Audio Rendering of Mathematical Formulae using MathML and AudioMath

Helder Ferreira, Diamantino Freitas
Faculty of Engineering University of Porto
DEEC / LPF-ESI / Portugal
hfilipe@fe.up.pt, dfreitas@fe.up.pt

Abstract

Technical, scientific or even simple documents presented online that involve mathematical expressions, issue a big problem with accessibility regarding visually impaired persons. One possible solution is to parse and interpret the mathematical contents and convert them into an audio format. This is a complex and multidisciplinary problem, which *AudioMath* [1], a work in progress at LPF-ESI [2], studies and aims to contribute to solve.
AudioMath provides conversions from mathematical expressions in W3C's MathML [3][4] format into plain text, something that a capable text-to-speech (TTS) engine would "understand" and read-out.
This paper reviews the state of the art in publishing mathematical documents in the Internet, and introduces the *AudioMath* prototype and its working mechanisms, as an accessibility tool for use together with a TTS engine or to use as a standalone application, with great advantages for visual impaired persons. The main scientific and technical challenges, specifically at the levels of interpretation of the content of the MathML coded expressions and conversion into spoken form, are considered and the current status of the work is described.

Keywords: Accessibility, Audio Rendering of Mathematical Expressions, Text-to-Speech, MathML, Conversion of mathematical formulae into text.

1. Publishing scientific documents with mathematics in the Internet

"*How hard can it be to communicate about math on the Internet? The truth is, it's a fairly difficult task.* [5]". The publication of scientific documents containing mathematical formulae is extremely demanding. The appearance of the TeX [6] system developed by Donald Knuth solved the majority of problems with printed documents. Then WYSIWYG (*What You See Is What You Get*) editors, and subsequently mark-up languages, for Internet appeared, such as HTML (*Hypertext Mark-up Language*). However, HTML *per se* doesn't allow the use of a mathematical description language directly into the document. So being, alternatives were developed, as described in the following.

1.1. Images containing mathematical contents

This is the approach used by software like LaTeX2HTML [7] and TeX4ht [8]. The process consists in converting TeX or LaTeX documents to HTML, where the mathematical contents are converted into GIF objects. There are two main disadvantages in this process: the low resolution of images and the creation of an accessibility problem, once these images will have to contain a text description (ALT) [9] that allows the mathematical expression to be read-out, otherwise the only possibility is to interpret the image contents. The main advantage is that the user doesn't need to install any special plug-in because the browser can present the information easily.

1.2. Alternative formats, such as: PDF, TeX, Postscript, Word or RTF

Most of the times, documents containing mathematical expressions are saved in formats such as: PDF (*Portable Document Format*), TeX, Postscript, Microsoft Word or RTF (*Rich Text Format*), and placed as links in online documents. The visualization of those documents is only possible if the browser provides plug-ins for each type of format, or if the user edits the document after downloading it. There are two main disadvantages in the use of these alternative formats: the need to have a plug-in installed, and the lack of accessibility in reading those documents, once they open inside the browser. However some efforts have been made to provide more accessibility features; this is the case of Adobe PDF [10]. The main advantage of this approach resides in the fact that these formats are widely spread and most users or browsers support them for visual rendering.

1.3. Applets to create graphical representations of mathematical contents

This approach is used by WebEQ [11], which consists in an applet server that receives a mark-up language defining a mathematical expression. Currently, WebEQ understands WebTeX (mark-up language that can be included in a HTML document). This approach has the same disadvantages and advantages than the use of images to represent mathematical content. However, the applet download makes the rendering a little slower.

1.4. Plug-ins to create visual rendering of mathematical contents

The use of plug-ins is identical to the use of Java applets; however, using a plug-in requires that the user has already installed it in the browser, since the application runs at the client instead of at the server. TechExplorer [12] from IBM and MathPlayer [13] from Design Science are examples belonging to this approach. Both applications have some support to W3C's MathML language.

1.5. HTML and Symbol Fonts

This approach supplies the simplest documents; however, it compromises the aesthetic form of the document. Translator TtH [14] (TeX to HTML) is an example. It uses special HTML supported fonts and symbols to represent mathematical glyphs, and HTML mark-up to structure the mathematical expression representation. The main advantages are simplicity and swiftness of download. The downsides are the accessibility limitations introduced, since the mathematical expression is geometrically structured using tables [9].

1.6. Use of XHTML and CSS

This is a relatively recent approach to render mathematical expressions online. The main disadvantages are: the rather complex learning of the mark-up languages, and the fact that not all the browsers support XHTML [15] and CSS (Cascading Style Sheets) [16]. As a main disadvantage, the range of mathematical expressions capable to be expressed in XHTML and CSS is limited. The main advantages are: simplicity (once you know the mark-up), swiftness in rendering and the fact that the user doesn't need any additional plug-in in the browser.

1.7. Mark-up languages such as: MINSE, MathML, SVG and others

Currently there are several mark-up languages that allow the publication of scientific documents in the internet; however the majority are still in a development phase. Some examples of mark-up languages that can be used to publish mathematical contents in the Internet are: MINSE [17], MathML, SVG [18], WebTeX, TeX and HyperLaTeX [19]. The future of the publication of scientific and technical documents resides in these mark-up languages, specially, in MathML. However there's still a long way to go, once not all the browsers support them.

1.8. Creating the documents

Several authoring software packages allow a relatively easy composition and edition of mathematical expressions for inclusion in web documents:

- MathType, from DesignScience, an evolution of Microsoft Equation Editor, produces documents in HTML+GIF or HTML+MathML formats. The MathPlayer plug-in is required for visualization. The code produced is MathML Presentation Markup [4].
- WebEQ also from DesignScience is composed of two packages, WebEQ Editor and WebEQ Publisher. They produce documents in the formats HTML+GIF, HTML+Java Applet or HTML+MathML. This code is either MathML Presentation Markup or MathML Content Markup [4].
- Mathematica and Publicon, from Wolfram Research, produce documents in the formats XHTML+MathML, XML (NotebookML), XML (NotebookML+MathML), TeX, HTML or TechExplorer. The resulting code is MathML Content Markup.
- Scientific Word, from Mackichan Software, produces documents in the formats HTML+MathML and LaTeX. The output code is MathML Presentation Markup.
- Amaya, from W3C is a free browser with built-in editor producing documents in the formats HTML, XHTML, XML, Text, MathML, CSS and SVG. The output code is MathML Presentation Markup.
- EZMath from David Ragget, is a free mathematical expressions editor with a special notation and browser plug-in. Output code blocks can be embedded in HTML and XHTML documents. The code is MathML Content Markup.
- For TeX or LaTeX users there are tools like:
  - Itex2mml that converts ITeX coded formulas into XHTML+MathML and the output code is MathML Presentation Markup.
  - TeX2MML, an online converter from LaTeX to MathML Presentation Markup.
  - TeX4ht that converts LaTeX and TeX into HTML. By default produces GIF images for formulas, but can also produce HTML+MathML in the MathML Presentation Markup.
  - TtM converts LaTeX or TeX documents in HTML+MathML and the code is MathML Presentation Markup.
  - LaTeX2HTML converts LaTeX into HTML.

As can be seen the MathML Presentation Markup dominates relatively to the MathML Content Markup, amongst authoring software.

2. The AudioMath Project

2.1. Introduction

The *AudioMath Project* is an initiative of the Laboratory of Speech Processing, Electro Acoustics, Signals and Instrumentation. The main aim of this project is to produce a tool,

either standalone or for integration in a TSS engine, that does the parsing, interpretation and conversion into European Portuguese (EP) plaintext form, of text or mark-up elements (MathML) not directly "understandable" by a regular TTS engine. This tool, under development, is called **AudioMath** and is, as far as we know, the first published application in the world that speaks mathematical expressions in the *MathML Presentation Mark-up* format in European Portuguese.
In its current form *AudioMath* is an ActiveX *dynamic link library* (DLL) that can be used by any program through internal calls. It has been developed in Perl and Microsoft Windows 9x/ME/2K/XP. The tests and refinement were made with help of a specifically developed console (see in figure 1 a screenshot of the program GUI). Text or text files can be retrieved or an Internet web page be opened for reading, and be subsequently processed, converted and read through the selected TTS engine.

Figure 1 - Screen shot from AudioMath Test and Development Application.

AudioMath can be used mainly for:

- Reading of technical and scientific documents online in an accessible way, with particular benefit for vision impaired persons.
- Teaching or learning how to read mathematical formulae [20].
- Enhancing general accessibility to computer-based applications, when applied to a TTS engine.

2.2. MathML - Mathematical Mark-up Language

MathML is an XML application for the publication of mathematical contents over the Internet. This mark-up language has a simple and concise syntax that codes either the notation (*MathML Presentation Mark-up*) or the meaning (*MathML Content Mark-up*) of a mathematical expression. The *AudioMath Project* has chosen MathML for the following reasons:

- The mark-up language was developed by W3C and will eventually become an ISO standard.
- A rapidly growing use by several relevant organizations associated with the teaching and learning of mathematical contents, such as the *American Mathematical Society* [21] and *The OpenMath Group* [22], as well as the involvement of software houses like *Design Science*, *HP, IBM*, *Microsoft* and *Wolfram Research*.
- Emergence of editors and applications that create and manipulate MathML documents.
- Existence of conversion tools for the main publishing formats.

- The fact that it is a mark-up language allows its parsing, interpretation and conversion to other formats, and consequently a higher accessibility, portability and platform independence.

2.3. MathML Presentation Mark-up vs. MathML Content Markup

The representation of mathematical formulae and other mathematical contents is perceived by two distinct aspects: the visual structure or notation of the mathematical expression, and the concept or meaning that it conveys [23]. For example: the same concept of "*division of a by b*" can have the different notations: a/b, $a÷b$, $\frac{a}{b}$ or ab^{-1}. The mapping from notation into concept can be ambiguous as well, for instance, the notation A_2^3 can have the meanings: "*square of A_2*" or "*permutations of 3 elements 2 at a time*".
The two sets of MathML mark-up address separately both modalities:

- *MathML Presentation Mark-up:*
  - Aimed to the visual presentation of a mathematical expression
  - Doesn't differentiate the meaning of the mathematical expression
  - Not intended to (but possible) audio rendering of mathematics
  - The conversion of Presentation Mark-up into Content Mark-up is not advised.
- *MathML Content Mark-up*:
  - Aimed to represent the meaning of a mathematical expression
  - Used to transfer MathML between applications
  - Doesn't differentiate the mathematical notation
  - Ideal for audio rendering of mathematical representations
  - Can be converted into Presentation Mark-up
  - Limited only by the mathematical operators and functions it supports.

Although *Content Mark-up* is recommended to be used in audio rendering of mathematical expressions [24] (once it preserves the meaning of the formulae) it has some limitations concerning the list of mathematical operators and functions it supports presently. The OpenMath Group, for example, offers a lot bigger dictionary. Moreover, most current editors are WYSIWYG and code the mathematical contents into *Presentation Mark-up* instead of *Content Mark-up*. Therefore the online documents that contain MathML expressions are almost all coded in *Presentation Mark-up*. For these reasons, *AudioMath* selected *MathML Presentation Mark-up*. However this introduces several difficulties to the interpretation of the formulae ambiguities. These difficulties can, in principle be overcome by contextual information or "intelligence".

2.4. The AudioMath Process

AudioMath was built in a modular, extensible and configurable architecture, in Perl. Therefore, the support for new target languages, the update of dictionaries and of algorithms can be done quite easily. Currently only the European Portuguese language is supported..

2.4.1. Text Analysis

A technical document can contain several elements such as: acronyms, abbreviations, numbers and similar expressions and mathematical expressions. Also, it is probably to come with special Unicode characters and math glyphs. Therefore the first step is to clean up the text, converting Unicode to Latin1 (in this case to support European Portuguese characters). Next step is an auto-discovery process that recognizes types of elements and makes calls to the modules that convert the element into a full written form. For example: if "det." is detected, it should be converted into "*determinant (of)*". To speed up the process the document should be divided into blocks of text, splitting the MathML Mark-up from the rest of the text.

2.4.2. Parsing and Interpreting MathML

The MathML Markup is parsed using the Perl module: *XML::Parser* [25] which acts as a SAX parser type (event-based), supporting Encoding ISO-8859-1 and discarding XML Namespaces.
Since *AudioMath* uses *MathML Presentation Mark-up*, a big effort and computation is needed on the interpretation of the mathematical expression. This is done using a process of raising flags each time a starting or ending tag is detected. By knowing the history of the mark-up the function or operation under parsing can be guessed and therefore information retrieved to understand the structure of the math expression and do its conversion.

2.4.3. Converting and Speaking Mathematical Contents

"*No standard protocol exists for articulating mathematical expressions as it does for articulating the words of an English sentence.*" [26]. Except for a few significant works [27] [28], currently there is an almost complete lack of studies or research on how should a mathematical expression be read. The authors don't know any work of the kind for Portuguese.
The conversion of the MathML tree to text form is done according to a database of rules that was built based on a collection of materials written by experienced professors.
Note that the bigger the expression the more complex the interpretation and conversion process become. The same happens during audio output time. The listener will hardly handle a full complex formula. A solution would be to introduce navigational mechanisms to browse inside the expression.
The objective of automatically speaking mathematical contents has to deal besides the non-trivial issues of text generation and phrasing, with the generation of the prosody to impose over the synthetic speech.
For example, consider the simple mathematical expression: $\sqrt{a^2+b^2}$. This could be rendered more or less ambiguously as:

- Square root of a squared plus b squared, end of radicand.
- Square root of a taken to power two, plus, b taken to power two, end of radicand.
- Square root of power base a exponent two, end of power, plus power base b exponent two, end of radicand.

Which of these forms is more correct, not ambiguous and efficient? Now consider the following expressions: $\sqrt{a^{2+b^2}}$, $\sqrt{(a^{2+b})^2}$ and $\sqrt{(a^2+b)^2}$. Taking in account the text forms presented before, how should we read these expressions?
One could do the experience of speaking the texts monotonically to someone that is not looking to the expression and ask for a written version after the dictation. If one is not careful enough they'll all sound much a like, and quite ambiguous, so identification of the right formula them can easily become very difficult.

The solution must pass by the adoption of formal ways of text generation that keep the right structure information of the formula.
How do things happen in speech communication then? Let's go back to the first example of text rendering of the first expression and consider the sample spoken version, whose waveform, f0 contour (intonation) and text labelling are depicted in figure 2, from top to bottom (the picture was obtained with the PRAAT [29] software).

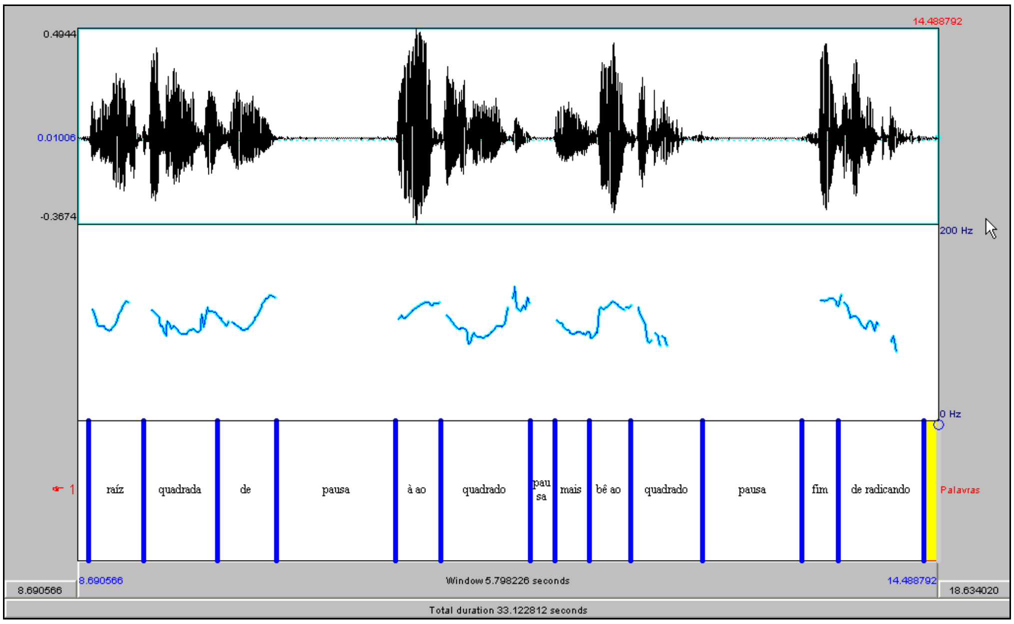

Figure 2 - "Square root of a square plus b square, end of radicand" in European Portuguese.

As can be seen in figure 2, there are two distinct large pauses and a smaller optional pause between *"á ao quadrado=a squared"* and *"mais bê ao quadrado=plus b squared*".
Another immediately apparent aspect in the middle waveform is the existence of rising and falling movements of f0 in the speaker's intonation intended to provide classification of the boundaries introduced by the pauses. A rising tone is used when a lower hierarchical level is starting (see at the end of *"quadrado de"*) and a falling tone is used when this level is ended (see at the end of *"bê ao quadrado"*) while a rising tone associated to a down tone are used to classify the smaller separating pause between identical elements, so indicating a continuation. Finally the emphatic falling f0 movement at *"fim de radicando=end of radicand"* signals the end of the expression.
The rules already defined in the present development phase of the project are implemented at conversion time by tagging the text with prosodic marks, to command the TTS device in order to produce the required pauses and f0 modulations.
Although this research is being carried out for European Portuguese, the authors believe that it can be substantially extrapolated to many other languages.

3. Conclusions and Trends of Future Work

Providing mathematical contents on-line, that is, in digital format, allows a greater access to information than before. However, problems do exist in reading those contents, in particular for blind persons that don't have access to graphical representations. It is known that the difficulties for blind persons rise and accessibility diminishes with the increase of technical level in documents [30]. More efforts are therefore needed in order to provide more means to achieve the full accessibility of technical documents in the Internet. *AudioMath* is, without

doubt for the authors, a valuable contribution to the increase of accessibility in reading on-line documents with mathematical contents.
However it is a work in progress and plenty still needs to be done, such as: - complete the support to *MathML Presentation Markup* and add support to *MathML Content Markup*; - develop modules that support HTML, XHTML, XML [31] and SSML [32]; - add support for new languages; - continue to develop the study on the prosody of reading mathematical formulae; - provide mechanisms for navigating inside more complex mathematical formulae.

References

[1] AudioMath for European Portuguese – http://lpf-esi.fe.up.pt/~audiomath .
[2] LPF-ESI – Laboratory for Speech Processing, Electro acoustics, Signals and Instrumentation at the Faculty of Engineering University of Porto – http://lpf-esi.fe.up.pt .
[3] W3C – World Wide Web Consortium – http://www.w3c.org .
[4] MathML – Mathematical Markup Language – http://www.w3.org/Math .
[5] Math Typesetting for the Internet. The Math Forum - http://mathforum.org/typesetting .
[6] TeX Users Group (TUG) Homepage - http://www.tug.org .
[7] Latex2HTML Homepage - http://saftsack.fs.uni-bayreuth.de/~latex2ht/ .
[8] TeX4ht Homepage - http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html .
[9] Web Content Accessibility Guidelines 1.0, W3C - http://www.w3.org/TR/WCAG10/ .
[10] Accessibility in PDF - http://www.adobe.com/products/acrobat/access_overview.html .
[11] WebEQ Homepage - http://www.dessi.com/en/products/webeq/default.htm .
[12] TechExplorer Homepage - http://www-3.ibm.com/software/network/techexplorer .
[13] MathPlayer Homepage - http://www.dessi.com/en/products/mathplayer/ .
[14] Translator Tth Homepage - http://hutchinson.belmont.ma.us/tth .
[15] XHTML 1.1 - http://www.w3.org/TR/xhtml11/ .
[16] CSS 2.0 - http://www.w3.org/TR/REC-CSS2 .
[17] The MINSE Project - http://lfw.org/math/top.html .
[18] SVG - Scalar Vector Graphics - http://www.w3c.org/Graphics/SVG/ .
[19] The HyperLatex Package - http://www.cs.uu.nl/~otfried/Hyperlatex .
[20] Freitas, Diamantino. Ferreira, Helder. Et al. *A prototype application for helping to teach how to read numbers*. HCII2003 Proceedings. Volume I.
[21] AMS - American Mathematical Society - http://www.ams.org .
[22] The OpenMath Group – http://www.openmath.org/coccon/openmath/index.html .
[23] Pierce, John. *An Introduction to Information Theory. Symbols. Signals and Noise.* Dover Publications Inc. New York. 1961.
[24] Sandhu, Pavi. *The MathML Handbook.* Charles River Media. 2002.
[25] XML::Parser - http://search.cpan.org/dist/XML-Parser/Parser.pm .
[26] The MathSpeak Project - http://www.rit.edu/~easi/easisem/talkmath.htm (by A. Nemeth).
[27] Karshmer, Arthur. *How Well Can We Read Equations to Blind Mathematic Students?* HCII2003 Proceedings. Volume 4.
[28] T.V. Raman. *Audio System For Technical Readings*. Dissertation to the Faculty of the Graduate School of Cornell University. 1994.
[29] PRAAT: Doing Phonetics by computer, http://www.fon.hum.uva.nl/praat/.
[30] Monaghan, Alex. Fitzpatrick, Donal. *Browsing Technical Documents: Document Modelling And User Interface Design*. BULAG 1999.
[31] Mark-up languages from W3C used to publishing documents in Internet.
[32] SSML - Speech Synthesis Markup Language – http://www.w3.org/TR/speech-synthesis .