Context analysis is also an important method to build up and develop powerful machine-based information services for mathematical software which will be demonstrated by the swMATH concept in the next section.

4 swMATH

4.1 The publication-based-approach

An important feature of the swMATH [3] concept, the publication-based approach, has its origin in the analysis of context information. Instead of analyzing mathematical publications for software citations and information about software, the database zbMATH [4] is used. Especially, the zbMATH database contains the following data of publications which are of relevance for the analysis: title, keywords, reviews or abstracts, reference lists, and classification codes. The data analysis and knowledge generation of the swMATH approach has several steps:

1. *Identification of software references within the zbMATH data*
   The title, the review or abstract, and the reference lists of publications are searched for indicators for software references by heuristic means. Such indicators can be artificial names in combination with characteristic words as software, module, package, etc. Of course, the methods used are very simple but work surprisingly well. The acceptance of a software citation standard corresponding to the recommendations given above would make the heuristic methods obsolete and permit a secure and complete identification of software.

2. *Extraction of information about software*
   Reviews and abstracts of a mathematical publication contain most of all content information, especially a description of the problems investigated, the used methods, and results.
   For software references it is useful to distinct between two classes of publications: The publications which describe a software (labeled as "standard publications") and publications which use a software for solving a problem (labeled as "user publications"). Both types provide different information about a software and are processed in different ways. Some information directly enters into swMATH, e.g., keywords or MSC codes.

3. *Aggregation and ranking of information*
   Currently, swMATH has nearly 16,000 entries on software packages and other mathematical research data which contain more than 215,000 software citations in more than 130,000 publications. In other words, there is often a great number of publications citing a software. This allows to weight the information by the corresponding number of the keyword frequencies which is done in the keyword cloud, to create an "acceptance profile" of the software (citation graph) by the number of annual publications citing a software. It's also possible to give some information about related software based on the MSC classification codes. The number of publications citing a software could also be used as a measure for credit to the developers. Further features are possible, e.g., the definition of an application profile of the software. All this can be done automatically by heuristic means.

4.2 The Web-based approach

The publication-based method is a powerful tool but has limitations. Publications do not cover technical details about the implementation, the programming language, or the required hard- and software environment of a software. Typically, this information is given in manuals and documentations. Also other context information, e.g., test data and benchmarks or programming languages, are important for reproducing the results of a publication. As said above, this kind of information can often be found on other resources on the Web.

Therefore swMATH tries to enrich the information about software by adding information from the Web. At first, swMATH tries to find the website of a software and links it if the search was successful. We have started to develop methods for analyzing the websites, see [8]. For this purpose the Internet Archive [10] is used which provides also the data from a lot of websites of mathematical software from the past. Also the information of some repositories is integrated in swMATH. swMATH shows that the analysis of different resources is a promising way to run and maintain useful and efficient information services for mathematical software.

4.3 swMATH in a nutshell

swMATH lists more than 100 entries for CASs. A swMATH search for “computer algebra system” shows a list of CASs

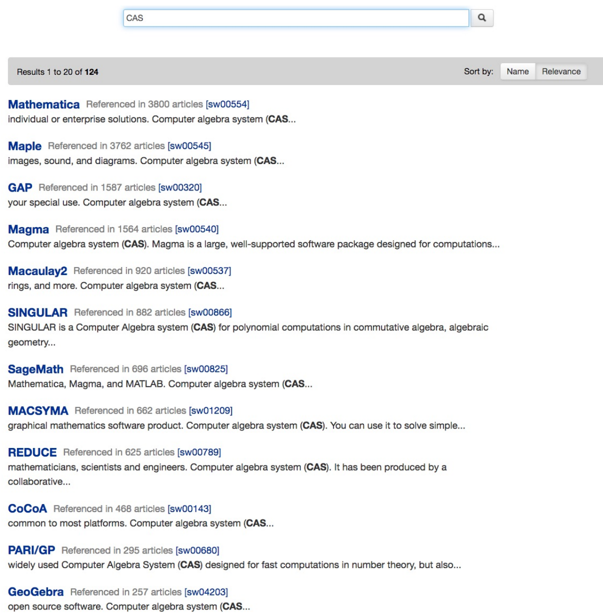

Figure 5: A snippet of the swMATH list of computer algebra systems.

Each CAS is presented by an own webpage, see, e.g., a cutout of the swMATH webpage for the software “Singular” in Fig. 6.

Typically, swMATH provides the following information:

1. a list of mathematical software packages (and other related mathematical research data), as complete as possible
2. a persistent identifier (a five digit number) for each software,
3. metadata, especially about its content (description, keywords, MSC codes),
4. links to the websites of the software (if existing).
5. a list of publications citing a software
6. a list of similar software
7. an acceptance profile for the software
8. links from the publications to the corresponding versions (currently only in the test version)
9. links to the Internet archive
10. links to manuals, documentation, source code, etc.

Moreover, swMATH provides a simple and an extended search functionality.

Figure 6: A snippet of the swMATH webpage for the software "Singular".

5 The Computer Algebra Social Network

Software information is an important part of the scientific digital information and communication infrastructure. The scientific digital infrastructure is widely distributed and must be able to process and link heterogeneous resources, e.g., information about researchers, publications, software, conferences and workshop, etc. and data formats in a semantic way. This requires an active goal-oriented conceptual and technical cooperation between different players. All relevant data must be digitized, semantically enriched and encoded in a machine-understandable way. The idea of the Computer Algebra Social Network (CASN), for an overview on CASN see H.-G. Gr"abe [7], is an advancement and continuation of the Symbolic Data concept. Established Web technologies, especially RDF, should be used for the semantic annotation of resources. Each resource, e.g., the servers of the German CA Fachgruppe or SIGSAM, can become a node in this network. The RDF files of the metadata which are created by the provider of a node guarantee that the information can be automatically linked and is accessible via CASN. The swMATH service is integrated in CASN.

6 Summary and outlook – what we can and should do

A powerful and sustainable infrastructure for mathematical software information is in the interest of developers and users. It is also important for the positioning and the role of this research subject in the sciences and in society. The infrastructure must be oriented towards the interests of the developers and users. The mathematical community should actively take part and influence the development. Specifically, the information about software is inconsistent, is distributed, not standardized, and not machine-processable, which hampers the combination of heterogeneous resources of mathematical software and related information. A better citation practice, standardization, enrichment of the semantic information, coordination, and a better integration of software as desired in the Open DreamKit project [19] opens new perspectives for this research subject. We need a broad dialogue and a communication forum which brings the developers, the user communities, information experts, and service providers together for a discussion of all aspects. The symbolic computation community could play a pioneering role to establish a sophisticated infrastructure for a mathematical subject which covers all relevant resources. These include

- definition of the overall goals and principles of an infrastructure for mathematical software

- standardization
  for the authors: The authors should cite the software corresponding to the recommendations of the SCPs, especially marking up the type and give information about the versions, releases, etc,
  for developers and service providers: There should be developed a standardized metadata scheme for mathematical software (analyzing the different facets of software information),
  for service providers: The information should be provided in a machine-understandable way which supports a semantically sensible combination of information,
  for service providers: Development of intuitive tools to support the standardized description of citations.

- a better linking of the information resources of software for service providers: This requires a cooperation between the providers of the different services for software and its context and the development of user interfaces.

Acknowledgement

We are grateful to the Fachgruppe Computeralgebra to help us reach a wider audience by additionally publishing this article in the “Computeralgebra Rundbrief”, issue 60 (March 2017).

References

[1] Boy, Maximilian: A Database for Class Groups of Number Fields, homepage. `http://www.mathematik.uni-kl.de/~numberfieldtables/`

[2] Fachgruppe Computeralgebra: Homepage. `http://www.fachgruppe-computeralgebra.de/systeme/`

[3] FIZ Karlsruhe, ZIB Berlin: swMATH Homepage. `http://www.swMATH.org`

[4] FIZ Karlsruhe: zbMATH Homepage. `http://www.zbMATH.org`

[5] Gräbe, Hans-Gert: RDF File of CA Conferences. `http://symbolicdata.org/rdf/Conferences/`

[6] Gr”abe, Hans-Gert: The SymbolicData Project. `http://wwww.symbolicdata.org`

[7] Gräbe, Hans-Gert: The SymbolicData Project – Maturing the Computer Algebra Social Network Perspective. *Computeralgebra Rundbrief 59, p. 17-21*, `http://symbolicdata.org/Papers/aca-16-paper.pdf`

[8] Holzmann, Helge; Sperber, Wolfram; Runnwerth, Mila: Archiving Software Surrogates on the Web for Future Reference. *In: Norbert Fuhr, Lászlo Kovács, Thomas Risse, Wolfgang Neidl, Research and Advanced Technology for Digital Libraries: 20th International Conferences on Theory and Practice of Digital Libraries, TPDL 2016, Hannover 2016, LNCS 9819, p. 215 - 226,*

[9] Howison, J.; Bullard, J.: Software in the scientific literature: Problems with seeing, finding, and using software mentionend in the biology literature. *Journal of the Association for Information Science and Technology, 2015,*
`http://dx.doi.org/10.1002/asi.23538`

[10] Internet Archive: Homepage.
`https://archive.org/web/`

[11] Jackson, Mike: How to cite and describe software?
`https://www.software.ac.uk/how-cite-and-describe-software`

[12] Journal of Software for Algebra and Geometry Editorial Board: Homepage.
`http://msp.org/jsag/about/cover/cover.html`

[13] Klüners, J; Malle, G: A Database for Number Fields, homepage.
`http://galoisdb.math.uni-paderborn.de/`

[14] Perl Foundation: The Comprehensive Perl Archive Network. `http://www.cpan.org/`

[15] R Foundation: The Comprehensive R Archive Network. `https://cran.r-project.org/`

[16] Smith, Arfon M.; Katz, Daniel S.; Niemeyer, Kyle E.; FORCE11 Software Citation Working Group: Software citation principles.
Peer J. Computer Science 2:e86, 2016,
`https://doi.org/10.7717/peerj-cs.86`

[17] Special Interest Group on Symbolic and Algebraic Manipulation of the ACM: Homepage.
`http://www.sigsam.org/Resources/Software.html`

[18] Software Citation Working Group: Homepage, `https://www.force11.org/group/software-citation-working-group`.

[19] Thiéry, Nicolas M.: OpenDreamKit: Open Digital Research Environment Toolkit for the Advancement of Mathematics. *Computeralgebra Rundbrief 57, p. 17-18,*
`http://www.fachgruppe-computeralgebra.de/data/CA-Rundbrief/car57.pdf`

[20] Wikipedia Community: The Wikipedia Web page of CASs.
`https://en.wikipedia.org/wiki/Listofcomputeralgebrasystems`