

CATO - a general user interface for CAS

Hans-Dieter Janetzko, HTWG Konstanz (University of Applied Science)

Often, scientists, engineers, and students apply computer algebra systems only occasionally. Rare usage of grammar and syntax is a big obstacle to them. Therefore, the author proposes a general user interface for many computer algebra systems (currently Maple, Mathematica, Maxima, MuPAD, MuPAD inside MATLAB, and Yacas) named CATO.

The interaction is independent of the used computer algebra system, and CATO implements various concepts for a guided use of computer algebra. Commands with more than one parameter have a two dimensional input structure, parameters being explained and the right kind of separators and brackets are set by CATO. Selecting and adjusting options for commands is always similar. Additionally, the help contains for each command often used paraphrases. Currently, CATO is based on German, but the proposed concept could be easily extended to other languages.

In CATO, commands are combined to packages simplifying the access and selection. Commands may be contained in more than one package, as they might belong to more than one concept. The "definition of a vector", for instance, is contained in both the "definition" and the "linear algebra" package. Accessing the most recently used commands is possible by lists storing the history of command invocations. Users are also able to define such a list as a new package and can export and import these as their own packages.

key notes: graphical user interface, computer algebra system, Java

1 Previous approaches and considerations to CA user interfaces

The demands to better designs of user interfaces for computer algebra systems are almost as old as these systems. Kajler has described and developed his ideas of a perfect user interface in various works (Kajler, 1992 & 1993). Finally he has elaborated those in (Kajler, 1998) and (Kajler & Soiffer, 1998).

The interface should be designed to provide an intuitive access. Therefore it should be possible to enter commands with more than one parameter in a two-dimensional fashion. This prevents syntactic and structural errors. All templates and masks should be to used from left to right as common.

The usual mathematical notation should apply, of course. Moreover, the surface should be decoupled of the CAS and be serviced independently, developed and updated. Ideally it should understand not just one CAS but also more different systems.

A couple of years later, the ideas of Kajler were further improved. Cojocaru et al. gave a new description of a good interface in (Cojocaru, S., Malahova, L. & Colesnicov, A., 2006). “The system should be self-explanatory” and not “based on a (specialized) programming language” was postulated. These authors agreed to Kajler that the interface should control more than one CAS. All problems using computer algebra are caused of “the absence of a user-friendly standard”.

Another approach to using different CA systems through the same interface was presented in (Oancea, C.E. & Watt, S.M., 2006). Aldor applications had direct access to Maple objects. It was a connection of “high-level mathematical libraries” with Maple. Here, Alma worked as an interface between Aldor and Maple, but the user had to learn this programming language.

Of course, the developers of CAS know these problems. They searched for solutions with different approaches, but mostly resigned with the next version of the applications respectively performed these ideas and approaches not consistently.

2 CATO as sublimation of own experiences and wishes of the students

At the university of applied sciences in Konstanz the author used various CA systems in the mathematic lectures, first Maple, then Mathematica and at last Maxima. His experiences of the occasional usage of CA are complex and not wholesome for all students.

- The good students are fascinated and become better. This is not surprising. They can be motivated by basically everything bringing them forward.
- Intermediate students will accept CA, perhaps because it is relevant for the tests. But they will not be better through the usage. They will not see in CA and the permission for its usage the potential but rather a stumbling block.
- The weak students finally resign in mathematics because they see just the additional effort for learning the vocabulary, syntax and grammar. They think the calculation of examples to the last details will help them to

understand the abstract facts. If you retain them this support for no apparent compensation they feel deprived of comprehension.

The author liked to facilitate the access to computer algebra for his students. Therefore he developed an input interface (first for Mathematica) to be able to use computer algebra casually in his courses. Thereby he followed many concepts of Kajler on the one hand. On the other hand he considered his experiences and reduced a lot of the inhibitions of the casual usage of CA. By the connection of further CA systems he generalized his concept and the internal commands. He was always in contact to the students for further improvements. These suggestions he integrated in the evolution of CATO and its concept.

3 The CATO system: first steps

After starting up CATO, the user interface of CATO will be shown. There are two text areas, several buttons, and combo boxes visible. At first, the student should select the CA system to use. All CA systems denoted here have to be installed and referenced during the installation process (see figure 1). CATO will instantiate a connection to the chosen CA system and output the version string of the CAS to the second text area, reserved for outputs.

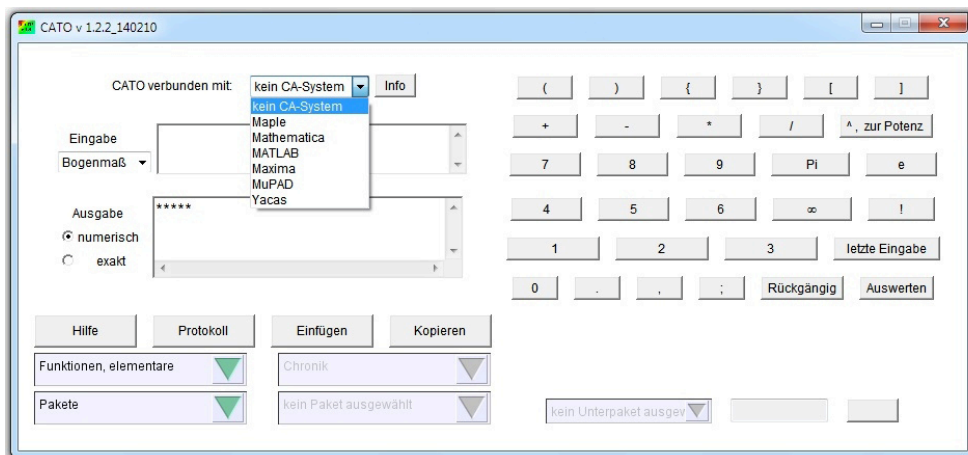


Fig. 1: The ComputerAlgebraTaschenrechnerOberfläche

Now CATO is ready to execute mathematical commands. CATO can be used as a common calculator and students can enter simple commands, such as “sin(4.5)”. The student may either click the button “Auswerten” (evaluate) or press the return key. Nearly all simple, one parameter functions, can be entered manually with the keyboard. Of course the student has to know the names of the commands for manually entering them. The author will show the exceptions to this rule of thumb later on.

Let us now regard a more interesting case. The student knows he has to differentiate a function, but isn't sure about the name of the respective command, because he forgot it. In these cases, the help of CATO will come to his aid. The help of CATO is accessible via a button on the left side, labelled with the word "Hilfe" (help).

The help of CATO is a HTML document and therefore independent of the used CAS and operating system. Furthermore, the help is always available even when CATO is not running. We do also offer the help online on the author's website. The help is structured as follows: there is an alphabetically ordered list of all commands and their synonyms on the left side. The most space is given to the respective help text.

CATO Information

[1](#) [2](#) [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Y](#) [Z](#)

[exakt](#)
[DGL, 4-ter Ord., AWP, exakt](#)
[Diagonalmatrix erzeugen](#)
[Differential- und Integralrechnung einer Veränderlichen](#)
[Differential- und Integralrechnung mehrerer Veränderlicher](#)
[Differential, totales](#)
[Differentialgleichungen](#)
[Differenziation](#)
[Differenzieren](#)
[Differenzieren, n-mal](#)
[DiracDelta](#)
[Diracsche Deltadistribution](#)
[Diskrete Gleichverteilung](#)
[Diskrete Gleichverteilung, Erwartungswert](#)
[Diskrete Gleichverteilung, Kurtosis](#)
[Diskrete Gleichverteilung, Quantil](#)
[Diskrete Gleichverteilung, Verteilungsfunktion](#)
[Diskrete Gleichverteilung, Wahrscheinlichkeitsfunktion](#)
[Diskrete Wahrscheinlichkeitsverteilung](#)
[Distribution, Dirac'sche](#)
[Divergenz](#)
[Divergenz in x,y,z](#)
[Division](#)
[Division von Brüchen](#)
[Division von Polynomen](#)
[Divisionsrest bei Polynomen](#)
[Divisoren](#)
[Divisoren, Anzahl bestimmen](#)
[Divisoren, Anzahl der](#)
[Divisoren, Anzahl der ... einer Zahl](#)
[Doppelintegral](#)
[Drei-D-Graphik](#)
[Dreiecksgestalt, obere](#)

Ableiten
 CATO: (Paket) Analysis I (auswählen) → Ableiten (auswählen)
 CATO: (Paket) Funktionen, elementare → Ableiten (auswählen)
 Maple, Mathematica, math, Toolbox, Maxima, MuPAD, Yacas

Mit diesem Befehl kann man die erste Ableitung einer Funktion in einer Veränderlichen bestimmen. Die erste Ableitung einer Funktion (bzw. das einmalige Differenzieren einer Funktion) beschreibt die Veränderung dieser Funktion in Abhängigkeit von der Variablen. Somit ist die Ableitung positiv, wenn die Funktionswerte mit wachsender Variablen zunehmen, und die Ableitung ist negativ, wenn die Funktionswerte mit wachsender Variablen abnehmen. Dadurch kann man dann auch (strenge) Monotonie [überprüfen](#). Selbstverständlich kann Computeralgebra [Produkte und Quotienten, sowie Verkettungen von Funktionen](#) differenzieren. Natürlich kann man bei einem Ausdruck mit mehreren Unbekannten auch [gezielt nach einer Unbekannten](#) ableiten. Für [höhere Ableitungen](#) oder [partielle Ableitungen](#) gibt es eigenständige Befehle.

Dieser Befehl ist in einigen CA-Systemen auch [listenfähig](#).
 Außerdem ist dieser Befehl in CA-Systemen auch auf Funktionen mit [komplexe Argumente](#) anwendbar.
 Der Befehl zum **Ableiten** befindet sich in dem Paket [Funktionen, elementare](#) sowie in dem Paket [Analysis I](#).

BEISPIELE:
 Bsp.: Es soll die Ableitung von $x^2/(x-5)^2$ gebildet werden. Dazu wählen wir in CATO links unten unter **Pakete** zuerst das Paket **Analysis I** aus, dann rechts daneben in diesem Paket den Befehl **Ableiten**. Es erscheint das Zusatzfenster „Ableiten“ mit dem Hinweis „Eine Funktion wird einmal nach der Variablen abgeleitet!“ und zwei Eingabezeilen:

Funktion:	
Variable:	

Wir sind automatisch in der ersten Eingabezeile positioniert und tippen ein:
 $x^2/(x-5)^2$
 und betätigen **weiter**!. Danach sind wir in der zweiten Eingabezeile und geben ein:
 x

Wir beenden die Eingabe mit **weiter**!, schließen des Fensters mit **alle Eingaben abgeschlossen** und können mit **Auswerten** den Befehl an CATO abschicken.

Wenn **Maple** angeschlossen ist, erhalten wir als Antwort:
 $3*x^2/(x-5)^2 - 2*x^3/(x-5)^3$

Wenn **math, Toolbox** angeschlossen ist, erhalten wir als Antwort:
 $3*x^2/(x-5)^2 - 2*x^3/(x-5)^3$

Wenn **Mathematica** angeschlossen ist, erhalten wir als Antwort:

$$\frac{3x^2}{(x-5)^2} - \frac{2x^3}{(x-5)^3}$$

Wenn **Maxima** angeschlossen ist, erhalten wir als Antwort:

$$\frac{3x^2}{(x-5)^2} - \frac{2x^3}{(x-5)^3}$$

Wenn **MuPAD** angeschlossen ist, erhalten wir als Antwort:
 $3*x^2/(x-5)^2 - 2*x^3/(x-5)^3$

Wenn **Yacas** angeschlossen ist, erhalten wir als Antwort:
 $(3*x^2/(x-5)^2 - 2*x^3/(x-5)^3)/(x-5)^4$

Bsp.: Es soll die Ableitung von $a*g^3 + c*g^7$ nach g bestimmt werden. Dazu wählen wir in CATO unter dem Ausgabefenster links im Paket **Funktionen, elementare** den Befehl **Ableiten** aus. Es erscheint das Zusatzfenster „Ableiten“ mit dem Hinweis „Eine Funktion wird nach der Variablen abgeleitet!“ und zwei Eingabezeilen:

Funktion:	
Variable:	

Fig. 2: The help of CATO, the command "Ableiten" (differentiate) is chosen.

The student of our previous example knows he wants to differentiate a function. He will find three different synonyms for this task. Namely “Ableiten”, “Differenzieren”, and “Differentiation”. Each of this names is linked to the very same help text for the differentiate command. After the clicking on any of the names the help text will be loaded in the central area of the HTML page (see figure 2).

Every help text is built the same: First is the name of the command (black), followed by the packages containing this command (green), and finally the CAS in which the command is available (red). The following free text contains first an abstract description and after that several comprehensible examples.

4 Selection of commands via packages and menus

Selecting commands in CATO follows a hierarchical paradigm via 27 packages. (4 of these consist of sub-packages). Packages group related commands together, but the same command may be included in more than only one package. For instance, the command “Definition eines Vektors” (definition of a vector) is contained in the packages “lineare algebra” (linear algebra) and “Definitionen” (definitions). The help text will refer to both locations and the student does not need to learn the locations by heart. But the student can rely on the fact that all commands needed for matrix computations are in the package “linear algebra”. Another example is the imaginary unit $i = \sqrt{-1}$. The imaginary unit is contained in the packages “komplexe Zahlen” (complex numbers) and also in the package “mathematische konstante” (mathematical constants). In this case, the user could also just enter i , the command in CATO, on the keyboard.

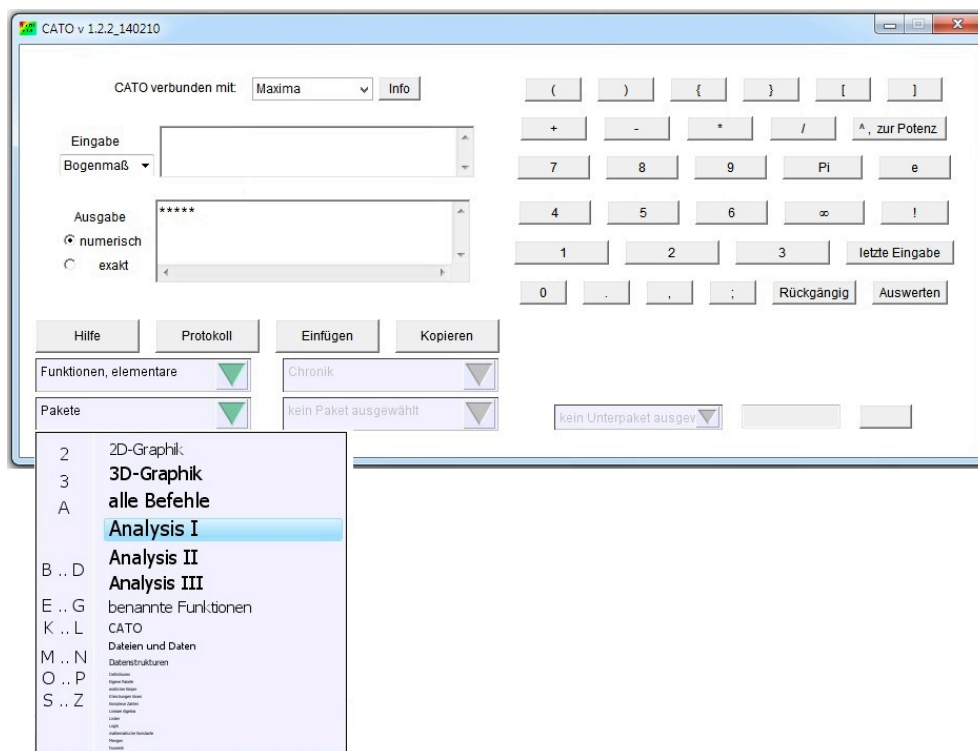


Fig. 3: selection of a package.

The selection of a package is possible via the combo box on the lower left (see figure 3). The selection of a command is therefore performed from left to right, see figure 4. There you can see the selection of the command “ableiten” (differentiate) in the package “Analysis I”.

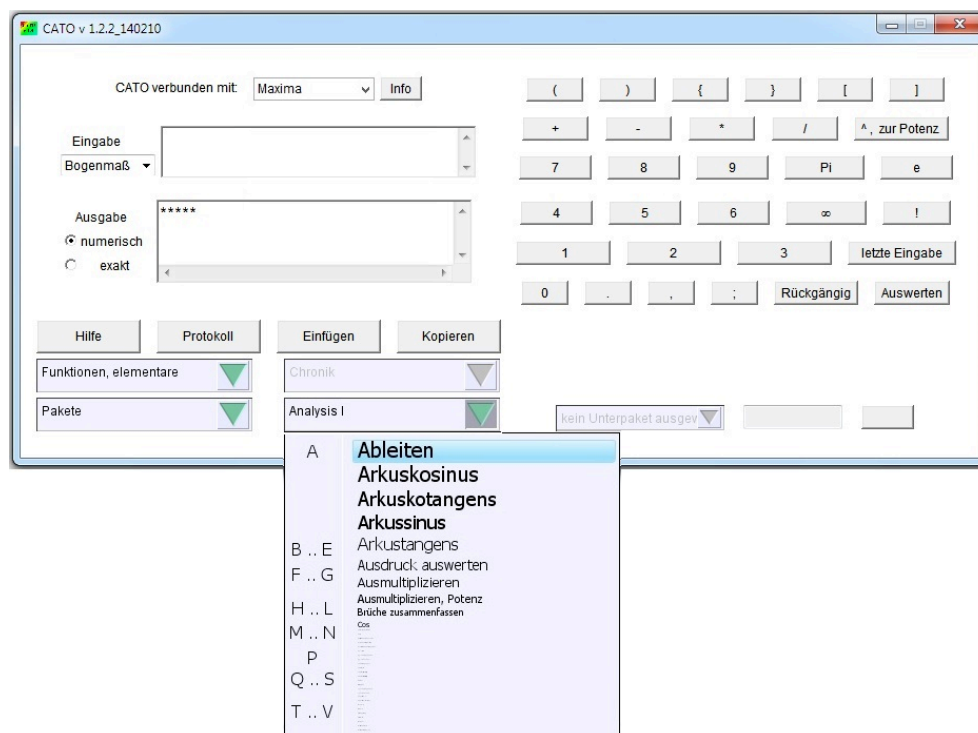


Fig. 4: Selection of a command in the package „Analysis I“.

The selection of command in a sub-package can be performed in the same manner, see fig. 5. There the package “statistics” is selected, followed by a selection of the sub-package “normal distribution” fig 6. The command is selected in fig. 7: “quantil of the normal distribution” from “commands regarding the normal distribution”.

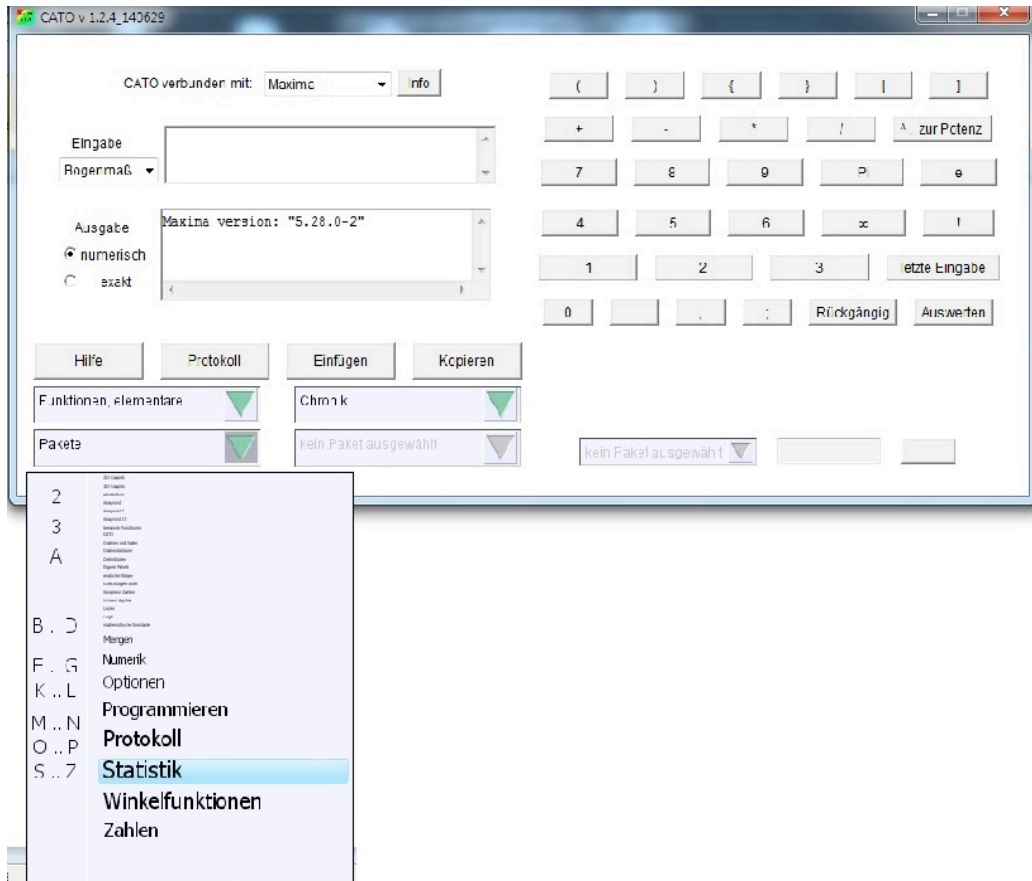


Fig. 5: Selection of "statistics"

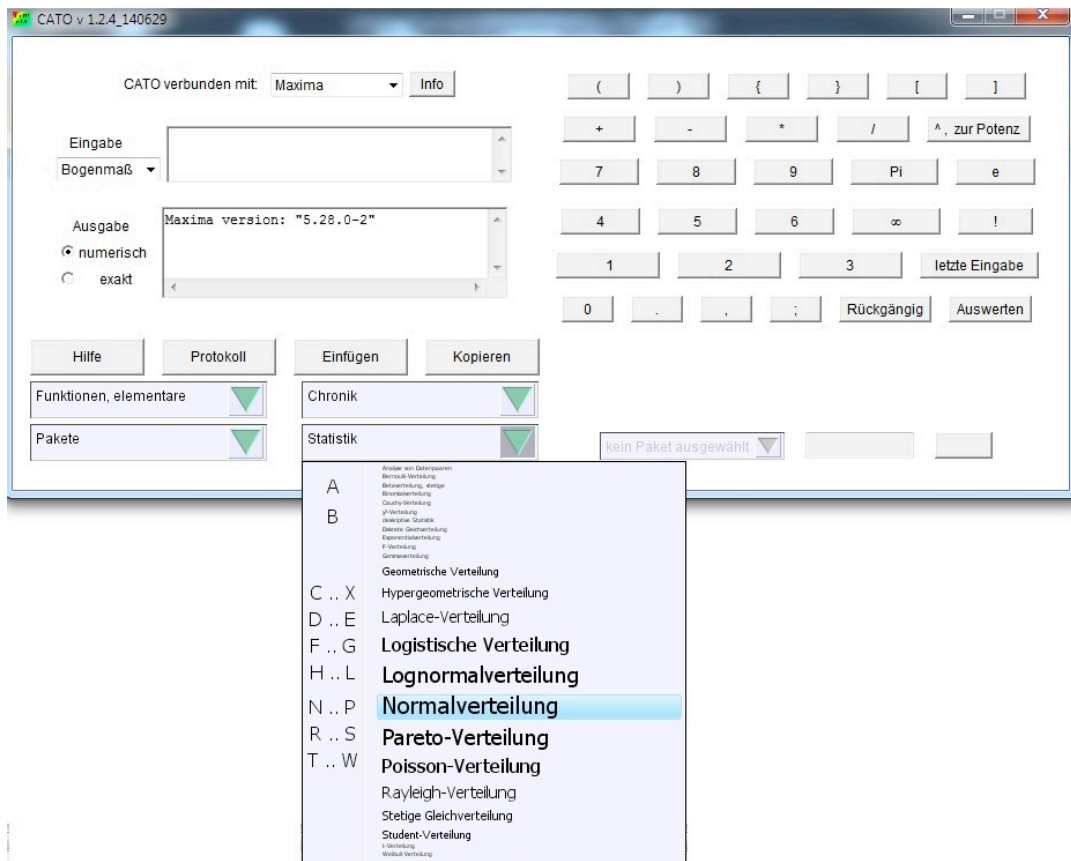


Fig. 6: Selection of the sub-package "Normalverteilung" (normal distribution).

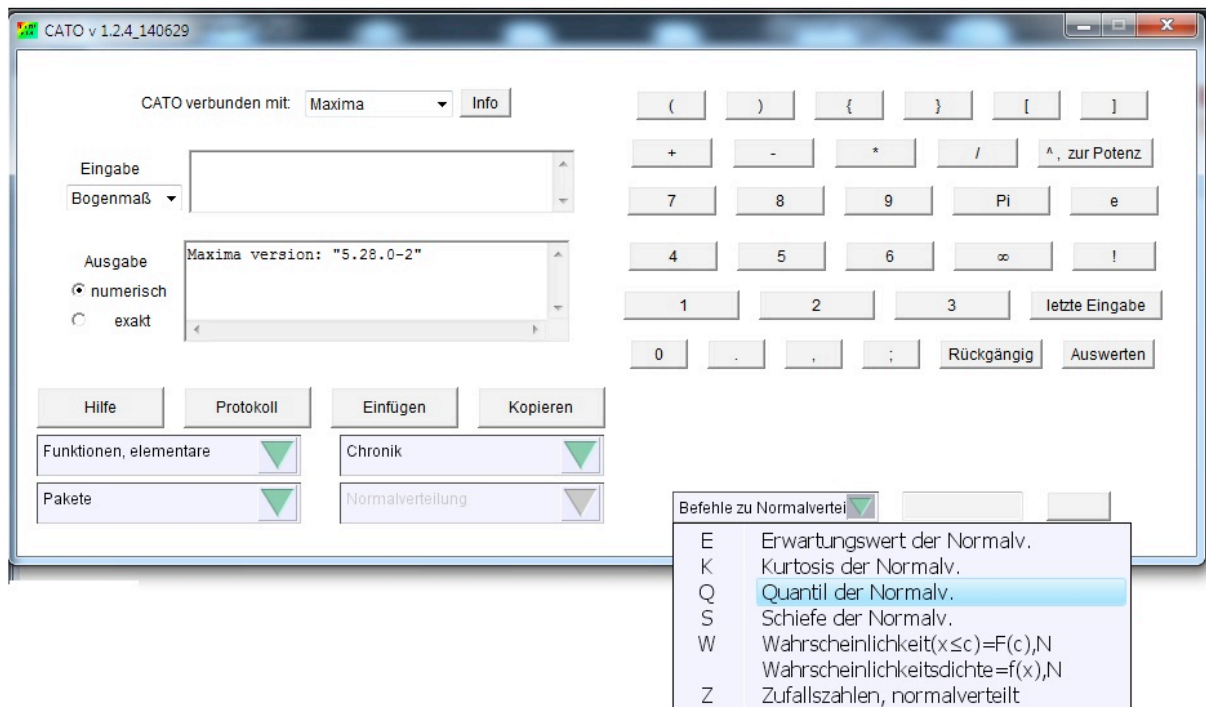


Fig. 7: Selection of one command of the sub-package.

5 Guidance for multi-parameter commands

We set our focus on the user needs when designing the command names and did not just translate the English commands. When designing the command names we asked ourselves two questions: How should the command be labelled in order to find it? How is the inherent structure of the commands regarding the parameter set?

Following these questions we came up with following points:

- The caption of a command is shorter in the combo boxes than in the input text area or in the log.
- One parameter commands can be directly entered and instantiated in the input text area.
- Multi-parameter commands will always invoke a two-dimensional input graphical user interface. Each parameter has its own documented input row. The user does not need to know the right input order of parameters or the right kind of separators or brackets, see fig 8. Internally, CATO will reorder the parameters to fit to the respective CAS requirements. (Some one parameter commands have an additional input GUI. These cases are documented and explained in the help of CATO.)

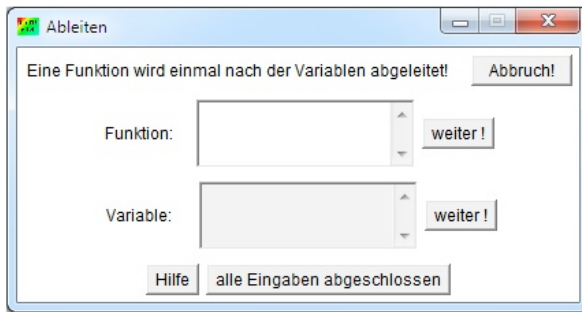


Fig. 8: The input GUI of the command “Ableiten”.

- CATO will enforce the right syntax for equations, by fixing one side to “zero equals ...”. Students have to pre-process and enter their equations accordingly. (see figure 9)

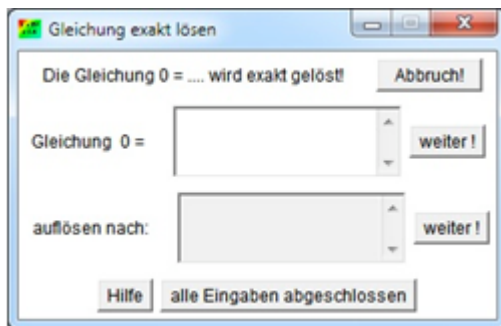


Fig. 9: The input GUI of the command “Gleichung exakt lösen”.

- Options for commands, mostly graphical options, are also entered in a left-to-right-manner with the help of combo boxes.

The usage of CATO is always the same, independent of the chosen computer algebra system. Neither the interface nor the input methods are affected when changing the computer algebra systems.

6 Creating Graphics with CATO

CATO can create graphics using the connected CAS in the same manner as executing calculation commands. There are two graphics-related packages, namely “2D-graphics” and “3D-graphics”. For each of the graphic commands a guided two-dimensional input interface is created and shown to the student. Graphic commands offer additional options, such as size or view perspective. Options can be directly changed during entering the graphic’s parameters or afterwards when viewing the graphical results. The resulting image is shown in its own frame with an option panel below (fig. 10 and 11).

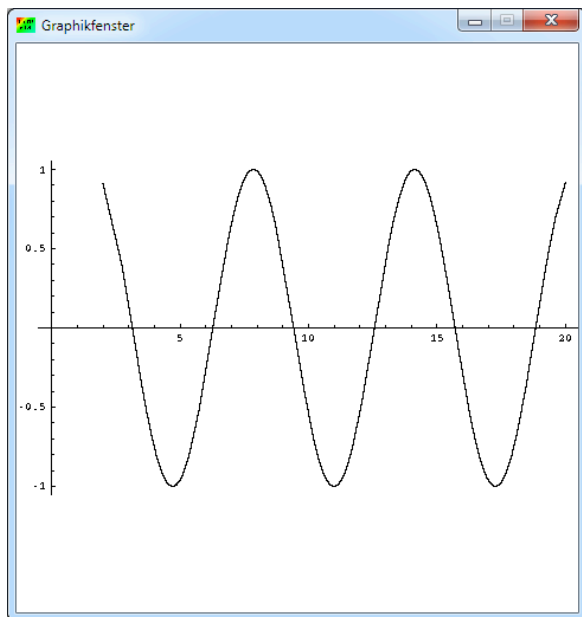


Fig. 10: The frame for graphical output: an image of sine by Mathematica.

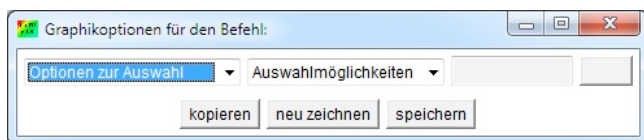


Fig. 11: The option panel.

The options are chosen and changed in the option panel from left to right and after setting the desired options the user can require the system to repaint the image by a button click (see fig 12). CATO will re-execute the command with the new options and display the new result. It is of course possible to re-iterate this process several times.

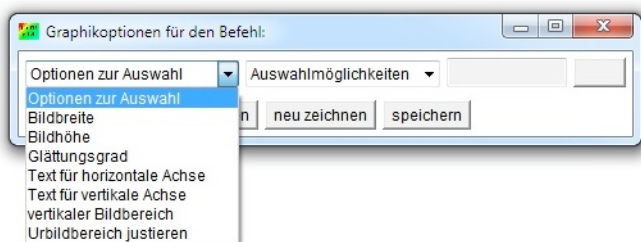


Fig. 12: Possible options for the selected graphics command.

7 Introduction of CATO to lectures

The author uses CATO in the lectures Mathematics One and Two in the study course Electrical Information Technology Bachelor and in Statistics Two in the Subject Business Studies. Note that there is no general lecture “introduction to computer algebra”. CATO will be explained in the first lecture, after the author

described what a computer algebra systems is and why it makes sense to use it. Students are able afterwards to practically get in touch by computer lab exercises. The tasks to solve are solving an equation, computing the length of a vector, drawing a Sine function, computing two probabilities for binomial distributed values. These small examples are enough to teach the general usage of CATO. Further usage of CATO is self-explanatory as always the same paradigms are followed.

The author uses CATO in connection with Maxima as Maxima is open-source and free to use. Students are able to apply CATO with Maxima at home without any additional costs. CATO is used at the University for Applied Sciences in Konstanz with Mathematica or Maxima by other lecturers as well. Mathematica is used in cases, when the computation power of Maxima is not large enough. (Maxima will reach its limit quite soon when computing eigenvalues and eigenvectors.)

8 First common errors when using CATO

What are the common errors when students use CATO for the first time?

- In cases of the multi-parameter commands are all parameters filled in the second input window, but the command is not executed by the students after finishing entering all parameters. This happens only the first two tries.
- Sometimes definitions are entered within other computations or commands. It seems as if the programming skills are missing in these cases, because definitions are typically independent commands.
- Some students use the German decimal separator (a comma instead of a point) to denote decimal numbers.
- Students may omit multiplication operators. For instance they should write $5 * (x + 7)$ and $3*x+2$ with the multiplication operator.

9 More than only acceptance of CATO

CATO connected with Maxima is accepted as an auxiliary tool from the beginning. In the meantime of the lecture, there are often questions regarding CATO by the students, resulting from independent self-studies with CATO (also by intermediate students). Shortly before the final exams, there are complaints about the too small number of commands offered by CATO or the sometimes cryptic results from Maxima. These complaints are also mentioned by weak students. In the exam, the weak students will also write down the input and output numbers of CATO without any deep understanding. There is a log available in CATO where all the previous computations and results can be looked up.

10 Changes to the Lecture

The author uses computer algebra more often than in previous times. He always brings his computer to the lectures for interactive usage of CATO. The students are always able to follow the application of CATO, and can enter the same commands on their own if desired. During lectures might be the need for additional examples which can be interactively designed with the help of CATO.

Using CATO requires the lecturer to ask other questions and deal with another kind of problems more related to understanding which technique to use when. Problems and examples built only on pure manual computation are replaced by understanding-focused problems. The written exam is split into two parts: The first part is without any additional tools such as calculators or computer algebra and basic skills are required. In the second part, the usage of CATO with Maxima is one of the allowed auxiliary tools.

11 Feedback changes CATO without changing the user` access

The author does not only use CATO during the lecture but allows the students to use it as well. The students may either have practical exercises in one of the computer labs or have their own computers with them. The immediate feedback seeing students use CATO helps to improve CATO every term. Besides oral feedback is always welcome for further improvements.

One of the changes resulting from students' feedback is the rephrased label for the computations of the probability density function or cumulative distribution function. We rephrased the label to $f(x)$ or $F(x)$ respectively. After entering the command, the student can therefore check whether the right command was chosen. (see fig. 13) This change seems simple, but only the abstraction between command label and entering the parameters makes this change possible.

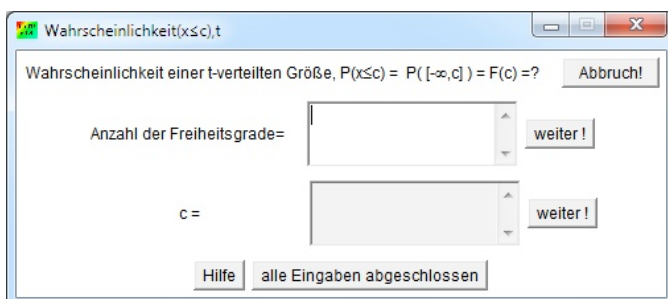


Fig. 13: The cumulative distribution function of the Student t distribution.

12 Feedback extends CATO, without changing the structure of CATO.

Other feedback included the wishes for a simple access of the recently used commands. Especially in cases, where commands of different packages have been chosen, this comes in handy. I developed two possibilities:

- There is the package “own packages” containing the sub-packages “last 20 commands”, “last 50 commands” and “last 100 commands”. These packages will include all the commands being recently used also including commands of last CATO sessions.
- There is the menu “Chronic” containing all commands of the current session.

These packages allow to choose commands as usual following the common usage paradigm.

Both approaches are dynamically updating and reflecting the user interactions with CATO. We could easily integrate these approaches, because of CATO’s concept to group commands into packages, and one command being represented in more than only one package.

13 Feedback integrated to enhance the user’s experience

Students asked for the possibility to create sets of commands grouping all the important commands for one written exam, export this grouping and import it again on any computer.

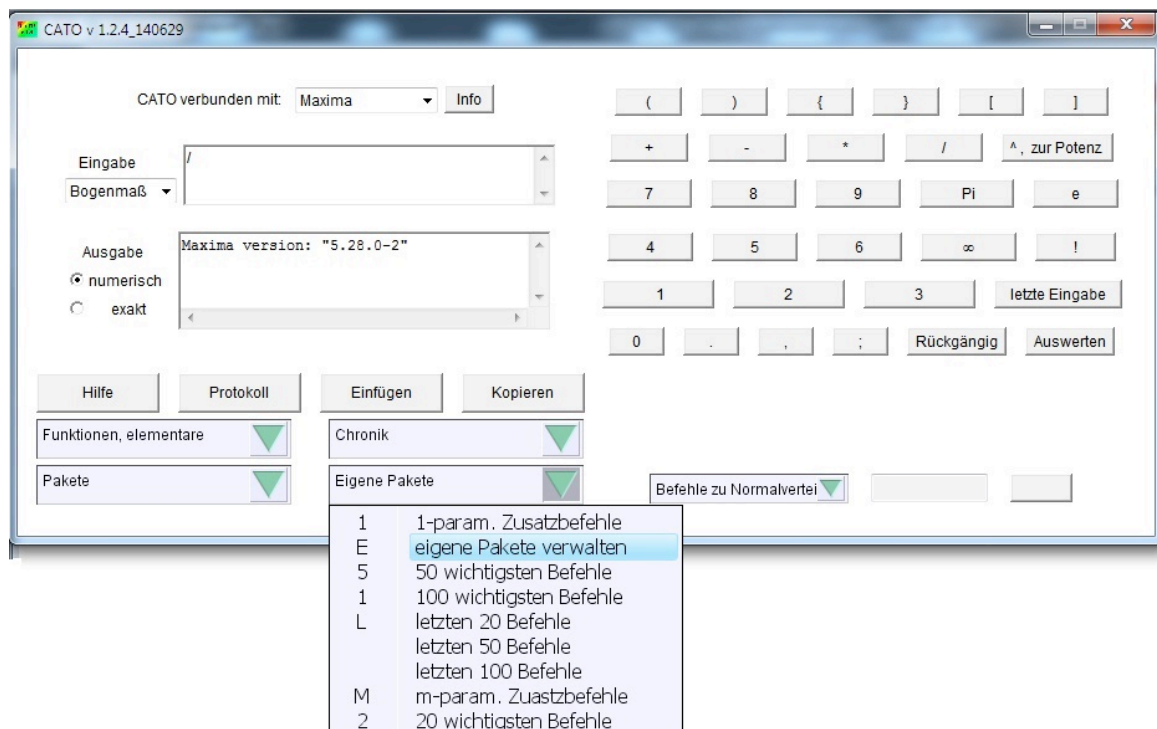


Fig. 14: Section of the sub sub-packages “manage own packages”.

Saving time is the main reason for this requirement. The author integrated this wish as follows:

- There is the possibility to save own packages with a user-given name. These packages can be imported at any time again into CATO. They are listed under the main category “packages” beneath the CATO-defined packages. The selection of these packages is supported in the usual way and the contained commands can be accessed in the same manner as CATO-defined commands. (Fig. 14)
- Furthermore, the package “own packages” contains the sub-package “manage own packages”.

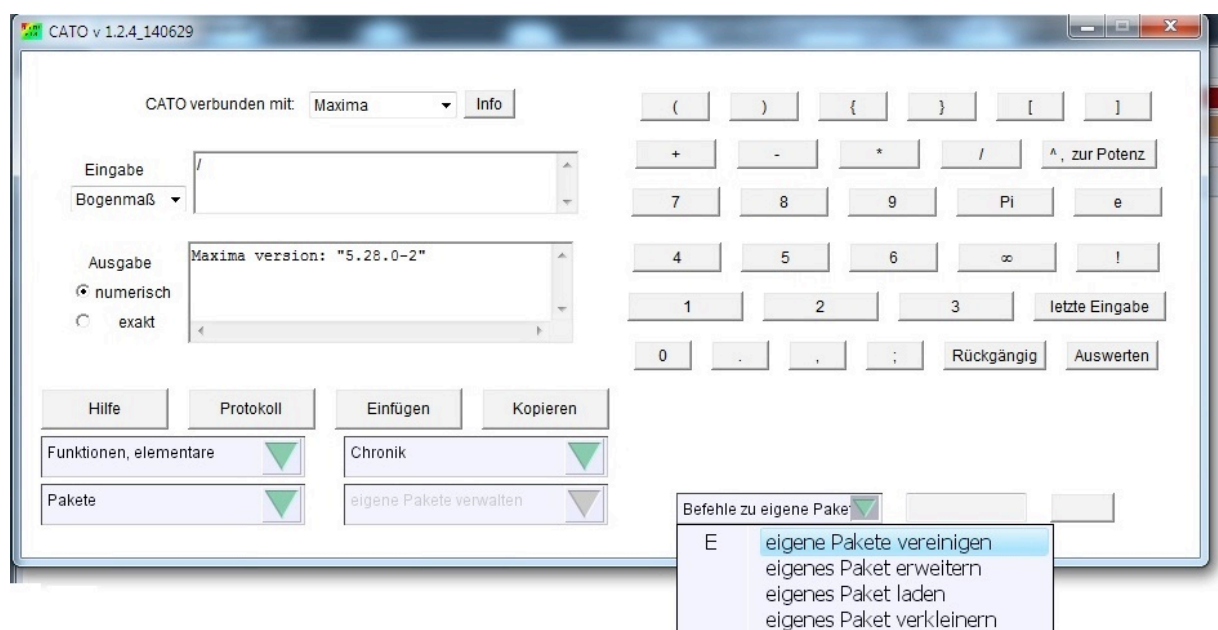


Fig. 15: commands of “mange own packages”

There is the possibility to shrink a self-defined package to exclude unnecessary commands (see fig. 15). A window will be shown, where all commands of this package are shown and marked. The student can uncheck the commands he wants to remove from the package and store the new selection (see fig. 16).

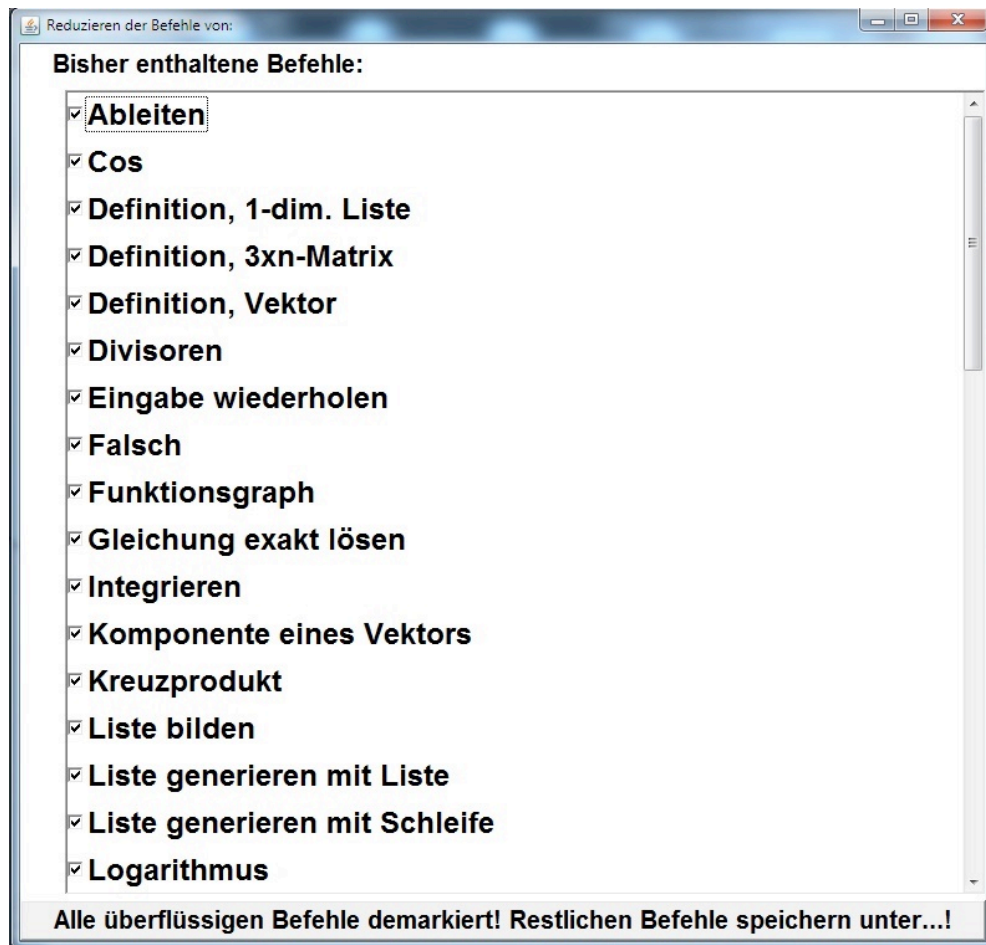


Fig. 16 : A window, where the student can uncheck commands

- Two self-defined packages can be combined to one package and can be stored. Commands included in both packages will be only added once to the merged package
- Own packages can be defined by the chronic and using all of the desired commands. The chronic can be saved as soon as it contains more than one command. Furthermore, the chronic can be added to any other user-defined package.

14 Current status of CATO

Currently there are approximately 500 commands for Maxima, approximately 500 commands for Mathematica (version 4.0 or higher), approximately 400 commands for the mathematical toolbox of MATLAB, approximately 300 commands for Maple (version 9.5 or higher), approximately 300 commands for MuPAD 3, approximately 200 commands for Yacas, and approximately 100 commands for MATLAB. Additional are some CATO internal commands.

The connection to MATLAB has the noteworthiness that the mathematical Toolbox is automatically selected whenever available. Furthermore, an external

application interface has to be installed to use MATLAB resulting in the requirement for version R 2009 or higher.

The code of CATO consists out of 20500 lines of Java excluding the command data base. CATO has been tested on Mac OS X and Linux/Ubuntu.

Free trial versions of CATO can be downloaded from the author's website at any time.

15 Conclusions

The graphical user interface CATO is a possibility to use CAS only sporadically. Computations and commands follow always the same paradigm. Previous concepts and paradigms postulated by related works have been integrated and implemented. The weekly usage of CATO has successfully shown that CATO is helpful during lectures. The consequent implementation of CATO's paradigms has been approved by students and is supporting students during mathematics lectures. The author fulfilled his goal to apply computer algebra during the lecture helping the students and not teaching the students computer algebra. The goal is the result using a tool, not how to use a tool.

This work is resulting from presentations the author held in the years 2013 and 2014 extended by the latest developments of CATO.

16 Literaturverzeichnis

Cojocaru, S., Malahova, L. and Colesnicov, A. (2006). Providing Modern Software Environments to Computer Algebra Systems. Lecture Notes in Computer Science, Vol 4194. Springer

Janetzko, H.-D. (2013). CATO - Eine deutschsprachige CA-Oberfläche. In ÖMG -DMV Congress 2013, Innsbruck, 23.-27.09.2013, Seite 166.

Janetzko, H.-D. (2014). CATO - beiläufiger, selbsterklärender Einsatz von Computeralgebra in Mathematikvorlesungen für Ingenieure. 48. Jahrestagung der Gesellschaft für Didaktik der Mathematik, Koblenz Landau, 10.-14.03.2014, Beiträge zum Mathematikunterricht 2014, Seite 567-570.

Janetzko, H.-D. (2014). CATO - ein einfacher Zugriff auf CA-Systeme, unter anderem die math. Toolbox von MATLAB. In MNU-Bundeskongress 2014, Kassel, 10.-14.04.2014, Seite 55.

Janetzko, H.-D. (2007 -), Die Hilfe zu CATO, (revised regularly) , www.computeralgebra.biz

Kajler, N. (1992). CAS/PI: a portable and extensible interface for computer algebra systems. In International Conference on Symbolic and Algebraic Computation (p. 376-386). ACM New York, NY, USA.

Kajler, N. (1993). Building a computer algebra environment by composition of collaborative tools. LECTURE NOTES IN COMPUTER SCIENCE, 1993, (p. 85-85)

Kajler, N. (1993). {User interfaces for symbolic computation: a case study. In Proceedings of the 6th annual ACM symposium on User interface software and technology (p. 1-10). ACM New York, NY, USA.

Kajler, N. (1998). Computer-human interaction in symbolic computation. Springer.

Kajler, N. and Soiffer, N. (1998). A survey of user interfaces for computer algebra systems. Journal of Symbolic Computation, 1998.(25), p. 127-159.

Oancea, C.E. and Watt, S.M. (2006). Domains and expressions: An interface between two approaches to computer algebra. In Proceedings of the 2005 international symposium on Symbolic and algebraic computation (p. 268). ACM.