

Friend or Foe? Examining CAS Use in Mathematics Research

Andrea Bunt, Michael Terry & Edward Lank
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
{abunt, mterry, lank}@cs.uwaterloo.ca

ABSTRACT

Computer Algebra Systems (CAS) provide sophisticated functionality to assist with mathematical problem solving. Despite their widespread adoption, however, little work in the HCI community has examined the extent to which these computational tools support domain experts. In this paper, we report findings from a qualitative study investigating the work practices and tools of nine mathematicians in a research setting. Counter to our expectations, our data suggests that computational tools play only a minor role in their workflow, with the limited use of CAS owing primarily to four factors: (1) *the need for transparency* in CAS's reasoning to explain computed results; (2) *the problem of rigidity and formality* in CAS's input/output style dialogue; (3) *the need for 2D input* to support a wide range of annotations, diagrams, and in-place manipulation of objects of interest; and (4) *the need for collaboration*, particularly in early stages of problem solving. While grounded in the study of mathematicians, these findings (particularly the first) have implications for the design of computational systems intended to support complex problem solving.

Author Keywords

mathematical problem solving, Computer Algebra Systems

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

INTRODUCTION

Computer Algebra Systems (CAS) provide sophisticated functionality to support symbolic manipulation of mathematical expressions. CAS figure prominently in disciplines such as computer science, physics, economics, and engineering, where hundreds of thousands of users make use of them on a day-to-day basis. For example, when Maple (a popular CAS) released a new version of its software in 2006, 795,000 licenses were activated within a two-month period [1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4 - 9, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.

Despite the popularity of these systems, little HCI research has examined these tools in *professional* contexts. Instead, existing research has typically studied this software in educational settings (e.g., [16], [3], [15], [14], [10]), or has focused on specific features in laboratory situations (e.g., the usability of mechanisms to input mathematical expressions [2], [7]). While this past research provides valuable perspectives on this class of software, there is still a need to examine these systems in the context of professional use, where the tools have the potential to impact cognitively demanding tasks in a significant way. Designing tools that augment human intellect remains a challenging endeavor, but must begin with a thorough understanding of the context of use. In this case, there is a need to understand how mathematicians work, as well as how current computational tools integrate with practices, to understand the specific HCI challenges for this class of software.

This paper reports the results of a qualitative study examining the practices, tools, and artifacts of nine expert mathematicians from a research university. We performed semi-structured interviews in the mathematicians' places of work and took photographs of their work environments and artifacts to help us better understand their overall goals, workflow, and use of tools and media. The participants represent various mathematical specialties, though all can be considered theoretical mathematicians. The interview data and work artifacts provide insight into the work practices and goals of this user group, as well as the roles computational tools play in their problem-solving practices.

Contrary to our expectations, participants' comments suggest that computational tools play only a minor role in our participants' work, despite their functionality. For our participants, the primary goal appears to be the creation of *new mathematical knowledge* communicated via a *mathematical narrative*. Work artifacts reveal that this mathematical narrative is composed of both text and mathematical constructs (e.g., figures and mathematical expressions) and simultaneously contains and *constitutes* the result of their work. For example, while the work may derive a new formula (what one might consider the final "result" of the work), it is the derivation of this formula and its proof of correctness that constitute the primary contributions of our participants' work. It seems that this need to both derive *and prove* new mathematical knowledge significantly affects our participants' perceived utility of current CAS software. In particular, the fol-

lowing themes emerged with respect to these tools:

- *The Need for Transparency*: Given the goal of discovering and proving new mathematical knowledge, there appears to be a concomitant need to understand every step of the process. However, the CAS software used by our participants lacks sufficient *transparency* in communicating its internal processes. Specifically, given input and a command, it produces an output, but without accompanying information explaining how that result was derived. For the purposes of proving the correctness of one's work, it is essential to understand this derivation, especially since CAS can employ sophisticated algorithms when operating on mathematical expressions. Interview data suggests that this lack of transparency has two consequences: It limits researchers' ability to deepen their understanding of the problem, and it contributes to a lack of trust in the system itself. For example, when an unexpected result is returned, the mathematician must question whether the system itself is correct, whether there was an input error, or whether he, himself, is in error.
- *The Problem of Rigidity and Formality*: While CAS could be used in various phases of the mathematical problem-solving process, we found that the software's imposed levels of formality and rigid input/output format can also deter its use, especially in earlier problem-solving phases where mathematicians tend to move fluidly between informal notes, sketches, formulae, and mathematical arguments. For example, terms need to be explicitly defined so the CAS knows how to manipulate them, introducing an overhead compared to the use of physical media.
- *The Need for Free-Form 2D Input*: We found mathematicians make liberal use of sketches, mathematical expressions, and annotations to render abstract mathematical concepts more concrete. In the context of performing mathematical work, all of these representational forms can be viewed as *dynamic* objects that change over time, for example, as terms in an expression are crossed out, content is added to sketches, and new insights lead to new annotations. Despite its importance for performing work in this domain, current mathematical tools provide limited support for this free-form use of 2D space.
- *The Need for Collaboration*: Mathematics is often a highly collaborative activity with individuals working around a shared whiteboard or piece of paper. Collaboration with these media is extremely fluid, whereas collaboration around a shared computer console requires coordination of physical input devices. These requirements reduce the feasibility of using CAS for collocated, collaborative work.

The results in this paper are a first look at CAS use in a specific type of professional mathematics: theoretical research in a university setting. While limited in scope to this particular user group, this work can form the basis for future study of other types of professional mathematics. Our findings also suggest implications for the design of tools intended for cognitively complex tasks outside of the domain of mathematics. For example, CAS are intended to perform manipulations of mathematical entities to offload these cognitively

demanding tasks to the computer. As we found, however, users do not wish to be completely independent of this process. Instead, they seek to *understand* the operations performed by the system to help them better comprehend the problem. Findings of this nature are likely to be equally applicable to the development of cognitive aids in other fields. We further expand on these implications later in the paper, particularly with respect to tools that assist with mathematical work.

The remainder of the paper is structured as followed. We contextualize this research by first considering previous work in the area of mathematical tools. We then describe our study design and detail its findings. From these findings, we derive a set of potential implications for the design of mathematical software for professionals. We conclude by discussing our study's limitations and directions for future research.

RELATED WORK

A wide range of computational systems, both commercial and experimental, have been developed to support mathematical work. For the purposes of this paper, we are most concerned with systems that emphasize *symbolic computation* capabilities, as opposed to data processing. Thus, we are most interested in Computer Algebra Systems such as Maple and Mathematica, as opposed to systems such as Matlab or S/R, which are most tuned to dealing with data. With this focus in mind, there are two general thrusts to HCI research studying the design of these systems: laboratory studies assessing these systems' general feature sets (including expression entry capabilities) and studies examining the impact of these systems in educational contexts.

In a laboratory evaluation of mathematical problem solving with high-school students as participants, Oviatt *et al.* studied the impact of different media on problem-solving performance. The experiment included four conditions: pen and paper, an Annoto pen, a pen-based tabletPC, and a graphical equation editor [13]. The authors found that problem-solving performance was better with pen and paper or the Annoto pen compared to the other two conditions. Drawing on Cognitive Load Theory (e.g., [19]), the authors attribute the results to the familiarity students have with entering and manipulating expressions with physical media, leading to comparatively higher cognitive loads when using digital media for these tasks.

A number of studies have focused on the problem of entering mathematical expressions. Anthony *et al.* compared pen-based entry to keyboard-and-mouse, speech, and pen plus speech. The authors found that expression entry with keyboard-and-mouse was significantly slower than the other three conditions and that pen-based entry was the most preferred [2]. A pair of studies have also considered expression entry in the context of *pen-math* systems: systems use a tablet PC as an interface to CAS software (or some type of mathematical backend) with the goal of creating a more natural input interface. LaViola [8] and Labahn *et al.* [7] both assessed the user's ability to correctly enter expressions and solve a number of small problems. The evaluations showed

that while expression recognition can be challenging in such systems, users were able to complete their tasks effectively once their expressions had been recognized. These studies again suggest that pen-based input has particular advantages, but that digital systems do not offer clear-cut advantages with current interfaces and recognition engines.

Moving outside of the laboratory setting, various research efforts have investigated how CAS integrate with high-school and undergraduate education (e.g., [3], [10], [14], [15], [16]). These studies have uncovered a number of advantages to using CAS in the classroom. For example, one study found that students are able to experiment with different expressions more easily, which can promote a higher-level understanding of relevant concepts [3]. In addition, by delegating some of the work to the CAS, students are more able to focus on problem-solving processes as opposed to focusing solely on calculation details [10]. This line of research, however, has also shown that integrating CAS into the classroom requires careful lesson planning [10] and teacher support [14], and that some students have difficulty translating CAS output into representations that they understand [3]. Finally, some students feel that they learn more when solving problems by hand or that “real mathematics” is done by hand, not by computers [15].

This initial set of studies, both in the laboratory and in educational settings, provides important insights into the potential benefits and limitations of current CAS software. For example, expression input can be singled out as one area in need of further improvements. However, these studies characterize only *short-term* use of such software, typically in fairly well-defined, well-directed ways (e.g., in an experimental study or in the classroom with well-defined task). To the best of our knowledge, no study has considered how these tools are utilized by professional mathematicians in authentic work situations where problems are ill-defined in nature. An understanding of how these tools are adopted and applied in professional environments would be valuable, both to guide future design and to identify open research problems. Given the cognitively demanding nature of mathematics [6], it is likely that the insights gained from studying the role (and potential role) of computation in mathematics will be applicable to other cognitively demanding tasks.

STUDY

Method

To understand the current work practices of mathematicians in a professional setting, we conducted a series of interviews with nine mathematics researchers (eight males, one female) at a research university. All participants have advanced mathematics knowledge, with all but one participant at the PhD level or higher. The title and research area of each participant is listed in table 1.

Interviews took place at locations identified by participants as their primary workspace (either offices or labs), enabling us to photograph their work environments. Conducting the interviews in their workplace also allowed us to view, discuss, and document samples of relevant work materials. The

ID	Title	Research Area
P1	Postdoc	Theoretical Computer Science
P2	PhD student	Quantum Computing
P3	Faculty	Applied Math
P4	MSc student	Pure Math
P5	Postdoc	Symbolic Computation
P6	Postdoc	Mechanical Engineering
P7	PhD student	Pure Math
P8	Faculty	Applied Math
P9	Faculty	Theoretical Computer Science

Table 1. The backgrounds of our study participants.

interviews were semi-structured and lasted approximately 30-45 minutes each. During the interviews, we asked participants to educate us about their research practices and how they perform their work. To ground the interviews and assist with recall, we asked participants to walk us through specific instances of recent research work.

We collected data by audio taping the interviews and taking digital photographs, with two exceptions. The audio recording device failed during one interview; immediately following this interview, the interviewer created detailed notes and later had the participant review the notes for accuracy. A second participant declined to have photographs of his/her working materials taken for reasons of privacy.

Interview Questions and Data Analysis

In our interviews, we sought to answer the following questions:

- What is the goal of the mathematicians? What are they seeking to accomplish? What is the “product” of their work?
- What characterizes the mathematicians’ workflow? That is, how do they accomplish their work?
- Which tools are used in mathematical problem solving (e.g., paper, whiteboard, LaTeX, CAS), at which points in the work process, and for what reasons?
- What types of tasks are best supported by the different tools and why?
- What preferences do they have with respect to tools and media?

Data were analyzed by creating two separate affinity diagrams of participants’ responses and their work artifacts. These affinity diagrams revealed common themes in work practices and goals, as well as common conventions within the artifacts themselves. In analyzing the data, we also discovered a strong temporal theme within the artifacts. In particular, we found that the samples collected served to document the progression of mathematical solutions from early problem-solving stages to final solution forms. Thus, in addition to the affinity diagrams, we developed a timeline composed of pictures of the artifacts. The timeline incorporated samples from all participants, which had the benefit of providing multiple example artifacts from similar points in the problem-solving process. From this timeline, we were

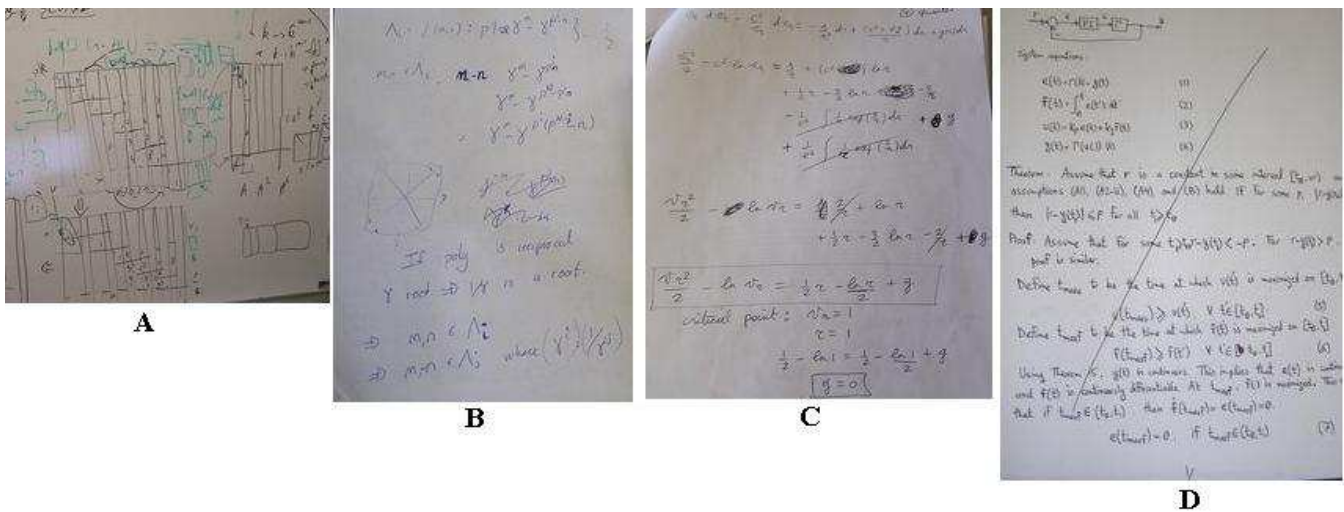


Figure 1. Example images illustrating a subset of the timeline present in mathematicians’ work. (A) is example of early work on one participant’s whiteboard and (B) is another participant’s work on paper. In the early stages, expressions and diagrams are rough, with little attention paid to alignment or formal prose. As the work progresses (C) shows that there is increased structure, with greater attention paid to alignment. In (C), however, items are still being actively manipulated as the derivation unfolds. Finally in (D), we see an example where the narrative has reached a more formal state – the writing is neat, the structure is clean, and rhetorical conventions are used in a more rigorous fashion.

able to identify general trends in how solutions are developed over time. Figure 1 shows a subset of this timeline.

FINDINGS

To understand the role of computational tools in the mathematical process, it is essential to first understand the desired outcome of the mathematicians’ work. Accordingly, we frame our findings by considering the primary goal of mathematicians, how they achieve that goal, and the roles computational tools play in the process.

Mathematicians’ Goal: Mathematical Narratives

Based on comments made during the interviews and analysis of the work artifacts, the primary goal of our subjects’ work appears to develop *new mathematical knowledge*, rather than to apply mathematics to specific problems (such as computing a result given a set of data). This goal of constructing new knowledge has important consequences for the role of computation in the work process.

For our participants, the specific output of their work is a formalized *mathematical narrative*. The purpose of the narrative is to describe the *transformation* of mathematical entities from an initial form to another, more desirable form, and to prove the correctness of this transformation. The narrative itself is a mixture of prose, mathematical expressions, and graphs and diagrams of the mathematical phenomena under study (e.g., Figure 1(D)). It is a highly structured document following established conventions in rhetorical style and the visual presentation of mathematical material.

The work artifacts indicate that the narrative serves two purposes: It communicates the mathematical phenomena to others, but just as importantly, it argues for the correctness of the work. Thus, while the end result may be the derivation of a new mathematical formula, it is the *description* of the deriva-

tion and the *argumentation* for its correctness that form the primary contributions of the work.¹ Consequently, the mathematical narrative simultaneously *contains* and *constitutes* the results of the mathematician’s work.

Developing the Mathematical Narrative

We found that the mathematical narrative gradually evolves through an iterative process that shares a number of similarities with design practices, such as those described by Schon in the *Reflective Practitioner* [17]. P6 provides a cursory summarization of this process:

Okay, this is how I work. First of all, I think about the problem. I draw some meaningless figures like this [artifact] and then I translate what I see to some equations. Then I write my equations down [in a way] that is readable by someone else, like this [artifact]. [...] And then I type it and then I submit it. (P6)

We expand on this process, noting the roles computation plays along the way.

In the early stages, the problem is represented using rough sketches, basic diagrams, and the informal use of text, mathematical formulae, and rudimentary mathematical argumentation. Figure 1(A) shows one mathematician’s whiteboard at an early stage, while Figure 1(B) shows early work on a piece of paper. While one can observe some of the elements and conventions eventually used in a final narrative (e.g., the use of whitespace, indentation, and labels to visually structure the document), these initial representations are intended for the mathematician himself, rather than a third party.

¹Note that while we use “formula” here as an example, the role of the narrative is the same for other types of mathematical work, such as proving a mathematical relationship or concept.

Work progresses through continual manipulation of these mathematical entities on the physical media. Figure 1(C) and 2 show snapshots of this early work, embellished with notes, content crossed out, and manipulation of expressions into other forms. Gradually, these representations become more formalized and structured, and the narrative begins to form. This gradual formalization not only helps prepare the document for eventual presentation to a third party, it also serves as a problem-solving tool itself. More specifically, as one moves to more formal representations and argumentation, the problem is subjected to increasing levels of mathematical rigor, which can uncover flaws not obvious with earlier, rough, higher-level representational forms. For example, Figure 1(D) shows a solution at an advanced stage where the presentation has achieved a fairly high degree of formalization. However, the “slash” through the page here indicates that the solution was found to be incorrect at this late stage.²

Summarizing the Phases of Mathematical Work

From this description of our subjects’ work practices, it appears as though there are a number of phases to mathematical problem solving in this type of professional setting. To facilitate the discussion of computational tools in the upcoming section, we label these phases as follows:

- *Ideation*: A brainstorming phase where ideas are generated.
- *Execution*: Ideas are carried out by solving, deriving, and constructing mathematical proofs.
- *Formalization*: The results of the previous two phases are refined such that the work becomes a more complete mathematical narrative.
- *Dissemination*: The work is prepared such that it can be presented to others, either via publication or a more formal presentation to a supervisor.

Computational Tools for Mathematics

Contrary to our expectations, our participants reported only limited use of computers during problem solving. Instead, they indicated making heavy use of physical media, particularly paper and whiteboards (or blackboards).³ As the work becomes more formalized, it may be transferred to a computer, particularly when the work is being prepared for publication, but prior to this latter stage, there do not appear to be strong incentives for our subjects to switch to computer-based representations. There seem to be two primary reasons for this limited use: Physical media easily accommodate the various mathematical representations (e.g., expressions, diagrams) and their manipulation; and collaboration is more easily accomplished using physical media.

When they are used, computational tools are employed to communicate results formally with typesetting software, and

²The slash could also represent the fact that the material is no longer needed, but in this case, it represents a dead-end.

³Some participants use blackboards rather than whiteboards, but in terms of the findings presented here, the two are equivalent.

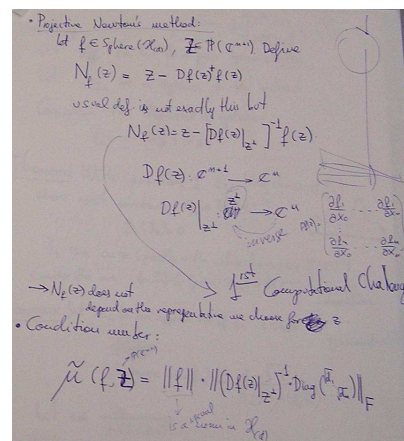


Figure 2. Another example of early work using pen and paper. Of note are the use of annotations at an angle to distinguish them from the primary narrative and short-hand notation for matrix definition.

to support symbolic manipulation. We expand on these uses then consider the shortcomings of current offerings.

Formatting and Formal Presentation of Results

Typesetting software, specifically LaTeX, is frequently used by our subjects to formalize the work and communicate it to others (i.e., the *Formalization* and *Dissemination* phases). Our participants are quite comfortable with LaTeX’s syntax, feeling that it was well worth the time to learn given that LaTeX is seen as the “Gold standard” for typesetting mathematics because of the output that it produces.

LaTeX is also used by some participants to help organize their work as it evolves over time. For example, P1 showed us a LaTeX-based scheme for this type of project management. The document resembled a draft of a paper, but also contained notes on things that need further exploration, and ideas and formulations that did not end up in the main narrative (but were stored in an appendix).

The Use of Computer Algebra Systems (CAS)

The other major tool used by our participants are Computer Algebra Systems, particularly Maple. In our study, we found participants use CAS software to solve certain types of complex expressions, to verify hand-derived work, and occasionally in the *Ideation* phase to explore a range of possible solutions. We describe each of these uses.

As we have noted, participants said that they tend to do the majority of their work using physical media, such as pen and paper or a whiteboard. When used, they feel that their primary use of CAS is to deal with an expression that they find either too long, or “tedious” to compute by hand:

Usually if it is a complicated expression that I can’t resolve myself. [...] the kind of tedious work that is sort of boring and uninteresting but where it is easy to make mistakes. (P1)

If I have some horrible expression that I don’t like,

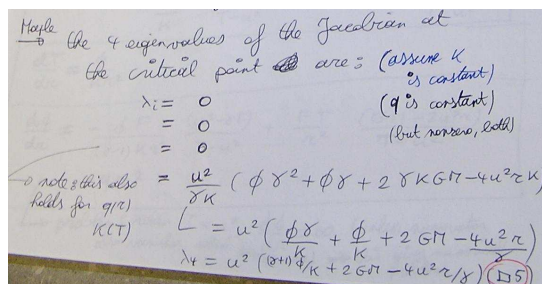


Figure 3. A portion of a participant’s pen and paper work where s/he has used Maple and noted so directly within the narrative under construction.

some large amount of tedious computation, integrate this or reduce this giant mess to something useful, then sometimes I’ll stick it in Maple to see if it can solve the problem for me. (P3)

When solving these types of expressions, participants said that they use paper to formulate the expressions, use the CAS to solve/simplify the expression, then transfer the result back to their paper work. Figure 3 illustrates an instance where a participant has engaged in such a process and noted the use of Maple directly in the work. Noting the use of Maple in the narrative under construction can serve multiple purposes. It could simply act as a reminder of the path taken to arrive at the solution. It could also, however, relate to the trust issues that we discuss in the next section. Since some users have difficulty trusting the CAS, some may feel it important to note any reliance on its results directly in the narrative.

Participants also move between physical media and the computer for a second popular use case, namely to verify hand-derived work. In these instances, participants work out the details by hand and use the CAS to increase their confidence in their solutions.

Some of our participants indicated using Maple in the *Ideation* phase to experiment with the output of a number of similar expressions. This use case, however, was not frequently reported among our participants. One participant who was more willing to use a CAS than most, commented on the tradeoff between doing this type of work by hand or with a CAS. Altering the form of an expression is easy to do by hand, but performing multiple evaluations without a CAS takes a significant amount of time:

It is easy to do different things [in Maple] [...] If I had to solve all of that by hand... I mean I can put a 1 over 2PI [on paper], but if I want to know what that solves to... (P1)

Similarly, another participant showed us how he writes code to have Maple generate multiple plots and then examines these plots visually to see if he can detect patterns.

Finally, Maple occasionally is also used for sophisticated searching. Rather than using Maple as a interactive problem solver, these participants write code to have Maple search

through a space of solutions for a counterexample that violates one or more mathematical properties.

It’s a matter of just testing all possible solutions to see if they are solutions or not. And the algorithms are really the fastest way I can test that. (P2)

Reflecting on these uses of Maple, we note that participants feel that they are most likely to delegate work that they find uninteresting or inelegant to the computer. The rest of the work they feel is best done by hand, even if the computer could do that same work much faster.

The fact that, for the most part, CAS tools play such a minor role in the problem-solving process is worthy of further consideration, especially since one would assume that the ability to manipulate complex mathematical expressions computationally would be of great use to the mathematicians. In the next section, we describe open issues in the design of CAS.

OPEN ISSUES IN THE DESIGN OF MATHEMATICAL SOFTWARE

In analyzing current CAS, we found that they are applied to only a small subset of problems encountered when performing mathematical work. Furthermore, we found that features of their current design can work against the processes vital to constructing this narrative. The specific issues discovered can be summarized as:

- A need for transparency in the problem-solving process
- A need for free-form 2D representations
- A challenge in transcribing representations from physical to digital representations
- A need for collaboration

We describe each of these issues in turn.

The Need for Insight and Transparency

Through our interviews, we discovered that even when performing individual calculations, our participants seek more than just an answer from the system; they also seek to further their understanding of the problem under study. Our participants claimed that they are able to gain better insight and can better detect patterns when solving problems by hand than by using a CAS. Furthermore, some indicated that working out the details keeps their mathematics skills sharp. The following quotes from participants illustrate these perceived benefits of manually solving problems:

Sometimes [...] it is a good exercise for me to try to do it as much by hand as possible because then I exercise certain parts of my grade 12 calculus class and keep those fresh. (P1)

Computers are great for running through large amounts of examples, but you don’t get the same insights. Whereas if you did something by hand, sometimes you just get more insight and can figure out the general pattern. (P2)

You can notice patterns better if you've done it yourself rather than just the way Maple has grouped it. (P9)

Along these same lines, the interviews also reveal that transparency is an issue with such systems and that many participants have difficulty trusting the computer-generated results:

Sometimes the software package comes back with something even more horrible than you expected and it is hard to translate that back to something you understand. (P2)

Sometimes the computer algebra, it skips steps, or you can't see, or in the end you have to go back... (P9)

I tend to not trust the results from the symbolic toolbox [...] Although it is very infrequent that the results are incorrect. (P6)

Whenever you do something in Maple, you'd like to be able to re-produce it by hand. (P1)

I don't have a good understanding of what kinds of things [Maple] trips on. (P1)

It is important to keep in mind that these are quotes from experienced mathematicians, not those learning the basics of the problem domain. Given these results, it appears that current CAS technology is most suited to situations where an individual seeks only an answer, not the process that created that answer. Our participants feel that transparency is key; without this transparency, they do not feel able to follow the CAS's reasoning in a way that gives them insight into why and how a particular answer was achieved. For our participants, this leads to difficulty trusting CAS output, which in turn creates a desire to reproduce the results by hand. Ultimately, this increases the time spent solving the problem.

The Need for Free-Form 2D Representational Forms

Throughout problem solving, our participants make use of symbols, diagrams, prose, and physical space itself to represent and manipulate the problem. In some cases, it is possible to see the progression of the solution from one form to another through the symbols and prose on paper, particularly when expressions are written sequentially in a top-down fashion (e.g., Figure 1(C)). In other cases, the flow of the work is less obvious, as in the cases of diagrams, which may include many in-place modifications as the understanding of the problem evolves. However, in all cases, it is important to note the rather *direct* interaction with the objects of interest – items are annotated, embellished, edited, and crossed-out in place. This in-place interaction is an important feature of the problem-solving process, as it documents not only the process of transforming the initial state into the more desirable end state, but also the approaches that *don't* work. P7 comments on this work process:

And I don't even necessarily work down the page. [...] I just sort of have everything all in one spot. Obviously it's not very neat or easy to deal with, but just having

everything on one page kind of makes a big difference [...] I think it's easy having everything all in one spot. It just stops me from forgetting anything. (P7)

Participant P2 also speaks to this iterative process, and the benefits of using physical media to support the work:

So it is sort of an iterative process. [...] So at first you figure out how you might approach a problem. You try it and it either works or it doesn't. [...] I think this [paper artifact] went through a couple more refinements before it turned into an actual argument. (P2)

Physical space is also used as a tool via the spatial arrangements of content within a single document, by using large surfaces such as tables to lay out multiple sheets of paper to obtain an overview of the entire problem, and by grouping related papers in folders and special notebooks.

In contrast to the free-form, unstructured nature of paper and physical space, commonly available computational tools, including CAS and LaTeX, enforce a highly linear, rigid structure. In fact, one participant commented that he tends to avoid LaTeX until as late as possible (i.e., until the *Dissemination* phase) due to this enforced linearity.

Current systems also have limited means by which one can choose an appropriate level of formalism when representing content computationally. For example, using a CAS to manipulate expressions requires all terms to be formally defined and short-hand notation cannot be used (as is done for the matrix in Figure 2). Tools such as CAS and LaTeX impart their own level of formalism which cannot readily be adjusted to suit the current problem-solving context.

Transcription Problems

Transcribing information in physical media to computational forms can also pose challenges. This problem is particularly relevant to transcribing equations into a form that allows the CAS to manipulate them. If done imperfectly, unexpected results can be attributed to two causes: a transcription problem or a potential error in the system itself:

I'll type in an expression, I'll have spent an hour trying to figure out what it means and what the results are, and then I realize I've made an error typing. (P1)

The only concern is that sometimes you end up having too many brackets. Although [Matlab] has [parentheses matching] I still find it sometimes tricky and it is very easy to make mistakes, stupid mistakes. (P6)

Part of the transcription problem is obviously owing to the reduction in dimensionality, where one must reduce a two-dimensional expression into one-dimensional representation inputted via a keyboard. Another problem, however, is the inability for the system to perform sophisticated error-checking on the input. In contrast to natural language, where tools such as spell checkers and grammar checkers can help detect errors, error detection in mathematical input is primarily

Phase	Tools
Ideation	primarily paper/whiteboard, small amount of Maple
Execution	primarily paper/whiteboard, some Maple
Formalization	paper or latex
Dissemination	latex

Table 2. Summarizing tool use according to phase of work.

limited to rudimentary syntax checking, such as checking for missing parentheses. The mathematician himself may have difficulty in performing error-checking, since he is still becoming familiar with the problem and thus less tuned to what the expression “should” look like.

When considering transcription problems, one can ask whether these errors are due to the input system itself, or are simply the result of human error that would occur in other circumstances as well. In our interviews, there were comments indicating that current systems’ expression syntax overhead is prohibitive, or that expression entry can be error-prone or unnatural when using a CAS. We were surprised, however, at the rarity of these types of comments. For example, with LaTeX, which has similar syntax requirements to a CAS, most participants indicated that expression entry was not an issue. They had, however, initially invested significant effort to learn the syntax of the system. One participant also commented that his/her work requires a fairly restricted sets of symbols, lessening the need to learn a wide range of syntax: “I hardly look anything up because I use similar symbols all of the time so it’s pretty fast” (P9). They also make use of macros to make expression input easier.

I do find that writing in long and complicated things becomes very painful unless you start using macros [...] You very quickly come up with a set of macros that you are constantly using and including. (P4)

Thus, overall perceptions concerning equation inputting are mixed. Some find the syntax to be problematic, while others feel that they have either mastered their required syntax or have developed other coping strategies. These results suggest that with a high enough perceived benefit, expert users are willing to invest the time to learn the syntax for expression entry or develop strategies to simplify the process.

The Need to Collaborate

Our participants revealed that mathematical problem solving, particularly during *Ideation*, is often a highly collaborative activity. While one participant indicated using paper to collaborate, a whiteboard’s large surface and opportunity for multiple input make it the primary medium for such tasks. Despite the importance of this activity, current CAS software offers no real support for collaborative work.

Summarizing the Current Role of Computational Tools

The final product of our mathematicians’ work is not a single number nor a single, derived formula. Instead, it is the demonstration and proof of a new mathematical concept, contained within a mathematical narrative that evolves over time. As summarized in Table 2, participants comments suggest that the majority of this work is performed using physi-

cal media, which affords direct interaction with multiple representation forms. Certain types of work can be performed with a CAS that would be difficult, if not impossible to do by hand, such as an exhaustive search through a large solution space for a solution that violates certain mathematical properties. Overall, however, current CAS appear to support only a small segment of type mathematics. As a consequence, there seems to be a high cost to using CAS software for our participants: They must transfer their work to this other medium, which can unintentionally introduce errors, and there is an inherent distrust in the results it produces because of a lack of transparency. Thus, there are numerous costs to using CAS software for this user group, with few clear-cut advantages.

DISCUSSION

Prior to discussing the implications for design that arise from our findings, we first compare our findings to those from educational research. We end the section with discussion of limitations of our study.

Comparison to CAS Use in Educational Settings

As we mentioned earlier, prior human-centered CAS research has focused mainly on educational settings. One of the primary differences that we observed in our study pertains to a CAS’s ability handle routine calculations, allowing the user to focus on higher-level principles. Unlike in educational settings (e.g., [3]), this sentiment was not expressed by our participants, who instead feel that they use CAS primarily for work that they are unable to do by hand. Educational research has also found that experimenting with different concepts promotes a higher-level understanding. This type of usage was not frequently mentioned by our participants, who feel that they use CAS for more targeted purposes, such as simplifying an individual expression, than for exploration.

Despite the differing levels of mathematical experience, we did observe some similarities to students’ perceptions of CAS in educational settings. First, some students feel they are able to learn more by doing the work by hand [15]. We were surprised to such intellectual advantages expressed by our expert mathematicians, since these users have mastered the basic concepts. Despite their high levels of expertise, our participants still feel that they gain more new insight into the problem domain by solving expressions by hand or that they are able to keep their existing skill sharp. Second, when describing the type of work that they do using a CAS, our participants often used negative adjectives – adjectives that were not used to describe their pen and paper work. The notion that “real mathematics” is done by hand not by computers was expressed by some students in educational settings [15]. The negative language used to describe the type of work best delegated to a CAS suggests that our expert mathematicians might share this perception.

Implications for Design

In the section “Open Issues in the Design of Mathematical Software”, we highlighted ways in which current computational tools fail to support our expert mathematicians’ work practices. In this section we discuss potential avenues for

improvement focusing on three main areas: 1) narrative construction, 2) reasoning transparency, and 3) collaboration. We also discuss ways in which existing systems, including research prototypes, have begun to address these concerns.

Support for Narrative Construction

To better support narrative construction, there is a need to decrease the amount of enforced structure imposed by the environment and a corresponding need to increase the range of annotations that are possible. As we saw from the work artifacts, desirable annotations include freehand diagramming, crossing out items while keeping the original content, multiple colours, and aligning, tagging and cross-referencing parts of the document.

Existing commercial CAS provide only very limited support for this type of narrative creation. Mathematica does have the ability to combine prose, diagrams and input/output to/from the CAS backend. The interaction, however, is still highly structured, and likely would not be suitable for the early stages of problem solving. Certain pen-based research prototypes, in addition to allowing more natural input, have begun to address some of the features necessary. For instance, the designers of MathPad² have investigated integrating free-form diagrams with text [9], while the designers of MathBrush have begun to address the issue of short-hand notation in early problem solving [18]. Also of note is MathJournal, whose flexible environment supports a wide range of annotations (e.g., free-form diagrams, different colours, alignment) [12]. MathJournal's backend, however, is not as powerful as a CAS and, therefore, would not likely be sophisticated enough for our participants.

In general, pen-based input appears to be a promising direction for narrative construction, particularly given its ability to support free-hand diagramming and other free-form annotations, and to alleviate the need for transcription. With pen-based input, one could use either an Annoto pen and paper or a tablet PC. Given some participants' desire to use physical space as a tool, such as grouping related documents or arranging them on a large surface, the tablet PC might not suit all users' needs. Instead, something similar to Paper Augmented Digital Documents (PADD) [5], might be more appropriate. With PADD, users could continue to use paper, but also have access to computational power by docking the pen when CAS functionality is needed. At this point, the user could enter into a dialogue with the system to define terms more formally, if necessary, and select the appropriate manipulations. Such a dialogue would allow users to interact with expressions in a free-form fashion, formalizing this content only when the services of the CAS are required.

A downside of pen-base input is the recognition process, which becomes increasingly error prone as expression length and complexity increases [9]. Unfortunately, this characterizes the type of expressions that our expert mathematicians are most likely to turn to a CAS for help in solving. An encouraging finding from this study, however, concerns the high acceptance and heavy use of LaTeX, despite the relatively large amount of syntax that users must cope with.

This finding indicates that expert users might be willing to use syntax and/or macros to clarify their intentions in a pen-based environment to ease the recognition burden. How to combine pen-based input with some syntax for expert users is a promising avenue of future work.

Increasing Transparency

When users enter an expression into a CAS, currently the response they receive consists only of the answer. We found in our study that this amount of information is not always sufficient to provide insight into the problem space and often leads to issues with trust and predictability. Consequently, a promising area of future work in designing such systems is increasing the level of transparency in the system's underlying reasoning. It may be that such information is already present in the system's reasoning process and the challenge then becomes determining how to present the user with the information in a comprehensible and meaningful format. The problem becomes more difficult if the underlying algorithms have to be revisited to enable this type of feedback. At the same time, there is a question as to how much commercial software companies wish to reveal their underlying algorithms, given that they may be considered intellectual property and a part of their competitive advantage.

While no support for transparency is included within Maple (the CAS used by our participants), this is not the case for all commercially available CAS. Mathematica, for instance, does provide some transparency within their student edition, showing the steps of certain derivations [11]. Similarly, Derive can display the steps of a simplification along with the rules used in the transformation [4]. Our findings motivate continued efforts towards the design of such mechanisms for both learning basic mathematical principles and developing new mathematical insight.

Support for Collaboration

We note that many of our participants collaborate with others, particularly during early phases of the work. For this collaboration to be possible with computational tools, there is a need for large surfaces that allow multiple, simultaneous input. Existing commercially available systems such as SmartBoards, already go a long way towards providing this hardware interface. However, there is still the need to transform the CAS software into a form that allows for the possibility of simultaneous interaction by users.

Study Limitations

There are two aspects of our study design that could impact the generalizability and validity of our findings. First, we interviewed only a certain type of professional mathematician: one conducting theoretical mathematics research in a university setting. It is possible that the work practices and attitudes of our participants are not representative of all expert mathematicians. For example, in more applied fields or in industrial settings, users might be less focused on narrative construction and more interested in obtaining answers to individual questions. As such, the need for free-form 2D input with in place manipulation might be lessened for these users and as a result, there might be less overhead associated

with using a CAS. Those in disciplines outside of theoretical mathematics might also be less confident in their own mathematical skills and/or less interested in keeping these skills sharp, also providing greater incentive for CAS use.

A second limitation of our study is the self-reported nature of the interview data. We note that in many cases our findings are based on both interview data and work artifacts, where the work artifacts provide evidence that our participants' statements are reflective of their actual work practices. In particular, the artifacts serve to support the existence of different phases of mathematical work and illustrate the extent to which participants rely on free-form 2D representations with physical media. Other findings, however, rely solely on these self-reports, such as frequency of and reasons for CAS use. In-situ observations and/or experience sampling would be required to validate our findings, especially those based solely on the interview data.

SUMMARY AND FUTURE WORK

Whereas most prior work on computational support for mathematics has focused on novice users still learning basic mathematics techniques, this research examined the work practices of mathematical researchers whose contributions lie in mathematics and closely related disciplines. Through interviews and an analysis of work artifacts, we found that the goal of these individuals is to build a complete mathematical narrative that communicates and proves a new mathematical concept. Computational tools are used to varying extents at isolated points throughout the work process, however, they are failing to support the process as a whole. Key areas that are problematic for these users include the inability to create rich annotations, the inability to support multiple levels of formality, and a lack of sufficient transparency to allow the researcher to develop insight into the nature of the problem.

In addition to exploring the design recommendations discussed in the previous section, there are a number of avenues to explore in the future in terms of when and why computational tools are best suited for mathematical problem solving. As discussed in the previous section, studying users in other disciplines and in industry is necessary to understand whether the themes uncovered through our work are present in other professional settings. It would also be interesting to deploy some of the pen-math systems, such as MathBrush or MathPad², to see whether the increased flexibility they provide is better able to support user needs than the tools that are in widespread use.

REFERENCES

1. Adoption of Maple 10 into Academic Market Exceeds All Expectations. www.marketwire.com/press-release/Maplesoft-563405.html.
2. L. Anthony, J. Yang, and K. Koedinger. Evaluation of multimodal input for entering mathematical equations on the computer. In *Proc. of CHI*, pages 1184–1187, 2005.
3. M. Artigue. Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. *Int. J. of Computers for Mathematical Learning*, 7:245–274, 2002.
4. Derive. www.chartwellyorke.com/derive.html.
5. F. Guimbretiere. Paper augmented digital documents. In *Proc. of UIST*, pages 51–60, 2003.
6. J. Hadamard. *The psychology of invention in the mathematical field*. Dover Publications, Inc., New York, N.Y., 1945.
7. G. Labahn, E. Lank, M. Marzouk, A. Bunt, S. MacLean, and D. Tausky. MathBrush: A case study for pen-based interactive mathematics. In *Proc. of SBIM*, 2008.
8. J. LaViola. An initial evaluation of MathPad²: A tool for creating dynamic mathematical illustrations. *Computers and Graphics*, 31:540–553, 2007.
9. J. LaViola and R. R. Zeleznik. MathPad²: A system for the creation and exploration of mathematical sketches. In *Proc. of SIGGRAPH*, pages 432–440, 2004.
10. C. Leinback, D. Pountney, and T. Etschells. Appropriate use of a CAS in the teaching and learning of mathematics. *Int. J. of Mathematical Education in Science and Technology*, 33(1):1–14, 2002.
11. Mathematica for Students. www.wolfram.com/products/student/mathforstudents.
12. Mathjournal. www.xthink.com/MathJournal.html.
13. S. Oviatt, A. Arther, and J. Cohen. Quiet interfaces that help students think. In *Proc. of UIST*, pages 191–200, 2006.
14. R. Pierce, S. Herbert, and J. Giri. CAS: Student engagement requires unambiguous advantages. In *Proc. of the 27th Annual Conf. of the Mathematics Education Research Group of Australasia*, pages 462–469, 2004.
15. R. Pierce and K. Stacey. Observations on students' responses to learning in a CAS environment. *Mathematics Education Research Journal*, 13(1):28–46, 2001.
16. K. Ruthven. Instrumenting mathematical activity: Reflections on key studies of the educational use of computer algebra systems. *Int. J. of Computers for Mathematical Learning*, 7:275–291, 2002.
17. D. Schon. *The reflective practitioner: How professional think in action*. 1983.
18. D. Tausky, G. Labahn, E. Lank, and M. Marzouk. Managing ambiguity in mathematical matrices. In *Proc. of SBIM*, 2007.
19. J. vanMerriënboer and J. Sweller. Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review*, 17(2):147–177, 2005.