# Maths Content Accessibility

Dominique Archambault
EA 4004 – CHArt-THIM
Université Paris 8
dominique.archambault@univ-paris8.fr

**Abstract**

In this paper we are going to study fist the way people can interact with mathematical expressions without the sight, and then we will consider the possible formats for representing Maths, the input methods and the accessibility problem encountered by people. Finally we will discuss the opportunity to start the development of some Maths Content Accessibility guidelines.

## 1 Introduction

Since a couple of decades, a lot of research projects have been dealing with mathematical expressions. After a long period when almost only LaTeX was used to represent mathematical content, the needs of online contents made it necessary to design new formats to represent it. The MathML format, designed by the W3C has become the biggest standard, and hundreds of pieces of software use this format. Nowadays, any application which deals with mathematics, if it does not use MathML natively, is able to import or export expressions in MathML.

In the field of Assistive Technology especially a number of projects are aiming at helping visually impaired people to work with mathematical expressions. Indeed Mathematics and more generally all subjects which involve calculation, that is Statistics and most Sciences, are of particular difficulty for visually impaired people. The main reason for this particular difficulty comes from the representations that blind and partially sighted people have to use to do Mathematics: essentially Braille and synthetic speech. In [6] we have brushed a quite comprehensive state of the art of this topic, and in [1] we have proposed some prospective ideas of concepts for future projects aiming at helping people with print impairment doing mathematical tasks.

## 2 Non visual access to mathematical content

### 2.1 Available non visual modalities

#### 2.1.1 Braille

Braille readers, who are a proportion of Blind people, varying according the countries and ages (especially the age where blindness occurred), may use one of the mathematical Braille notations that have been developed in several countries during the twentieth century (about 20 can be counted)[1]. Nevertheless, less and less Braille readers are able to read such specific mathematical Braille notations. There are multiple reasons to that. First these notations are very complex and difficult to understand and to learn. Since a few decades, the number of blind children without additional troubles (like behaviour or learning difficulties) have been decreasing (due to the fact that various causes of blindness can be avoided or cured). Another important factor comes from the way inclusive education is implemented in a number of countries. Indeed it is well known that inclusive education, with a high level of quality, is much more expensive than segregative education, because all specific education should be made by travelling personal. In lots of countries, the choice was made not to provide this specific education and

K. Yamaguchi, M. Suzuki (eds.): DEIMS 2012, volume 1, issue: 1, pp. 1-8

to leave the teachers and impaired pupils more or less on their own. For instance there is not enough travelling teachers for Mathematical Braille. This is a choice of society. Most of the pupils in such case will study Mathematics using simple linearisation of mathematical expressions, transcribed to literary Braille, which make formulas longer, and then a more difficult access to Maths and Scientific curricula.

#### 2.1.2 Synthetic speech

The second way of accessing Mathematical content is to use synthetic speech. This is used by a lot of Blind users, who cannot use Braille, and most partially sighted people. In that case the expression must be linearised into a sentence that is pronounced. This modality is much cheaper nowadays than Braille devices, and accessible to all visually impaired people who cannot read Braille. Editing is nevertheless more difficult than using Braille.

### 2.2 Problems due to use of linear modalities

As seen above, the two non visual modalities that can be used to work in Mathematics are intrinsically linear. Sighted users are traditionally using bi-dimensional layout which makes it easier to understand the mathematical semantic of formulas [10]. This layout has 2 main advantages that are not possible with linear modalities: first it allows the reader to apprehend the structure of expressions at a glance, thanks to the overview possibility provided by the sight, and in a second hand it allows to mark more easily the boundaries of the blocks.

Let us take an example: the simple fraction of $x+1$ over $x-1$. The visual bi-dimensional way of representing this fraction (1) the helps the reader to understand immediately that it is a fraction, and thanks to the layout too, the blocks do not need to be explicitly marked so the representation involves only 7 symbols. When linearised to simple ASCII (2) the number of symbols increases to 11, because the blocs have to be explicitly marked, and the reader will not have a chance to understand that it is a fraction before reading the sixth symbol.

$$\frac{x+1}{x-1} \qquad (1) \qquad\qquad (x+1)/(x-1) \qquad (2)$$

Speaking this expression would produce the following sentence: *"x plus one over x minus one"*. Here again the reader cannot understand it is a fraction before the middle of the sentence. Additionally it is quite longer to say than to read with the eyes (about 3 seconds).

With linear modalities, the reader has to:

- build mentally the understanding of the expression incrementally while the sight would offer the possibility to build the structure first and to populate it with values after;
- memorise the whole expression while the sight offers an additional memory on the sheet of paper.

The problem of marking the boundaries of blocks is the biggest in all linear representations, as well in Speech synthesis than in any Braille codes.

### 2.3 Length of Braille representations

Translating this expression into literary Braille, that is without using a specific Braille code will produce the result proposed in (3). The number of symbols increases again, up to 16. The reason is that there are not enough symbols in Braille to represent all characters necessary including letters in different alphabets, digits, punctuation signs and mathematical symbols. Standard Braille is using six dots, and therefore the number of different Braille characters is $2^8 = 64$. The result is that, in French Braille, the digits have a prefix, to differentiate them from some accentuated letters (1 is the same Braille code than 'â' : ⠡, with the number prefix ⠠, so it will be written ⠠⠡. All the mathematical operators have a prefix too.

⠦⠭⠠⠖⠠⠡⠴⠠⠒⠦⠭⠠⠤⠠⠡⠴ (3)

The specific Braille codes would make the representation of the formula much shorter. (4) shows the same formula in the 2007 release of the French code. The specific rules of this mathematical notation allows to reduce the size to 11 symbols. In that case the numbers cannot be mixed up with accentuated letters that cannot be found in mathematical expressions, and operators, that are frequently used in formulas, are represented on single symbols. Usually it allows to seriously reduce the number of symbols, based on specific grammar and rules, but it must be said that this example is rather simple so it ends up with the same number of symbols than the simple ASCII linearisation (2), while in most cases it will not be as efficient.

⠰⠭⠖⠡⠆⠲⠰⠭⠤⠡⠆ (4)

In the French code the reader will still have to wait for reading the sixth symbol, the fraction bar to understand it is a fraction. Some other codes have implemented specific symbols to mark the boundaries of the fraction numerator and denominator, so the reader will know while reading the first symbol that it is a fraction. In the Nemeth code (5), the symbol ⠹ indicates clearly that it is a beginning of fraction.

⠹⠭⠬⠂⠌⠭⠤⠂⠼ (5)

## 2.4 Ambiguity of synthetic speech

In the case of speech, we have seen that the previous expression (1) would be spoken as : *"x plus one over x minus one"*. Unfortunately this sentence could be as well understood in 3 different meanings (6), (7), (8) [9] has showed, based on analysis of transcription of a number of Mathematical expressions, dictated to students who could not read them, the difficulty of understanding spoken maths.

$$x+\frac{1}{x-1} \quad (6) \qquad x+\frac{1}{x}-1 \quad (7) \qquad \frac{x+1}{x}-1 \quad (8)$$

There are two ways of dealing with this problem. The easiest is to add some explicit markers for blocks, speaking instead *"open parenthesis x plus one close parenthesis over open parenthesis x minus one close parenthesis"*. In that case there is no more ambiguity, but unfortunately the length increases dramatically, and it gets really hard when several levels of blocks are nested. The second way is based on prosody, which necessitates variation of voice tones and of silences between words. This is very promising, since with human speakers it would allow to increase the understandability of spoken mathematical expressions, but remains difficult to implement.

[7] has recently proposed a real break-through, based on audio object they call spearcons, constructed from extremely accelerated synthetic speech, and different tones showing the level of nested blocks. This proposal allows to considerably reduce the duration of sentences while reinforcing the understanding of the formulas.

### 2.5 Access to content

One other problem encountered by visually impaired people is linked to the limited availability of mathematical material they can have access to. First documents had to be transcribed manually to mathematical Braille notations. Since computers can be used, the first assistive technologies in this domain have consisted in producing automatic transcription software from mainstream formats to Braille. Unfortunately these software applications were linked to one specific mainstream format and one specific Braille code, for instance Labradoor [8] is an application converting from LaTeX to Marburg and Bramanet from MathML to French Braille, while no application could translate from LaTeX to French Braille.

In collaboration with an international group of experts of the fields, we have proposed a programming library, the *Universal Maths Conversion Library* (UMCL), which can be linked to any application to produce output in several Braille codes [2]. We have recently released an online converter[2] which allows to convert as well from MathML and from LaTeX to various national Braille mathematical notations [3].

### 2.6 Helping to understand and to do maths

Several projects have been aiming at making it easier to understand mathematical content rendered by these linear modalities, as well as to actually work on mathematical content, solve problems or perform calculations. These software allow to navigate within the structure of the formulas whether using speech [9] with the Math Genie, or with Braille in the MaWEn prototypes [5].

One of the interesting ideas in these software, especially on the Braille side, is that they give an opportunity to use the simple ASCII linearisation and to use the navigation within the content to deal with the length and complexity of expressions instead of reporting the difficulty to an extremely difficult format (the Braille specific notations for Mathematics).

## 3 Formats and input methods

As seen above a number of products and research prototypes have been designed to help visually impaired people with mathematical content. Beyond these, an increasing number of products are available, especially transcription tools.

To be able to use these applications, users need to get some mathematical content in a usable format. Several problems occur then. The choice of a relevant format first, and then problems due to the way the authors choose to design their documents using these formats. These problems are first of all faced by professional transcribers, who for instance have to transcribe school books for school pupils. End users who want to use help applications will have the same of problems too.

### 3.1 Images of expressions

If one searches for online content including mathematics, one will find much more documents containing expressions in pure graphical form, that is an image file of the formula. This is the worse possible way of putting Maths online since it is completely inaccessible for print impaired users and actually very poorly accessible for the mainstream.

$$\int_0^1 f(t)dt = \lim_{n\to+\infty} \sum_{k=0}^{n} \frac{1}{n} f\left(\frac{1}{k}\right) \tag{9}$$

$$\int_0^1 f(t)dt = \lim_{n\to+\infty} \sum_{k=0}^{n} \frac{1}{n} f\left(\frac{1}{k}\right) \tag{10}$$

The expression (9) was copied from an online university course. As these content are supposed to be viewed on a screen, the resolution is quite low. Actually even for a perfectly sighted person, it is very hard to read. We have asked some students to transcribe these formulas in MathML using a formula editor and they actually made a lot of mistake because they could hardly read the indices. On the example for instance the $k$ and the $n$, when in indices, were mixed up by students who did not have enough mathematical skill to perfectly understand the content.

Using a proper format (10) would solve the problem at least for sighted users. Nowadays, most Web browsers are able to display MathML expressions properly, MathJax[3] allows to do it on any browser.

### 3.2 What format for mathematical content

As mentioned in the introduction, the 2 main formats used to represent mathematical expressions are LaTeX and MathML. MathML would be more used at secondary level while LaTeX is still leader in Universities.

Unfortunately these two formats are based on a representation of the graphical layout of formulas. Actually in the case of MathML, there exist 2 variants : Presentation MathML and Content MathML, the latter coding the semantics instead of the way it looks. In reality the number of applications which are able to produce or to use Content MathML is close to zero, and basically only Presentation MathML is currently available.

One objective reason is that the semantic of some very common mathematical representation cannot be decided automatically. For instance in the expression $a(x+1)$, the identifier $a$ may be a coefficient to multiply with the sum $x+1$, but could also be a function $a$, that we want to apply to $x+1$. Without an understanding of the context it cannot be solved.

This is not a real problem for Braille users, since the same ambiguity remains in Braille, while it would be very useful for speech synthesis.

The next generation of Mathematical formats will be based on semantic (MathML 3 and OpenMath 3 are both semantic, and converging to each other). This will be possible due to the fact that most content will be designed directly in these semantic formats instead of translated from one format to another. But in fact it will take a very long time before these formats will be really used. Currently most maths software only use presentation MathML.

### 3.3 Inputting maths

A important issue is the way to input content that can be stored in these formats. For LaTeX the content is directly typeset in LaTeX by the designer. As LaTeX is a language for presenting documents, it is natural that its maths code is focusing on presentation.

The situation is different with MathML since, as a XML language, it is very difficult to type manually. Actually the authors will use software like MathType of OpenOffice.org to design the mathematical elements. OpenOffice.org has designed a quite convenient input method that can be used in conjunction to the menu based interface. Other tools like CMaths or DMaths plugins may help authors.

Here it must be remarked that these software solutions to design maths content are mostly used to design nice expressions, but rarely to work. Actually some visually impaired persons use LaTeX to work with maths despite the size of representation, because they don't have many choices, but the main use remains to author books, reports or curricula.

This input problem has some big consequences on the use of technology to work on maths. Most sighted persons would prefer to use a computer if they have to work on a text. Indeed there are lots of advantages that are clearly recognised : correction, possibility to move text, etc. When it comes to Maths, if one is asked to solve a problem, one will ask for a piece of paper ans a pen.

Another reason is the possibility that paper offers to easily draw graffiti around formulas, strike factors, write notes *etc.*

### 3.4 Multiple ways of representing the same concepts

Whether an expression is coded using LaTeX or MathML, many mathematical structures can be represented in several equivalent ways. A very basic example in LaTeX is about exponents and subscripts. When both are present the exponent can be put before or after the subscript, both are correct. For instance `$x^2_1$` and `$x_1^2$` are completely equivalent and would both display like: $x_1^2$. This is still true with more complex exponents like `$x^{p+1}_a$` and `$x_a^{p+1}$`, which would both display like: $x_a^{p+1}$.

It is easier to show an example of this situation using LaTeX, but similar things occur in MathML, and there are even much more ways of representing the same concepts.

This is not a problem when the objective is limited to print a mathematical expression or to display it on a screen, but when it comes to automatic transcription it increases dramatically the number of cases to be taken in consideration and makes the software more difficult to design and to maintain.

This is one of the reasons why we decided to definite a subset of MathML, which we called *Canonical MathML* [4], in which each mathematical structure can only be represented in a definite way. The canonisation is the most cost effective process. Indeed it takes usually about 90 % of the processing time when transcribing a MathML document to mathematical Braille.

### 3.5 Coding problems in Mathematical expressions

Let us now consider some coding problems that professional transcribers in Universities have quite often to face, and which are extremely time consuming since they have to fix these problems in order to be able to process a document. These problems are found in documents that are provided to the transcribers in digital format, by maths content authors: teachers, book authors, *etc.*. Then one would expect that transcription would be much easier in that case than when documents are provided with pictures representing the expressions. Actually it is not always the case and in a lot of times the transcribers have to fix the input file before they are able to work. Sometimes they have to re-design completely the maths expressions exactly as if they would have been provided as images.

*Note* : in the following examples will be given with LaTeX code because it is shorter to write and easier to show, but these issues can be found as well in LaTeX documents than in documents designed with a word processor and a formula editor, producing MathML output.

The following example illustrates 2 problems often found in Maths sources:

- Multiple expressions in a single mathematical object
- Maths style applied to a word or a group of words to put it in italics

In the following example `[Latex source 1a]` the two solutions of the quadratic equation are put together in the same maths object (between the characters $) and the word *"and"* in the middle is marked as text. It should be separated in two maths objects as shown in `[Latex source 1b]`.

`[Latex source 1a]`

```
The solutions from the quadratic equations
are, if the $discriminant~\Delta$
is positive, $\frac{-b+\sqrt{\Delta}}{2a}
~\text{and}~ \frac{-b-\sqrt{\Delta}}{2a}$.
```

`[Result]`

The solutions from the quadratic equations are, if the *discriminant* $\Delta$ is positive, $\frac{-b+\sqrt{\Delta}}{2a}$ and $\frac{-b-\sqrt{\Delta}}{2a}$.

Also the word *"discriminant"* is inserted in the first maths object in order to display it in italics. It should be separated and of course the italics should be put using the correct code. Each element of the document must be identified clearly.

[Latex source 1b]

```
The solutions from the quadratic equations
are, if the \textit{discriminant} $\Delta$
is positive, $\frac{-b+\sqrt{\Delta}}{2a}$
and $\frac{-b-\sqrt{\Delta}}{2a}$.
```

[Result]

The solutions from the quadratic equations are, if the *discriminant* $\Delta$ is positive, $\frac{-b+\sqrt{\Delta}}{2a}$ and $\frac{-b-\sqrt{\Delta}}{2a}$.

The next example illustrates some other problems also found in Maths sources:

- Expression split into several maths objects
- Expressions symbols not marked as maths objects in following explanation
- Maths style applied to a word or a group of words to put it in italics

In the `[Latex source 2a]`, the equation $E = \frac{1}{2}mv^2$ is split into 2 parts and the symbol '=' is put as a text. This will completely mess up a transcribing application. Each expression should be put in a single maths object. Another problem is that the two variables $m$ and $v$ are not identified as maths object in the following line, where their roles are explained. Furthermore the words *weight* and *speed* are identified as math objects only to set them to italics.

[Latex source 2b]

```
Let us consider the function\\
$E$ = $\frac{1}{2}mv^2$\\
where m is the $weight$
and v the $speed$...
```

[Result]

Let us consider the function
$E = \frac{1}{2}mv^2$
where m is the $weight$ and v is the $speed$...

The `[Latex source 2b]` shows how this example should be coded in LaTeX.

[Latex source 2b]

```
Let us consider the function\\
$E = \frac{1}{2}mv^2$\\
where $m$ is the \textit{weight}
and $v$ the \textit{speed}...
```

[Result]

Let us consider the function
$E = \frac{1}{2}mv^2$
where $m$ is the *weight* and $v$ is the *speed*...

## 4 Need for Maths Content Accessibility Guidelines

The recurrent accessibility problems faced by transcribers show that these problems are not understood by maths authors. In order to improve this situation we think that a document describing the accessibility guidelines for documents containing maths should be designed by an international group of experts.

It is right that most of these rules are actually already present in several document accessibility guidelines, like WCAG, in a more general form. Nevertheless these accessibility guidelines are too general and too large to be proposed to maths authors as is.

A shorter document, focused on problems found in Maths content would benefit to the community. Giving this document the status of a standard *de facto* by making it authored by an international work group of recognised experts, on the model of WCAG, would give this document more weight, and more chances to be followed by maths authors.

Here are a few items that could be taken as an input discussion paper.

- Never use an image file to put a mathematical expression;
- Never put multiple mathematical expressions in a single mathematical object;
- Never split an expression into several maths objects
- All expressions variables must be identified as maths objects, even when in explanation following an expression
- Maths style should not be applied to simulate italics

As mentioned a lot of applications are using MathML as a *de facto* standard. A second level of guidelines, on the model of the ATAG (*Authoring Tools Accessibility Guidelines*) from the W3C, would help to avoid the problems mentioned in subsection 3.4, about multiple ways to represent the same concepts.

## Notes

[1] See `http://chezdom.net/mathematicalbraillecodes`
[2] See `http://chezdom.net/umcl`
[3] See `http://www.mathjax.org`

## References


[1] Dominique Archambault. Non visual access to mathematical contents: State of the art and prospective. In *Proceedings of the WEIMS Conference 2009 (The Workshop on E-Inclusion in Mathematics and Science)*, pages 43–52, 2009.

[2] Dominique Archambault, Donal Fitzpatrick, Gopal Gupta, Arthur I. Karshmer, Klaus Miesenberger, and Enrico Pontelli. Towards a universal maths conversion library. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Dominique Burger, editors, *Proc. ICCHP 2004 ($9^{th}$ International Conference on Computers Helping People with Special Needs)*, volume 3118 of *LNCS*, pages 664–669, Berlin, July 2004. Springer.

[3] Dominique Archambault and Franck Guyon. Umcl transcription tools. In Gert Jan Gelderblom, Mathijs Soede, Leon Adriaens, and Klaus Miesenberger, editors, *Proc. of AAATE 2009 (Everyday Technology for Independence and Care - AAATE 2011)*, pages 416–423, Maastricht, The Netherlands, August 2011.

[4] Dominique Archambault and Victor Moço. Canonical mathml to simplify conversion of mathml to braille mathematical notations. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur I. Karshmer, editors, *Proc. ICCHP 2006 ($10^{th}$ International Conference on Computers Helping People with Special Needs)*, volume 4061 of *LNCS*, pages 1191–1198, Linz, Austria, July 2006. Springer.

[5] Dominique Archambault, Bernhard Stöger, Mario Batušić, Claudia Fahrengruber, and Klaus Miesenberger. A software model to support collaborative mathematical work between braille and sighted users. In *Proc. ASSETS 2007 (Ninth International ACM SIGACCESS Conference on Computers and Accessibility)*, pages 115–122, Tempe, AZ, USA, October 2007. ACM SIGACCESS.

[6] Dominique Archambault, Bernhard Stöger, Donal Fitzpatrick, and Klaus Miesenberger. Access to scientific content by visually impaired people. *Upgrade*, VIII(2):29–42, April 2007. Digital journal of CEPIS.

[7] Enda Bates and Dónal Fitzpatrick. Spoken mathematics using prosody, earcons and spearcons. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur I. Karshmer, editors, *Computers Helping People with Special Needs, 12th International Conference, ICCHP 2010, Vienna, Austria, July 14-16, 2010. Proceedings, Part 2*, volume 6180 of *Lecture Notes in Computer Science*, pages 407–414. Springer, 2010.

[8] Mario Batušić, Klaus Miesenberger, and Bernhard Stöger. Labradoor, a contribution to making mathematics accessible for the blind. In Alistair Edwards, Andras Arato, and Wolfgang Zagler, editors, *Proc. ICCHP 98 ($6^{th}$ International Conference on Computers Helping People with Special Needs)*, Oldenbourg, Wien, München, 1998.

[9] Douglas Gillan, Paula Barraza, Arthur Karshmer, and Skye Pazuchanics. Cognitive Analysis of Equation Reading: Application to the Development of the Math Genie. In Joachim Klaus, Klaus Miesenberger, Wolfgang Zagler, and Dominique Burger, editors, *Proc. ICCHP 2004 in Paris (International Conference on Computers Helping People with Special Needs)*, volume 3118 of *LNCS*, pages 630–637, Linz, Austria, July 2004. Springer.

[10] Arthur Karshmer and Douglas Gillan. How well can we read equations to blind mathematics students: some answers from psychology. In Constantine Stephanidis, editor, *Universal Access in HCI – Inclusive Design in the Information Societ*, volume 4, pages 1290—1294, Mahwah, New Jersey, USA, 2003.