



ISSUES IN COMBINING THE USE OF VARIOUS TOOLS IN SOLVING MATHEMATICAL PROBLEMS – WHY IS COPY-PASTE OFTEN USELESS

Matija Lokar, Paul Libbrecht

Abstract: Mathematical formulae are information objects that can be entered in a computer, visualized, and evaluated. Thus, by the majority of (mostly occasional) users it is also expected that they are transferable through the simple copy-paste procedure. This transfer is particularly interesting when users are involved in tasks that span different mathematical activities or domains. For example, when performing computations and writing a report about them, or when performing algebraic computations from geometric constructions. Essentially, using copy-paste also allows users to use a particular mathematical software for the tasks that it does best.

To uncover the possible difficulties within this process, we approached students who are beginning to learn the use of mathematical tools. Through analysing their home-work where they report on the usage of various tools we observed their use of the copy-paste transfer procedure, both from their reports and from the dialogue they had with the teachers. Their attempts show a multitude of issues which we try to explain. It appears that the copy-paste procedure is often useless, especially for inexperienced users.

Key words: copy-paste, transfer, representation, input

1. Motivation: Exchanging Formulae between Programs

When something is displayed and selectable on a computer screen, users expect it to be copy-paste-able: the copy function can be used and the selection inserted (pasted) into other places, within the same program or transferred between two different programs. Copy-paste is an exchange mechanism between different “places” inside documents. It consists of the following phases:

- An origin “content” is selected (e.g., with the mouse).
- It is copied; this transfers several copies of the content at the origin into the system’s clipboard, one for each *flavour*.
- A target place is selected in the document being edited.
- The contents of the clipboard are pasted. The application editing the document chooses which *flavour* is best and inserts its associated content in the given place.

Operating systems and applications have been offering this procedure for decades. It is already part of the daily routine; at least for texts and images.

Mathematical formulae displayed within various computation or editing systems could, in principle, also be transferred. How successful is this? How often do users assume that it is possible to use the copy-paste procedure for the exchange between systems and how often does this procedure fail to produce the desired result? Should there be more effort put towards making the exchange process between applications smoother? These questions represent the core of this paper.

This appears particularly interesting as mathematical software is diverse and rich, and each software can perform some tasks very well and others only below par. For example, it is natural within a dynamic geometry system to study the geometry of various configurations using the visual aspect, but it is quite difficult to obtain the symbolic expressions of the coordinates of an intersection. Similarly, preparing

the layout of a document containing mathematical formulae with word-processing or page-layout software is feature rich, flexible, and often considered easy. It can, however, also be done with computer algebra systems, sometimes with spreadsheet systems, or even with geometry systems, but doing so appears to be a largely unrealistic task where users have to agree to multiple compromises and may have to do considerably more work to achieve the result of a similar quality.

Our observations confirmed the initial postulation that students mostly do not expect any difficulties in using the copy-paste procedure; contrary to more experienced users, who are well aware of the limitations, as they understand the concepts behind the representations of each system. The students were surprised when things did not work or even worse – they performed the procedure without noticing the incomplete or wrong transfer of a mathematical object due to the copy-paste procedure. An experienced user can find a roundabout solution for quite a few of the reported problems, as they understand the meaning of the underlying representations. However, the main emphasis in this paper is on the problems an inexperienced user, as students mostly are, has when using the copy-paste procedure.

It is thus argued in this paper that more emphasis should be put on the possibilities of exchange between programs, and that it should be investigated to what extent such exchange is expected from current users.

1.1 Outline

The paper is organized as follows: first, we describe the backgrounds of the students that we observed, including the competencies aimed at. We then present techniques that we used to analyze the copy-and-paste processes from a technical point of view and some of the most typical difficulties and issues that the students expressed concerning the usage of the copy-paste procedure. An attempt of analysis of the origins of those difficulties follows, where we describe what we consider the expectations of (especially inexperienced) users to be and suggest possible future avenues of research.

2. Related work

Very few academic papers describe user studies of the copy-paste function, and there are even fewer papers on the topic that discuss its use for mathematical purposes. A mention as a positive feature of copy-pasting TeX expressions is found in [21]. The survey [12] mentions it in connection with graphics only. The gesture survey [19] indicates the copy-paste feature (in abstraction) as an aspect of user interface difficulty, since the user may obtain different services depending on the selection.

Industrial specifications, however, commonly describe the operations that application programs should follow as being copy-paste, including user-interface guidelines. Notably, this includes the concept of Uniform Type Identifier to describe the data-type (generally called *flavour*) stored in the clipboard on Apple-branded computers [2], and the predefined data-types available in Windows environments [16].

Industrial standards attempt to support widespread use of agreed data-types. Among others one can find MathML [6] which attempts to define its clipboard flavours in the media-type declaration (in Annex B), a practice which has been followed by a few other standards.

Mathematical input can be considered within a *register*, a set of conventions and signs which represent mathematical objects [9]. In the technology-based landscape registers include the affordance to input mathematical objects and to apply operations to them. Examples of technology-based registers include formulae for computation systems, formulae for typography, geometric configurations, or tabular data. While these registers appear closed, many translations between these registers are feasible (e.g. displaying a formula as a result of a computation, converting a geometric configuration into a symbolic expression, plotting a data-set, or converting a formula from a text-document to a computable one). The copy-paste procedure expands the spectrum of possible operations on mathematical objects.

The book [13] gives broad spectrum of educational and mathematical innovations that technology empowers. As an eloquent description of the potential operations on mathematical objects, this book contains [20]. It depicts multiple facets of mathematical assessment tools including the very diverse procedures that evaluate mathematical objects.

We thus see that the literature we have found describes the copy and paste functionalities as a potentially important aspect but never as a central feature of the mathematical activities. Hence, the present paper

is an attempt to gather the current state of anticipations on how a beginner user would expect to use the functionalities of computer tools in their mathematical activities and how these expectations tend to succeed or fail.

3. Experiment Framework: A Course in Practical Mathematics

In order to investigate the possible difficulties in using the copy-paste procedure mathematical software users, especially beginners, could have, we have chosen a population of users of mathematics software which, we believe, represents a fair share of the mathematics professionals of the future: students of mathematics.

We chose to observe the students in the first cycle of the professional study programme *Practical Mathematics* at the Faculty of Mathematics and Physics, University of Ljubljana, where a course called Computer tools in mathematics is taught. The primary goal of the programme is to produce professionals capable of applying their mathematical knowledge to the workplace processes involving communications and information technology, (e.g. in technology or logistics companies). There are several subject specific competences to be developed by the students. Each student should have the ability to:

- **employ mathematical tools for practical problem solving,**
- identify mathematical processes in a professional environment,
- **analyze results,**
- **present results,**
- employ mathematical tools in everyday environment,
- initiate and conduct software upgrades,
- optimize business processes,
- model mathematically technological processes,
- employ numerical methods for mathematical problem solving.

In the preparation of the course, we focused mainly on the highlighted competences. Therefore, the main aim of this course is to show the students the practical usage of various tools in solving mathematical problems. Students learn about and learn to use techniques for computer tools suitable for various mathematical procedures. An important aspect of it is the learning of how to report on their work.

The syllabus is primarily focused on getting used to the programs for numerical and symbolic calculations, drawing graphs of functions, and solving mathematical tasks in analysis and algebra with a computer. Tools that are exposed include computer algebra systems, spreadsheets, visualization and computer-aided dynamic geometry systems. We agreed on the following intended learning outcomes:

- **Knowledge and understanding:** The students are familiar with computer tools and approaches to their usage in mathematical procedures. They are able to report on the process of solving a mathematical problem with computer tools.
- **Application:** The students are able to employ the knowledge gained to support various subjects.
- **Reflection:** The students are able to recognize the importance of computer tools in learning and doing mathematics and to report on solving mathematical problems.
- **Transferable skills:** The students are able to embrace other mathematical subjects employing learning tools, in order to deepen their knowledge of these subjects.

This course appears to be a well-chosen space to investigate what users naively do with the copy-paste procedure: In many cases, users are likely to be stronger at creating or studying mathematical topics within one environment instead of another. This will thus naturally lead them to expect the copy-paste procedure to work.

4. Observing the Clipboard Technically

As it has already been explained, the clipboard (the “place” containing what is being copied) is made of a series of data-fragments labelled by a *flavour*: source applications export all flavours while receiving applications choose the representations they consider best for importing data within the current insertion process. A single exception to this rule is the use of the copy-paste procedure within a single application which then has the ability to copy by reference and thus fetch the data in the internal representation of the program for the source.

To observe what is being exchanged over the clipboard, little programs called clipboard inspectors can be used. We present the two which were used: In Windows, the open-source project *ClipSpy* [8] provides a simple overview of the names and a hexadecimal dump of the content of each flavour. Similarly, in MacOS, *Clipboard Viewer.app* is a program available as sample software from Apple developer tools.

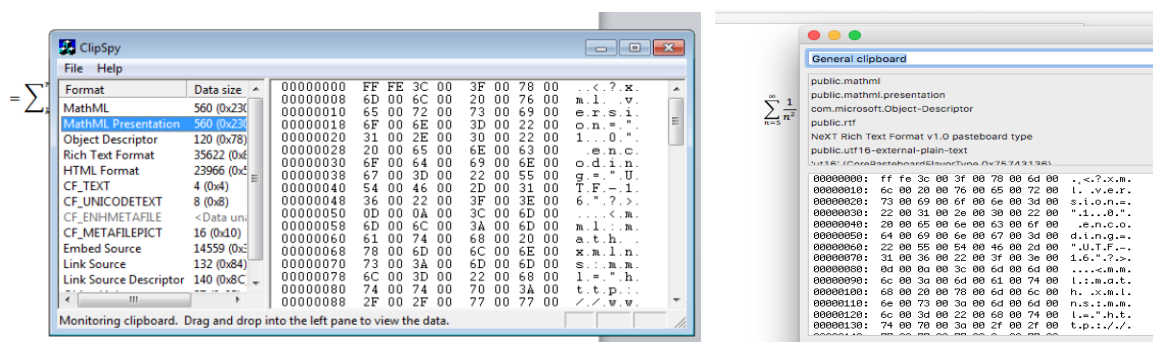


Figure 1: Screenshot of Clipboard Viewer (left) and ClipSpy (right)

A screenshot of each is presented in Figure 1.

The usage of these tools has allowed us to see what flavour has been used in exchanges we have reproduced below.

5. The Observations

After an introductory period during which students get a basic knowledge of the functionalities of a few applications (computer algebra (Mathematica), numerical processing (MatLab), dynamic geometry system (GeoGebra), layout tool (Microsoft word, LibreOffice Writer) students got an assignment to solve several mathematical tasks where the tools were used in the solution process and to report on the whole process. Some were allowed to use web-based editors such as the Wiris Input Editor.

The task given was to solve a mathematical problem and to present its solution in electronic media. They also had to write the report on their approach to the solution, describing the way they used software and to report on possible difficulties and expectations they had beforehand with their use of tools. In all the given problems the usage of different mathematical softwares was implicitly (or sometimes even explicitly) foreseen. Most often a combination of computer algebra system, dynamic geometry system, and numerically oriented matrix software was required. Therefore, even when certain tasks could be solved within one environment (software), the students' use of multiple softwares was highly likely, due to their limited familiarity with them.

For reporting, they mostly used the most commonly used word processing software. We were especially interested in observing how they would cope with the process of exchanging the mathematical object between programs and how they would report on that.

Here is an abbreviated example of a task (an important part of instructions was also the requirement to consult the lecturer and the teaching assistant about the task when they were directed towards the appropriate use of tools and the copy-paste process):

For the seminar paper, you have to choose one of the tasks given below. You have to study it thoroughly and consider all possible special cases. In the solving process, you will combine your mathematical knowledge as well as the capabilities of computers. To successfully solve the task (which, "purely" mathematically, should not be a problem for you) several of the steps have to be solved with computer tools we had learned about. It is expected that you will use various tools. If none of the tasks given is interesting for you, you may suggest another problem (which will (or not) be approved by the lecturer):

- How to solve a term paper in the Mathematics 1 course with computer tools?
- Different approaches in proving the Pythagoras theorem.
- Platonic solids.
- Euler's line with the similarity of triangles and trilinear coordinate system.
- Compare the capabilities of MATLAB, Mathematica, and GeoGebra in solving systems of equations.
- The Fibonacci sequence: geometrical meaning and usage
- ...

A very important part of your job is to write a report about the problem. Here you should use the capabilities of the word processor. Use results obtained from the tools and insert them in the report, as well as screenshots of the "tool in action".

Beside the solution process, you should also report to which extent the clipboard (copy-paste combination) is useful when transferring a mathematical object from one computer tool to another. For example, we perform a certain computation with MATLAB. Now we would like to transfer the matrix obtained to Mathematica (or to GeoGebra or to ...). Of course, the majority of those exchanges will occur during the writing of the report where equations and such will be transferred from or to the word processor. Describe your expectations and the experiences you got!

6. Selected Cases

In this section, the students' observations from their seminar papers are gathered and representative samples are presented. We also used information from the consultations students had with the teaching assistant and the lecturer. We group these observations according to the pair of software tools used. The majority of examples presented on pictures are taken from the student's reports, however some of them were reproduced by us to further analyse them (via Clipboard-viewer for example), to group them accordingly or as well as to improve the pictures quality which were sometimes unsuitable for publication.

Perhaps this quote from one of the seminar papers summarize best the opinion students had "It was an extremely tough task. I spent numerous hours doing it. I tried several techniques to force the programs to properly copy-paste the expression with no success. Finally, I transferred the expressions manually. But here I had problems again as even the usage of parentheses, not to even mention the names of commands, was not the same. I do not understand why each software producer uses their own syntax."

For quite a few of the reported problems, an experienced user can devise a workaround. However, the main emphasis in this paper is on the problems an inexperienced user, as students mostly are, has in using copy-paste procedure.

6.1 From Word to Mathematica

The most common pattern of use was MS Word and Mathematica pair. Through observations (mostly from consultations students used) we identified transfers of mathematical objects in both directions – from Word to Mathematica as well as from Mathematica to Word.

In Word, the student enters the expression using *Insert > Equation*, marks the expression, uses Ctrl-C (the copy operation), switches to Mathematica and uses Ctrl-V (Paste). How successful were the students in this first part?

If using that method of copying, a warning appears when the computation is requested, depicted in Figure 2, which indicates that interpretation issues may appear. As reported in consultations students usually decided not to press *Always* and were bothered by the warning at every pasting attempt.

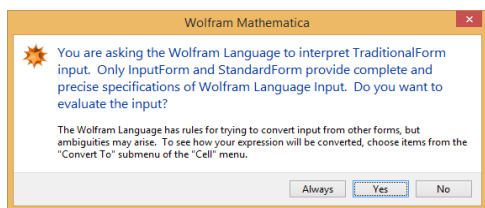


Figure 2: Mathematica's warning

$$\begin{aligned} \text{In}[1] &:= \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \\ \text{Out}[1] &:= \sum_{n=1}^{\infty} \left(\text{Cos} \left[\frac{n\pi x}{L} \right] a_n + \text{Sin} \left[\frac{n\pi x}{L} \right] b_n \right) \\ \text{In}[2] &:= \frac{2}{3 + \frac{1}{a-b}} \\ \text{Out}[2] &:= \frac{2}{\left(3 + \frac{1}{a-b} \right) (a-b)} \\ \text{In}[3] &:= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ \text{Out}[3] &:= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \end{aligned}$$

Figure 3: Complex expressions pasted from Word into Mathematica

Many examples perform well. The intuitive exchange works and further computations can be done right away. Mathematica had no problems with recognizing simple expressions; it properly recognized multiple elaborate expressions depicted in Figure 3.

Of course, there are issues when students employed notations, which are beyond Mathematica's notations. Examples reported include $\text{sen}(\alpha)$ for the sine function of alpha in the Spanish speaking countries (as reported as widespread notation for the sine and cosine functions on notations.hoplahup.net/Census/CD-transc1/sin – among students we observed we had two Erasmus students from Portugal), or the bullet for the multiplication (see Figure 4).

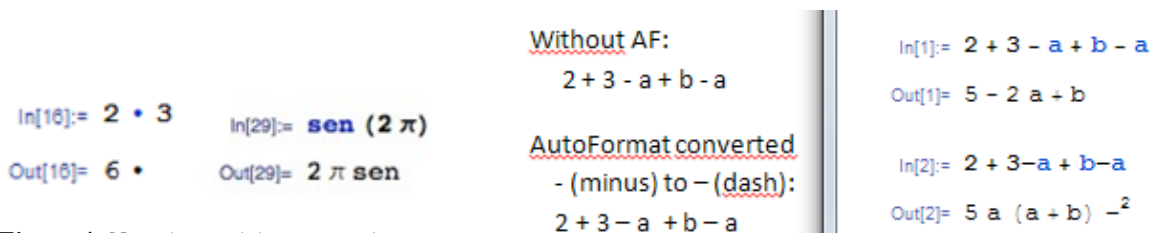


Figure 4: Notations misinterpretation.

Figure 5: Word-Auto-formatted expressions in Mathematica

Other input modes: We assumed that students employ the menu command *Insert > Equation*. We, however, frequently observed that simple formulae are generally typed as part of the text and that special transformations (called auto format - AF) support this practice: thus 6-b becomes 6 – b (where the minus sign becomes a dash), similarly 1/4 becomes ¼. These characters stay characters when pasted into

Mathematica and are thus treated as a variable-name as in Figure 5: Mathematica treats — (the dash form) as a variable when 2+3-a + b -a is entered as text. Screenshots in Word (left) with auto-format (above) or not (below), pasted in Mathematica (right). Students usually do not expect that as they blindly assume correctness of copy-paste procedure.

When using *Free form* input in Mathematica, it facilitates some of the issues described above (notably the dash instead of the minus). This mode, however, did not succeed with other parts which worked normally, notably on all variables that are made of several letters.

6.2. From Mathematica to Word

The probably most used transfer of mathematical objects via copy-paste students used was from Mathematica to Word. Having worked with Mathematica, students have some results in Mathematica's notebook and would like to copy them into Word. Many students made snapshots of Mathematica's window, crop it appropriately, and then inserted them into Word as a picture. This is clearly a non-mathematical practice which we shall not discuss. We are interested in the process where expressions stay expressions; it, however, satisfies the students as the consistency of display is insured.

If the same copy-paste technique is used, we did not observe any problems. The expression is entered into Word as an equation. This equation can be modified or used.

$$a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$

Figure 6: With classical notation, input in Word.

$$a_0 + \sum_{n=1}^{\infty} \left(\text{Cos}\left[\frac{n\pi x}{L}\right] a_n + \text{Sin}\left[\frac{n\pi x}{L}\right] b_n \right)$$

Figure 7: Changed notation, back from Mathematica.

However, the situation is not an ideal one: A couple of students inquired the teaching assistant how to insure that notations in Word and Mathematica be more similar. Due to some peculiarities of Mathematica's syntax, such as that of using square brackets for functions' arguments and function names starting with the uppercase letter are not converted back. So from Word, the equation in Figure 6 transferred to Mathematica and back, is converted to the equation in Figure 7 to which careful students object.

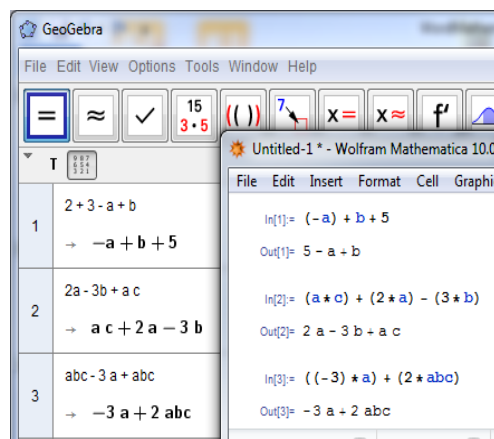


Figure 8: From GeoGebra to Mathematica: Simple expressions.

6.3 Between GeoGebra and Mathematica

One of the possible tasks students could choose was to compare capabilities of Mathematica and GeoGebra in dealing with basic algebraic expressions. As they have an apparently similar syntax, one of the student wrote in his report "as I was short in time, I picked this task expecting everything will be mostly the same and thus my seminar paper quickly done. However I had major problems". Also several other students reported on different issues occurred during the copy-paste transfer between this two programs. This section builds on issues reported by students and reports to what extent

Mathematica and GeoGebra (in CAS mode) are capable of exchanging expressions.

The reader is reminded that formulæ in GeoGebra CAS's output are copied using the contextual menu triggered by a right click on the expressions only. Thus only complete expressions can be copied.

If only basic operations are used, as can be seen in Figure 8, there are no special issues, except for a minor fact that the copy operation in GeoGebra already equips the expression with additional parentheses and inserts the multiplication symbol.

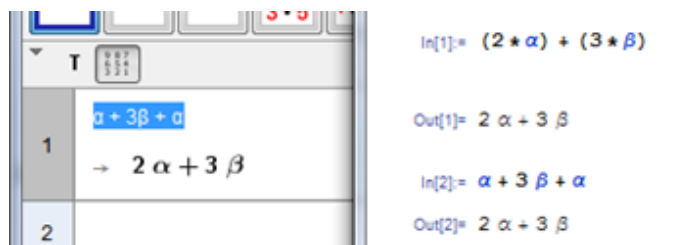


Figure 9: Using special symbols from GeoGebra to Mathematica.

There are also no problems with the usage of special symbols for variables, such as α or β (Figure 9), regardless of whether we copy from the input or the output expression.

Using *Free Form* input in Mathematica (starting the expression with = symbol) is problematic whenever variables with long names are used. As GeoGebra and Mathematica use similar notations (essentially characterized by square brackets for commands), there are no problems whenever commands with the same name and the same number of parameters are used (Expression 1 in GeoGebra vs. Expression 10 in Mathematica in Figure 10). This interoperability rapidly meets its limits, however. For example the command Factor in GeoGebra has a version with two parameters which is unknown to Mathematica (compare expressions 2, 3, and 4 in GeoGebra with expressions 11, 12, and 13 in Mathematica in Figure 10 and attempt using free-form-input in Mathematica in Figure 11).

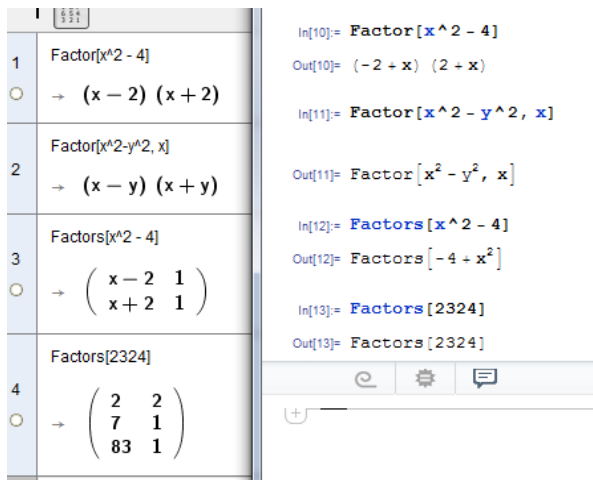


Figure 10: The Factor command in GeoGebra and Mathematica

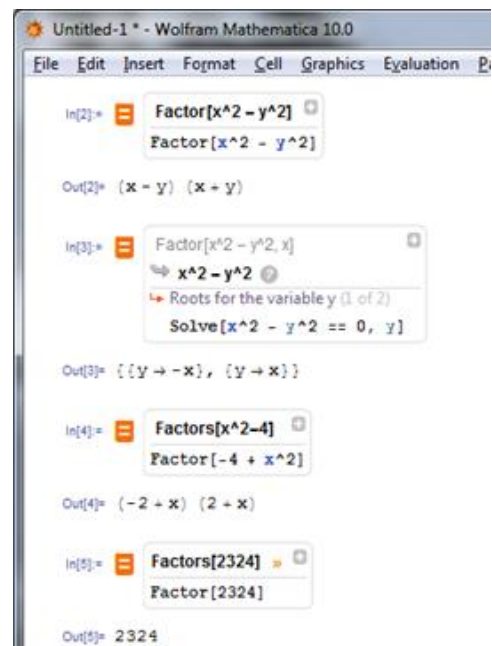


Figure 11: Using Free form input with copied Factor and Factors command from GeoGebra

Now take the opposite direction: Mathematica to GeoGebra.

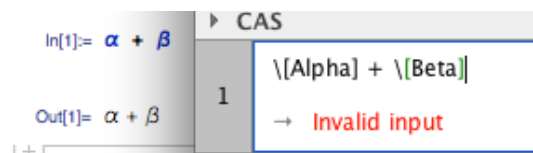


Figure 12: Copying Greek symbols from Mathematica to GeoGebra.

Ordinary expressions are not problematic, as long as special symbols are not used. However, copying expressions where we were previously successful, such as $\alpha + \beta$ now fails. As can be seen in Figure 7, Mathematica inserts its own syntax (which has a LaTeX style): this creates errors in GeoGebra (Figure 12).

Even the use of alternative copy functionality (*Copy as...*) is unsuccessful. For example, Figure 13 displays the result of pasting after having “Copy as MathML”.

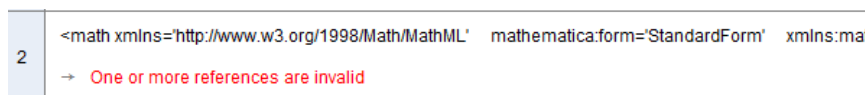


Figure 13: Using Copy As MathML then pasting in GeoGebra

With commands that are common to both Mathematica and GeoGebra, a transfer can happen – Mathematica copies them to the clipboard as “plain text”. When GeoGebra does not recognize the command, it also changes the parentheses from square to ordinary ones.

6.4 Wiris Input Editor and Mathematica

Some of the students decided to use the Wiris input editor as part of a reporting tool. As in the previous section, exchanges between two similar function-scopes are explored so that they match each other in different situations (e.g. Wiris when on browsers-only and Mathematica when at the lab, or Wiris to facilitate input, e.g. with pen input, and Mathematica to solve difficult differential equations...).

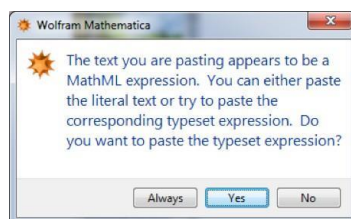


Figure 14: Mathematica warning about MathML.

The Wiris Input Editor is a web-based component and thus cannot export arbitrary flavours; instead, it puts MathML in the clipboard as plain text. Pasting MathML is *sniffed* at by Mathematica. That is, Mathematica detects that the plain text is not to be interpreted as source (generally expected from plain text). Thus, it displays a dialog proposing to paste the plain text (the MathML source) or interpret MathML (Figure 14). It is probable that eager users will answer always, but users that expect to manipulate MathML sources might feel uncertain doing so and will thus be bothered by such a dialog every time.

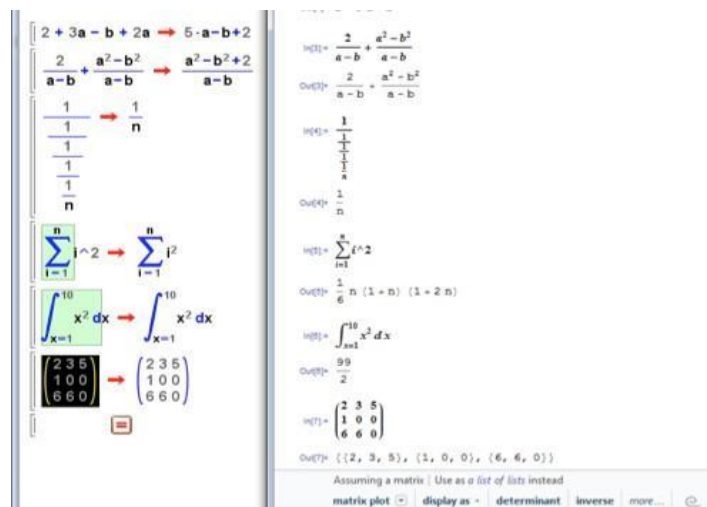


Figure 15: Copying Wiris to Mathematica

After MathML is input, a similar warning as that of Figure 3 about the usage of Mathematica's traditional form is displayed. It may also inform users that a possibly failing interpretation of MathML-presentation into a computable form will be performed. After going beyond these warnings, and using proper notations, the exchanges work rather well, as displayed in Figure 15.

However, these exchanges failed for some mathematical symbols, which are not interpreted as known notations by Mathematica. Among others, while the binomial coefficient is exchanged well; the permutations count (without repetitions) is not.

6.5 Matrices from Octave to Mathematica

One of the students who started his report with "I am going to show how matrix objects can be copied from one program to another. The programs in question will be Mathematica, GeoGebra, Octave and

Python (numpy).” quickly realized “*Since a matrix is represented differently in each of the programs, we cannot simply copy-paste it. Some conversion must be done.* “. What follows are selected problems students reported on.

The first problem arises when we try to define a vector (Figure 16).

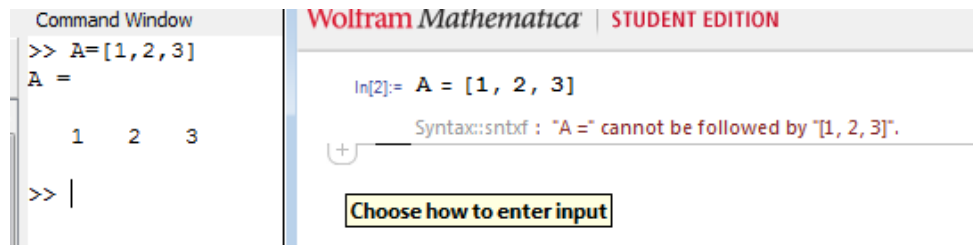


Figure 16: Vectors from Octave to Mathematica

Vectors in Octave are entered using square brackets, while curly brackets should be used in Mathematica. The same confusion happens when matrices are used (Figure 17).



Figure 17: Matrices from Octave to Mathematica

While trying to copy the defined vector or matrix from Octave to Mathematica, we get the product of each line as a result (Figure 18).

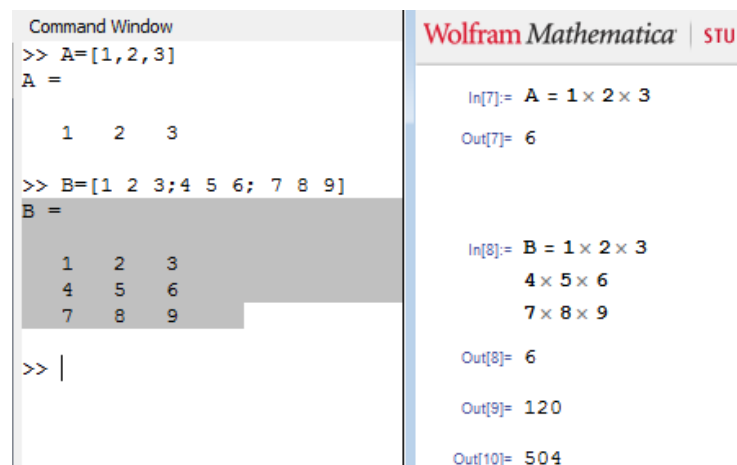


Figure 18: From Octave output to Mathematica input.

Elementary operators seem to work better between these two systems.

It is obvious from Figure 19 that while addition and subtraction work, multiplication in Mathematica multiplies element by element, while Octave does matrix multiplication. Transposing does not work either. Element by element division in Octave cannot be copy-pasted either.

```

>> A+B
ans =
    5    7    9
    5    7    9
   14   16   18

>> A-B
ans =
    3    3    3
   -3   -3   -3
    0    0    0

>> A*B
ans =
    66    81    96
    30    36    42
   102   126   150

>> A'
ans =
    4    1    7
    5    2    8
    6    3    9

>> A./B
ans =
    4.00000    2.50000    2.00000
    0.25000    0.40000    0.50000
    1.00000    1.00000    1.00000

>> |

In[25]= A + B
Out[25]= {{5, 7, 9}, {5, 7, 9}, {14, 16, 18}}

In[19]= A - B
Out[19]= {{-3, -3, -3}, {3, 3, 3}, {0, 0, 0}}

In[20]= A * B
Out[20]= {{4, 10, 18}, {4, 10, 18}, {49, 64, 81}}

In[21]= A '
Out[21]= {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}'

In[27]= A ./ B
Syntax::snbif: "A." cannot be followed by "/B".

```

Figure 19: Elementary operations from Octave to Mathematica.

When performing the transfer from Mathematica to Octave, the first issues are met right at the beginning since matrices are converted to *cells* which are mathematical objects of a different nature (Figure 20).

```

>> A = {{4, 5, 6}, {1, 2, 3}, {7, 8, 9}}
A =
{
  {
    [1,1] =
    {
      [1,1] = 4
      [1,2] = 5
      [1,3] = 6
    }
  }
  {
    [1,1] = 1
    [1,2] = 2
    [1,3] = 3
  }
  {
    [1,1] = 7
    [1,2] = 8
    [1,3] = 9
  }
}
>> x = {1, 2, 3}
x =
{
  [1,1] = 1
  [1,2] = 2
  [1,3] = 3
}
>> |

Wolfram Mathematica | STUDENT EDITION

In[27]= x = {1, 2, 3}
Out[27]= {1, 2, 3}

In[23]= A = {{4, 5, 6}, {1, 2, 3}, {7, 8, 9}}
Out[23]= {{4, 5, 6}, {1, 2, 3}, {7, 8, 9}}

```

Figure 20: Mathematica to Octave.

7 Analysis

In this section, we first present a feedback extract which gives representative feelings of the students towards the copy-paste procedure. We then try to discuss the issues that most of the students encountered. In the last part we sketch how one could approach the construction of a more useful copy-paste procedure.

7.1 Individual Feedbacks

Some observations and remarks of the students, copied from their seminar papers and expressed in consultations with teaching assistant and lecturer (translated from Slovenian by the authors):

- *“... Copy-paste method does not work with drawing plots in different programs. For drawing plots each of the programs used has its own ‘language’ which we have to use. “*
- *“When we copied expressions from Word to GeoGebra, it worked in some cases even without changing expressions. If exponents were written with symbol ‘^’ in Word, there would not be any problems. From there on, some changes and adaptations were necessary, especially when copying expressions from GeoGebra to Mathematica”*
- *“... I was very rarely able to use the copy-paste method to transfer expressions between programs and even then some corrections were usually needed for the programs to work”*
- *“After several attempts I gave up. The only way to transfer an expression between the programs is to manually retype it ...”*
- *“Entering a matrix in X is surprisingly identical to entering the matrix in Y. But unfortunately this is where the fun ends.”*
- *“Apart from matrix and scalar addition, and matrix scalar multiplication, the commands for matrices, even though really similar, are not compatible and are misinterpreted by those three (Octave, GeoGebra, Mathematica) applications.”*
- *“In this experiment we found that Octave and Mathematica are barely compatible, so copy-pasting from one to another would almost never work. They are only compatible in two basic operators + and -. Trying to copy anything from one and pasting it into another, would either generate an error or return a completely wrong answer.”*
- *“Mathematica converts expressions, but it works poorly.”*
- *“If we look back, we can see that matrices did not do well after the copy-paste procedure; be it in Mathematica or in Octave. But this is not the only issue. Equations are not capable of adapting from Mathematica to Octave and the other way around, either. Calculus has fewer problems with variables/unknowns, while basic math operations/expressions do not have any.”*
- *“Please do not give us such tasks anymore. It is impossible to switch between different syntaxes all the time. At the end I did not even know if I had got everything right, even though the mathematics behind the problems was simple enough.”*

In the discussion afterwards, they all expressed their profound disappointment with the software. When the students started the task, most of them expected some minor problems. As one student said, *“it is all mathematical software and there is but one mathematics, so I expected no problems in using the copy-paste procedure”*. From their feedback, a global frustration can be inferred: the programs do not allow a transparent continuation of work from one environment to another.

7.2 Issues to Copy-and-paste

Based on the examples, we deduce that the following issues are common to beginner users of mathematical systems:

- **Encodings mismatches:** While some programs transfer much of the information into the clipboard, allowing other programs to recover most information in principle, quite a few merely include information such as the identified meaning of an operator or the desired spacing around them in encodings which are difficult for other applications to receive in a consistent manner.

We could observe this behaviour when copying, e.g., from Mathematica to GeoGebra (e.g. the Greek letters), from Wiris to Mathematica (e.g. the permutation count), or when copying vectors from Octave to Mathematica. We have not encountered this issue for visual mathematical formulae when carried from Mathematica or Wiris to Word for example (where MathML-presentation seems to be successfully complete).

- **Preference for pasting plain text when other formats may be more appropriate.** It appears that most programs we met generally choose to export content in plain text and often prefer to paste it so. This is an issue because other encodings seem to be ignored even though they could offer a far richer experience and their encoders are already available in these systems (this is the case of Mathematica or Maple for example). Such a preference can be observed when copying Greek-letter variables from Mathematica (which then prefers to export the TeX-syntax of the Greek letters). This is also shown to be an issue in the Octave-Mathematica or GeoGebra-Mathematica exchanges described above: the syntax compatibility rapidly meets its limit.

This is also necessary for the Wiris Input Editor, being a JavaScript widget which has no authorization to export MathML or other flavours; the experience remains somewhat positive thanks to content-sniffing, the practice of analysing the bytes of a piece of content in order to bypass the flavours declarations and decide on the better of the flavours.

- **Misinterpretation of mathematical notation:** Employing a layout software to input mathematical formulae leaves the user quite free to use mathematical notations in a way that does not make sense. What makes sense or not, however, is not shared across the planet. Examples include the usage of the cross-sign for multiplications or the word *sen* to denote the sine function (which fails from Word to Mathematica). Similarly, different softwares use different function names which are not properly exchanged, e.g. the symbol P to denote the permutations' count from Wiris to Mathematica. Dedicated *extensions* of the parsing engines are offered but then compromise other aspects (e.g. the *free-form* of Mathematica).

- **Lack of usage of more semantic encodings:** All of the computer algebra systems described in this paper include a rich set of mathematical operators and functions. However, any interoperability they offer to the outside world is either in linear syntax (known to be very program-specific) or in presentation-oriented syntax (such as MathML presentation or pictures). We see that this usage rapidly encounters issues in the Octave-Mathematica or GeoGebra-Mathematica exchanges above.

- **Partial but unpredictable success:** Probably the biggest user-perceived issue, and what we have seen as the most frustrating aspect of the exercise is the non-uniformity of exchange capabilities: The examples described above show a very diverse set of interoperability issues: from elementary in/compatibilities to subtle mismatches that simply need a text adjustment. There appears to be no way of explaining to a user in simple terms, the rules of how to tell whether a formula that has just been processed will be transferred successfully.

7.3 Future Work

Many of the issues are clearly related to the fact that most software makers predict only a few possible software targets where the copy and paste exchange could happen. While one sees dedicated attempts to make things work (e.g. Mathematica to Maple, according to the software makers), it appears that there is probably too big a diversity to attempt simple pair-wise checking and that abstraction is required.

This abstraction is provided by standardized encodings: when exchanging pictures, for example, it is clear that raster pictures use one of the current formats (PNG, TIFF, JPEG, PDF ...). Would plain-text syntax be sufficient for this? This is not usually the case (e.g. when inspecting the GeoGebra-Mathematica exchanges). Attempts at making TeX such a standard have almost always failed since configurable macros are almost always desired and are not exchangeable.

Standardized encodings are provided by such standards as MathML [6], OpenMath [18], or the yet to become I2G [1] with a definition made outside the scope of a precise software and based on the consensus of software makers. Indeed, the use of MathML-presentation by the Wiris editor was successful to a large extent.

We claim that a computer algebra system that would perform its copy function as MathML-content, OpenMath, MathML-presentation, HTML-with-MathML, vector and raster pictures, and plain-text would have a much bigger success than what can currently be observed. Other computer algebra systems would understand all basic expressions of MathML content and OpenMath (avoiding the problems between Octave and Mathematica, and most of the problems between GeoGebra and Mathematica), paste them properly as formulae within Word-processing applications, email programs, and web-based editors, as vector and raster pictures in the respective applications, and finally as plain-text for all others. The use of proprietary syntax might be added to the mix (e.g. the input source syntax, and probably richer representations), but it should not be in the plain-text flavour but in dedicated flavours which indicate the syntax and version used.

To evaluate this claim, and to support the users in reporting a more precise reporting and inspection of the clipboards when involved in a mathematical task where reporting is not their main focus, we plan to construct a help tool that is able to detect the origin of a clipboard, perform a large part of automatic conversions as are generally available (e.g. the numerous parsers for Maple or Mathematica, online conversion services based on these software), and enrich the clipboard. This tool would also be the central place to report on the utility and success of a copy-paste procedure when performing a particular mathematical task. Such a tool would allow a more statistically exact evaluation of the users' needs, beyond the scope of single experiments within a lecture.

Another area of research lies in the mathematics-educational dimensions implied by the use of the copy-paste procedure: It represents a truly computer-empowered *conversion* between registers in the sense of [9]. As indicated there, a conversion keeps one part of the *content* and loses another. The mathematical exercise of representation change lies nearby and is likely to be an effective exploration of mathematical objects, provided appropriate questions are posed to the students about the transferred objects: A simple operation such a function evaluation, a derivation, or another well-known property check allows any user to verify that a copy-pasted expression has indeed been transferred as a function. Applying such strategies in teaching can support critical thinking in the use of the computer tools where mathematical operations support the user in questioning the validity of the objects being manipulated. A mix of media critiques and mathematical competencies is thus developed, especially if some of the programs used come from sources of unknown reputation such as web-pages or freshly installed software.

References

- [1] M. Abanades, F. Botana, J. Escribano, M. Hendriks, U. Kortenkamp, Y. Kreis, P. Libbrecht, D. Marques, and C. Mercat (2009), The Intergeo File Format in Progress, *Proceedings of the 22nd OpenMath Workshop 2009*, online.
- [2] Apple Inc. (2009) *Understanding Uniform Type Identifiers, Technical documentation*, http://developer.apple.com/mac/library/documentation/FileManagement/Conceptual/understanding_utis/.
- [3] Brownfield, K. R. (2007), *MIME media type Application/ IETF Tree-mathematica*, <http://www.iana.org/assignments/media-types/application/mathematica>. 3
- [4] W3C Math Group (2012), *Media Type Registrations*, <http://www.w3.org/TR/MathML3/appendixb.html>
- [5] D. Carlisle (2010), *MathML on the Clipboard*, Blog entry. <http://dpcarlisle.blogspot.de/2010/01/mathml-on-clipboard.html>.
- [6] D. Carlisle, P. Ion and R. Miner (2013), *Mathematical Markup Language, Version 3 (2nd edition)* W3C.<http://www.w3.org/TR/MathML3/>.
- [7] D. Cheng (2015), *Clipboard API: remove dangerous formats from mandatory data types*, public-webapps mailing list post on 2015-06-09. Archive of the thread visible at <https://lists.w3.org/Archives/Public/public-webapps/2015AprJun/0819.html>.

- [8] M. Dunn (2001), *ClipSpy*, Software page at CodeProject.
<https://www.codeproject.com/articles/168/clipspy>
- [9] R. Duval (1993). Registres de représentations sémiotique et fonctionnement cognitif de la pensée. *Annales de Didactique et de Sciences Cognitives*, 5, 37–65. Available at <https://mathinfo.unistra.fr/irem/publications/adsc/>
- [10] E. Kiciman and A. Fox (2000), Using Dynamic Mediation to Integrate COTS Entities in a Ubiquitous Computing Environment. in *Handheld and Ubiquitous Computing: Second International Symposium*, HUC 2000 Bristol, UK, Proceedings, pp. 221–226..
- [11] T. Kindberg and A. Fox (2002), System software for ubiquitous computing. *IEEE pervasive computing*, 1(1): pp. 70–81.
- [12] N. Kajler and N. Soiffer (1998), *A Survey of User Interfaces for Computer Algebra Systems*, Journal of Symbolic Computation 25(2), pp. 127-159.
- [13] C. Hoyles and J.-B. Lagrange (2010), *Mathematics Education and Technology—Rethinking the Terrain*. The 17th ICMI Study. New ICMI Series. Springer Verlag. ISBN 978-1-4419-0146-0
- [14] H. Lieberman and T. Selker (2000), *Out of context: Computer systems that adapt to, and learn from, context*. IBM Systems Journal, 3.4(39): pp. 617–632.
- [15] P. Libbrecht, É. Andrès, and Y. Gu, (2009), *Smart pasting for ActiveMath Authoring*, MathUI 09 Workshop, <http://www.activemath.org/workshops/MathUI/09/>.
- [16] Microsoft Inc. (2016), *Clipboard Class*, API Documentation available at <https://msdn.microsoft.com/en-us/library/system.windows.clipboard.aspx> (checked 2016-07-20).
- [17] B. A. Myers and C. H. Peck and J. Nichols and D. Kong and R. Miller (2001). Interacting at a Distance Using Semantic Snarfing. in *UbiComp 2001: Ubiquitous Computing*, Atlanta Georgia, USA, September 30–October 2, 2001 Proceedings, 305–314.
- [18] S. Buswell, O. Caprotti, D. Carlisle, M. Dewar, M. Gaëtano, and M. Kohlhase (2004), *The OpenMath Standard, version 2.0*. Technical report, The OpenMath Society. Available at <http://www.openmath.org/>.
- [19] L. Padovani and R. Solmi, (2004), An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics. in *Mathematical Knowledge Management: Third International Conference, MKM 2004*, Bialowieza, Poland. Proceedings, pp. 302–316. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.9746>.
- [20] C. Sangwin, C. Cazes, A. Lee, and K.-L. Wong (2010), Micro-level Automatic Assessment Supported by Digital Technologies, in [13].
- [21] L. Zhang and R. Fateman (2003), *Survey of user input models for mathematical recognition: Keyboards, mice, tablets, voice*. Computer Science Division, University of California.

Authors

Matija Lokar, University of Ljubljana Ljubljana, Slovenia, Matija.Lokar@fmf.uni-lj.si

Paul Libbrecht, German Institute for International Educational Research (DIPF), Frankfurt am Main, Germany, Paul.Libbrecht@dipf.de