Math-on-the-web Platform for Arabic Mathematical Notation

Mustapha EDDAHIBI, Azzeddine Lazrek, Ali Rachidi

Abstract— The purpose of this work is to help overcoming some of the issues associated with the typesetting, encoding and rendering mathematical expressions in Arabic notation. Indeed, we propose here a multi-platform solution to display Arabic mathematical expressions encoded using one of the three most used formats for coding mathematical expressions, namely, MathML, TEX and AsciiMath. This solution is based on web fonts or image fonts to provide a visual rendering that respects both Arabic and mathematical typographical rules regardless of the used browser.

Index Terms— Math-on-the-web, Arabic mathematical notation, MathML, TeX, AsciiMath, bidirectionality, cursivity, web fonts.

IJSER

1 Introduction

One of the most important activities in IT is associated with scientific papers typesetting. Actually, the digital representation of such documents is intended to facilitate the creation, modification, processing, archiving and exchange through communication networks. However, reaching this goal requires overcoming difficulties due in part to mathematical expressions' nature that is different from plain text [1]. The degree of difficulty increases more if mathematical expressions are in Arabic notation. This difficulty is due to the spreading direction of the expressions, the cursivity of the writing and the use of special notations in Arabic mathematical notation. Some works previously carried out by the authors of this paper, concerning firstly the study of the Arabic mathematical notation to identify the difficulties that may hinder the use of conceptual and material tools for the composition, publication and processing of mathematical expressions in this notation. Another part of the work aimed to identify new concepts necessary for encoding Arabic mathematical notation [2] to adopt in specialist standards in this field, whether TeX[3] or MathML[4]. We have created a set of IT solutions for the composition and display of mathematical expressions in Arabic notation. Among these solutions, RyDArab a TeX-based system, Dadzilla [5] a web browser to display MathML in Arabic notation and ArMathMLEd an Arabic MathML WYSIWYG editor. These IT solutions require minimal IT knowledge to install and use. Some of these systems are not cross-platform and have compatibility issues with different versions of the same operating system. For some tools the user needs to learn mathematical expressions coding languages that are either TeX or MathML.

In this paper, we will present a new solution for displaying mathematical expressions in Arabic notation using universal mathematical encodings, namely TeX, and MathML and asciimath. Before presenting this solution, let's see what are the characteristics of the Arabic mathematical notation?

2 Arabic Mathematical Notation

There are two types of notations for mathematical expressions in Arabic. The first notation is similar to that in French or English, depending on the cultural influence. The direction of the writing of the symbolic component is contrary to the natural direction of the Arabic script and contains Latin symbols and Arabic text. This notation is called "mixed mathematical notation".

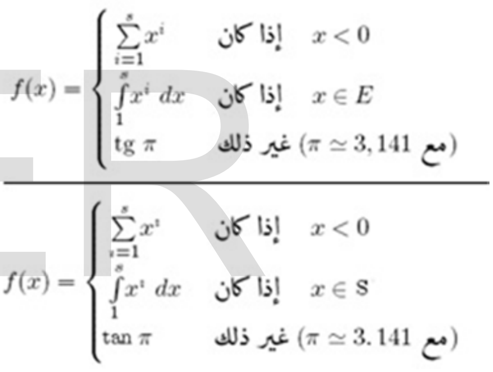


Fig.1. Mathematical expression in mixed notation

The second notation respects the spreading direction of Arabic writing from right to left. It uses specific symbols based on the Arabic alphabet, Latin reflected symbols and Arabic as well as Hindu numbers. This notation is called pure Arabic notation.

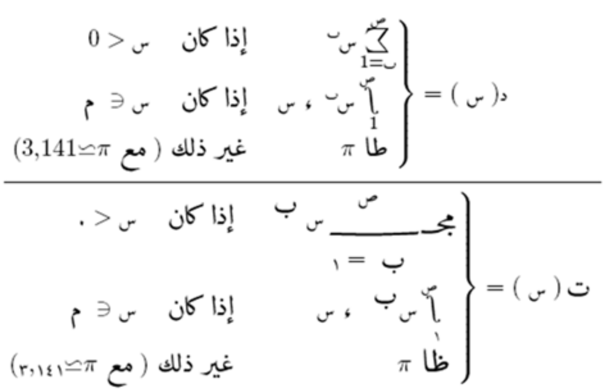


Fig.2. Mathematical expression in Arabic pure notation

3 Arabic mathematical documents' composition

Till Now, many Arabic mathematical documents are still written by hand. Such is the case even for a lot of e-documents which are digitalized images of handwritten documents. To remedy this situation, several scientific research projects focused the development of new methods, the adaptation of well known standards and the implementation of IT solutions to encode and display mathematical documents. RyDArab system was the first step that has paved the way for all other works in this field. The second step was the creation of Dadzilla, a web browser that displays mathematical expressions in Arabic notation.

3.1 RyDArab and TeX

Rydarab [6] is an extension of the system TeX that is used to compose mathematical expressions in Arabic notation. It is based on a set of fonts that contain variants of Arabic notation symbols. For the ligatures and the right-to-left of Arabic text, it uses one of the three well-known systems arabtex, omega and Arabi. Rydarab uses the same TeX commands as those used for the Latin notation e.g. \sum, \sqrt, \frac etc.. For the composition of the text and alphabetic symbols, RyDArab uses transtec transliteration. The expressions are generated in a high typographical quality especially when combined with Curext, a system for stretched symbols and kashida..

3.2 Dadzilla and MathML

IJSER

MathML is an XML extension dedicated to encode either presentation or semantic content of mathematical expressions. The third version of the standard MathML has added new recommendations for the presentation of mathematical expressions in Arabic notation. Among the guidelines of this version, for instance the use of the dir attribute with the value rtl to change the display direction of the mathematical expressions and to reflect glyphs of certain symbols as the sum, square root, parentheses, etc.. It recommends also the use of "initial", "tailed", "looped" and "stretched" variants for alphabetic symbols. It proposes the use of "madruwb" value for the notation attribute of the menclose element.

Dadzilla is a browser specially created to analyze potential problems when using MathML for the presentation of mathematical expressions in Arabic notation. The goal is to identify implementation challenges and the limitations of the MathML specification. Dadzilla is based on the Gecko rendering engine of Mozilla. The 1.1 version of Dadzilla is the most compliant to MathML 3 that runs on Microsoft Windows. We have created a WYSIWYG editor for the composition of mathematical expressions in Arabic notation. This editor is a Dadzilla Add-on, based on XUL and JavaScript.

After the announcement of the official release of the recommendation MathML 3 in October 21, 2010, several composition and display tools support this version of MathML like FireFox, MathType etc.. The support of the Arabic notation is in progress.

4 Math-on-the-web Platform

The method we present in this work is based on MathJax [7], an open-source JavaScript display engine for LaTeX, MathML, and AsciiMath notation. The mathematical expressions are processed using JavaScript to produce HTML, SVG or MathML equations. The result is that mathematical expressions can be viewed in any modern browser. Neither setup nor install is required on the part of the user. The latter does not have to download any plug-in. Authors of mathematical web pages can write their documents and be confident that all users can view it effortlessly.

In this work we propose a platform that uses JavaScript to convert the source coding of mathematical expressions into HTML and CSS. The transformation into MathML code is not used because the browsers that support MathML do not provide full support to the display of expressions in Arabic mathematical notation. Therefore, mathematical expressions are displayed using HTML and CSS. Knowing that this solution is based on web fonts, it was inevitable to create a set of Web fonts [8] in the formats needed across web browsers.

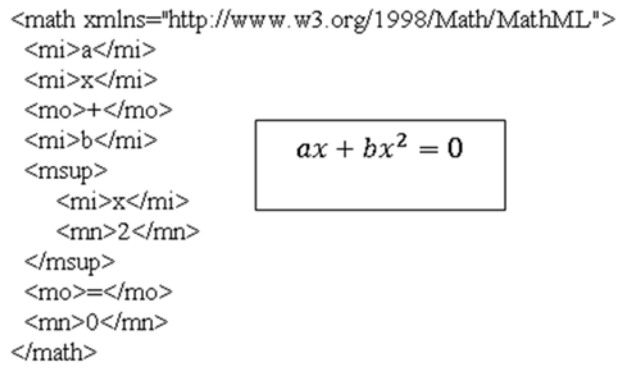
1. Web Open Font Format (WOFF)
2. Embedded OpenType (EOT)
3. OpenType Font
4. Scalable Vector Graphics (SVG)
5. Image fonts (PNG)

The PNG format is used in the case of browsers that do not support web fonts. The created fonts are divided into two main classes: one class for inverted Latin symbols and the other for Arabic alphabetical symbols. There are three types of encoding for mathematical expressions, MathML, TeX and AsciiMAth. Each type of coding requires a specific script to ensure its display. In the following subsections, we give some problems and their solutions within each of the three encoding languages.

5 Math-on-the-web and MathML

5.1 Right-to-left rendering

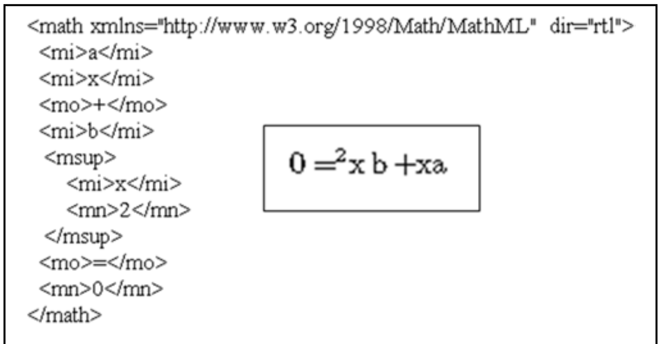
In MathJax, the right-to-left rendering of mathematical expressions in Arabic notation requires special care. Indeed, version 3 of the MathML standard recommends the use of the *dir* attribute with the value *rtl* in the math element to reverse the overall directionality of the mathematical expression [4]. For example, by adding the *dir* attribute with the value *rtl* to the following mathematical expression (fig.3.), we obtain the result in fig.4.



```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mi>a</mi>
  <mi>x</mi>
  <mo>+</mo>
  <mi>b</mi>
  <msup>
     <mi>x</mi>
     <mn>2</mn>
  </msup>
  <mo>=</mo>
  <mn>0</mn>
</math>
```

$$ax + bx^2 = 0$$

Fig.3. : Left-to-right mathematical expression

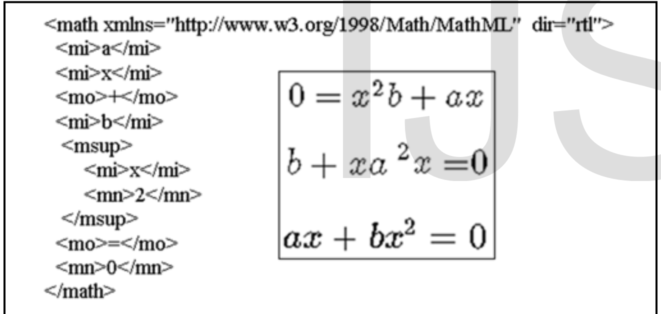


```
<math xmlns="http://www.w3.org/1998/Math/MathML"  dir="rtl">
  <mi>a</mi>
  <mi>x</mi>
  <mo>+</mo>
  <mi>b</mi>
   <msup>
     <mi>x</mi>
     <mn>2</mn>
   </msup>
  <mo>=</mo>
  <mn>0</mn>
</math>
```

Fig.4.: Right-to-left mathematical expression

In MathJax, MathML code is transformed to HTML for the display of mathematical expressions in browsers that do not support MathML or SVG natively. The mathematical expression is treated like a puzzle where each piece is an HTML element "*span*". Each span will be put in a precise position thanks to the CSS positioning attributes. Consequently, to display right-to-left mathematical expressions, a first solution is based on the use of the attribute *dir*= "*rtl*" in the HTML output code. This solution, which is used in the mml3 experimental extension of MathJax's 2.5 version, may not work for all browsers. Indeed, every browser behaves differently from others according to its interpretation of the dir attribute. The following figure shows the result of using the mml3 extension on some of the most used browsers.

IJSER

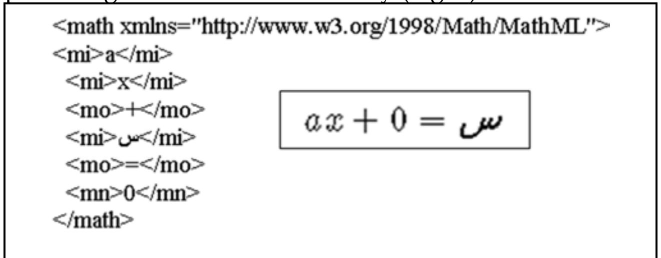


```
<math xmlns="http://www.w3.org/1998/Math/MathML"  dir="rtl">
  <mi>a</mi>
  <mi>x</mi>
  <mo>+</mo>
  <mi>b</mi>
   <msup>
     <mi>x</mi>
     <mn>2</mn>
   </msup>
  <mo>=</mo>
  <mn>0</mn>
</math>
```

Fig.5.: Display of HTML output code in some widespread browsers with the use of the *dir=''rtl''* attribute

To avoid the problems described in the previous paragraph and illustrated in Figure 6, we opted for another solution that is based on the inversion of the order of the tags in the output encoding. By reversing the order of output tags, the order of elements that compose the mathematical expression is reversed. HTML coding output of the mathematical expression (Fig.6.) will be reordered so that the tags would allow the display of MathML <*mi*> *0* </ *mi*> will be placed at the beginning of the output code, followed by those which correspond to <*mo*> = </ *mo*> and so on until the beginning of the MathML code of the expression. The values of the position attributes will not be modified. Hence, the same heights and the same sizes are used for left-to-right and right-to-left display, yet only the left is reversed to right and vice versa. By using this method the display is the same as in Fig.3..

Another problem is related to the use of both Arabic alphabetic symbols with high r-t-l directionality [9] and numbers or arithmetic operators with low directionality. The result of this mix is that low directionality characters inherit the preceding characters directionality (Fig.6.).



```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mi>a</mi>
  <mi>x</mi>
  <mo>+</mo>
  <mi>س</mi>
  <mo>=</mo>
  <mn>0</mn>
</math>
```

Fig.6.: directionality inheritance problem

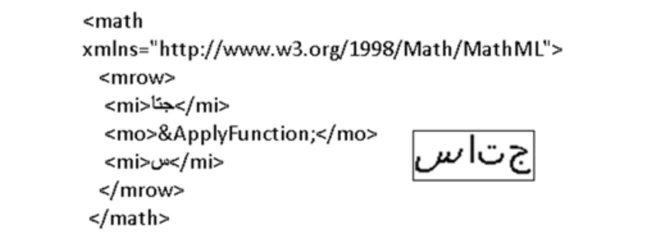
The challenge faced is to correctly display Arabic text contained in leaf elements such as "*mi*" or "*mtext*" without affecting the display of neighboring tags and therefore the display of the mathematical expression. Two solutions having the same principle are possible. There principle is the use of the bidirectional isolation [10]. This can be performed either by using the HTML element "*bdi*" or use of the *unicode-bidi* CSS property. Knowing that the "*bdi*" tag is not supported by several browsers, we opted for the second solution. So, the *unicode-bidi* property is used with the "isolate" value preceded by different vendor prefixes.

5.2 Text Cursivity

Arabic writing is cursive. Indeed, in every word, the neighboring letters are linked. According to its position in the word, a letter takes one of the four following forms:

1. Isolated: the letter is not attached to any other letter
2. Initial: the letter is attached only to its following letter
3. Final: The letter is attached only to its precedent letter
4. Median: The letter is attached to two other letters one on its right and the other on its left

In MathJax, each character is treated with special attention because of the nature of mathematical expressions compared with plain text. Indeed, mathematical expressions may contain, among others, some extensible symbols [11] such as square root, parentheses, braces etc. To stretch these symbols more than one character are used. For this reason, it was natural to organize characters in more than one list in MathJax. As these lists do not contain Arabic characters, the Arabic text is divided into characters and each character is placed alone in a "*span*" tag. For this reason the Arabic text loses its cursive nature.



```
<math
xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
   <mi>جتا</mi>
   <mo>&ApplyFunction;</mo>
   <mi>س</mi>
  </mrow>
 </math>
```

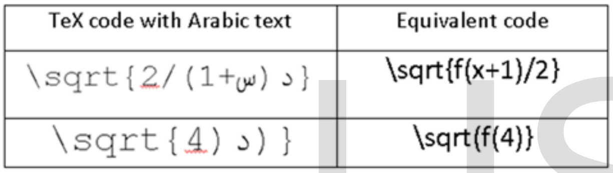
Fig.6.: Cursivity problem for the cosine Arabic function

Adding Arabic characters codepoints and indicating corresponding fonts and metrics, has overcome the cursivity problem.

6 Math-on-the-web and TeX

The composition of mathematical expressions in Arabic notation is possible through the RyDarab extension. Rydarab is a system that allows typesetting mathematical documents in a fully Arabic presentation. This means that the text is not only in natural language but equations are also composed with special symbols spreading out from right to left according to the natural direction of Arabic writing. Rydarab uses the same commands as those used for typesetting Latin based expressions (*sqrt* for square root, *frac* for fractions, etc.). For alphabetical symbols and Arabic text, Ryadarb is based on a latin transliteration. The use of transliterations is due to the fact that TeX does not support natively the use of Arabic text directly in source documents.

Contrarily to the TeX system, the use of the Arabic alphabet characters in TeX code displayed by MathJax should not raise any problem. In fact, the majority of browsers support Arabic natively. However, there is a problem owing to the cohabitation of the Arabic characters and other characters that compose TeX commands. We must therefore resolve bidirectional problems not only in output but also in the input code of mathematical expressions in this case.



TeX code with Arabic text	Equivalent code
\sqrt{2/(1+س) د}	\sqrt{f(x+1)/2}
\sqrt{4) د)}	\sqrt(f(4)}

Fig.6.: Bidirectionality problem source code

In this case, the TeX code is difficult to read and understand because the text isn't displayed correctly. One needs to use special text editor to write or visualize this code. But even if the text editor deals with such a problem, it will not be sufficient if the user wants to visualize the document source using a browser. Therefore, it will be interesting to give a transliteration based solution in addition to special text editor. Indeed, the Arabic text and the symbols will be typed using the Latin based transliteration as it is done in RyDArab. A third solution is the use of Arabic text and Arabic commands in the same way it is done in DadTeX [12].

7 Math-on-the-web and AsciiMath

A new experience is to use asciimath for typesetting mathematical expressions in Arabic notation. In this case the problems are the same as those for TeX encoding. The main problem is linked to the cohabitation of texts with different directionality. The solutions are then the same as those described in the preceding subsection.

7 Projects' state of advancement

The math-on-the-web platform needs continuous work to deal with all issues related to the display of mathematical expressions in Arabic notation. In fact, there are many details that should be taken into account to satisfy both Arabic Calligraphic and mathematical typesetting rules. Currently, the plateform provide right to left rendering, cursive text and reflected symbols. A great part of the future work will be dedicated to the creation of special fonts. Another part is related to tests to discover other possible problems for more browsers support. The first version of the platform provides the essential for typesetting mathematical expressions in both Arabic pure and mixed notations. The following figures show two samples of expressions obtained by the use of the platform.

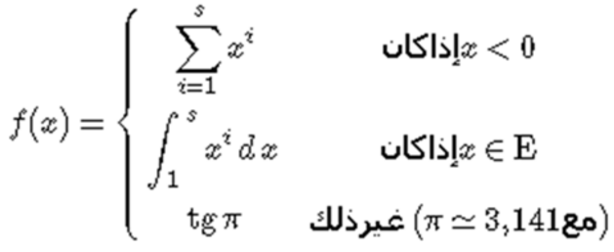

$f(x) =$
$\sum_{i=1}^{s} x^i$ إذاكان$x < 0$
$\int_1^s x^i\,dx$ إذاكان$x \in \mathrm{E}$
$\mathrm{tg}\,\pi$ غيرذلك $(\pi \simeq 3{,}141$مع$)$

Fig.8.: Sample of mixed expression composed by the Platform

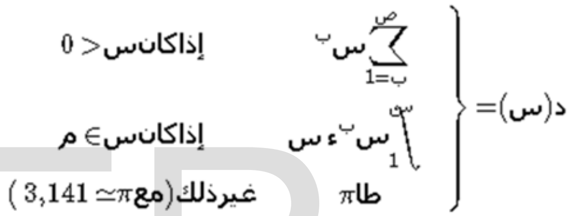


Fig.9.: Sample of pure Arabic expression composed by the Platform

8 Conclusion

This Work describes the issues encountered in presenting mathematical expressions in Arabic notation on the web using HTML and CSS output. The objective is to propose solutions for these issues and provide users with a cross-browser JavaScript platform that displays Arabic mathematical notation using MathML, LaTeX and ASCIIMath. The problems addressed in this paper are not limited to MathJax, but to any similar IT solution [13][14]. The aim of chosing an open source platform is to engage more people about this issue. The proposed platform enables users to view mathematical expressions without any significant technical knowledge or effort. It provides high display quality in compliance with Arabic calligraphic and mathematical typographic rules. Despite the efforts and the progress made in this project, it is at an early stage and needs more efforts.

References

[1] R. MINER, The importance of MathML to mathematics communication. Notices of the AMS, vol. 52, no 5, p. 532-538, 2005.

[2] A. LAZREK, M. EDDAHIBI, K. SAMI, B. MILLER, Arabic mathematical notation. W3C Interest Group Note, 2006.

[3] D. E. KNUTH, The texbook. Reading, Massachusetts : Addison-

IJSER

Wesley, 1986.

[4] R. AUSBROOKS, S. BUSWELL, D. CARLISLE, et al. Mathematical Markup Language (MathML) Version 3.0. W3C Candidate Recommendation of 15 December 2009. World Wide Web Consortium, vol. 13, 2009.

[5] M. EDDAHIBI, A. LAZREK, and K. SAMI,. Arabic mathematical e-documents. In : TeX, XML, and Digital Typography. Springer Berlin Heidelberg. p. 158-168, 2004.

[6] LAZREK, Azzeddine. RyDArab—Typesetting Arabic mathematical expressions. TUGboat, vol. 25, no 2, p. 141-149, 2004.

[7] CERVONE, Davide. MathJax: a platform for mathematics on the Web. Notices of the AMS, vol. 59, no 2, p. 312-316, 2012.

[8] F. KURTCU, Use of fonts on websites. Global Journal on Humanities and Social Sciences, vol. 1, no 1, 2015.

[9] M. DAVIS, et al. The bidirectional algorithm. Unicode Standard Annex, vol. 9, p. 95170-0519, 2004.

[10] R. Ishida, A. Lanin, Proposals to add isolation for bidi content in HTML5, W3 Internationalization Group, 2014.

[11] A. LAZREK, CurExt, typesetting variable-sized curved symbols. TUGboat, vol. 24, no 3, p. 323-327, 2003.

[12] M. EDDAHIBI, A. LAZREK, and K. SAMI, DadTEX—A full Arabic interface. The Communications of the TEX Users Group, p. 154, 2006.

[13] M. Bourne , KaTeX - a new way to display math on the Web, the IntMath blog, 2014.

IJSER