MathML for mathematics research articles

Arun Ram
Department of Mathematics
University of Wisconsin, Madison
Madison, WI 53706 USA
ram@math.wisc.edu

and

Department of Mathematics and Statistics
University of Melbourne
Parkville VIC 3010 Australia
A.Ram@ms.unimelb.edu.au

1. Introduction

Given the ubiquity with which the internet has taken over our lives and the extent to which society is using web pages written in HTML it is, perhaps, a bit surprising that research mathematicians still use TeX to distribute their results. Current web browsers and authoring tools are much more friendly towards formatting, graphics, animation, colors and cross referencing. The possibilities are endless, and it is a pity that we don't currently have them available in our standard mathematics presentations. For example,

- color,
- animated graphics,
- proofs that appear only when you want (click on) them,
- point and click jumping to references and previous equations,

are only a few of the possible tools that mathematicians might like. Perhaps it is time that we, as a community, make the switch and stop using TeX in favor of web markup.

In fact, some TeX addons (HyperTeX [HT], Beamer [Bm]), attempt to incorporate some weblike features. In 1998, MathML version 2 was released by the World Wide Web Consortium, the standards organisation for Web markup. So what is holding up our conversion? On one hand, research mathematicians are right to be hesitant to try to convert to MathML because the existing software is sometimes user unfriendly and this can be very frustrating. On the other hand, it is a vicious circle, because mathematicians have not demanded better software, software for MathML has developed extremely slowly. As soon as we, as a community, start trying to use the existing tools they will develop very quickly.

Thanks to the efforts of Mozilla (particularly Roger Sidje) there is a robust browser that processes MathML with good looking results.

- **Mozilla Firefox**   http://www.mozilla.org

What about authoring tools? A few hours of playing with Dreamweaver shows how easy typing into web pages can be. Unfortunately, Dreamweaver doesn't handle MathML. But

- **Amaya**   http://www.w3.org/Amaya

does and it is because of Mozilla and Amaya that I think that use of MathML is a realistic possibility for research mathematicians. Though Amaya crashes way too often to give it any more than beta-beta-beta status it is on the right track and a good programmer might take this template and make it into a wonderful MathML authoring tool in a month or two. I think that the mathematical community will jump to web presentation as soon as the rendering and authoring tools become polished. Greater usage of these tools by mathematicians would quickly push the currently available browsers to render mathematical markup even better.

This article has two goals:

- To illustrate that MathML is capable of satisfying the research mathematician's typsetting and presentation needs, and
- To provide a "getting started" guide for using MathML for writing mathematics papers.

Production of this article was partially supported by research and teaching intiatives supported by the National Science Foundation under Grant No. DMS-0353038. I thank National Science Foundation for continuing support of my research and teaching. This article was completed during the special semester on Combinatorial Representation Theory at Mathematical Sciences Research Institute (MSRI) in Berkeley. I thank MSRI for support and a wonderful and stimulating working environment.

2. Background information

The first and **most important** step is to get a browser that can display MathML. Mozilla Firefox [Fx], (or Camino [Cm], optimised for the Mac) seems to be the best option. Download the necessary fonts from STIX fonts [SX], MathML at MIT [MT], or the appropriate font installer in the right hand column of the fonts for MathML enabled Mozilla [ft]. Test your browser with the MathML torture test [Tt]. Other browser options are:

- Amaya [Am]
- Internet Explorer with the Design Science MathPlayer plugin [DS]

Opera [Op] does not advertise any support of MathML.

MathML is an add-on to HTML, the base language for writing webpages. All the basic formatting is done with HTML (titles, paragraphs, links,...) and so only the formatting of mathematical formulas needs to be done with MathML commands. The result is that MathML has fewer commands than TeX. It is not necessary to learn HTML and MathML separately. After reading this article, start by modifying the template for an article at http://www.math.wisc.edu/~ram/MathGlossary/template.xml [tmp]. If there is something you want to typeset which is not found in the template then look at the following useful pages, full of examples:

- Notes by Arun Ram in MathML   http://www.math.wisc.edu/~ram/MathGlossary
- Mozilla MathML Demos page   http://www.mozilla.org/projects/mathml/demo/
- The W3C MathML test suite   http://www.w3.org/Math/testsuite/

MathML comes in two flavours: *presentation markup* and *content markup*. At the moment, most browsers do not render content markup (see MathML for Mozilla [Mmz] and Amaya overview [Ao]). It seems that presentation markup is sufficient for mathematical typesetting needs and this article treats only presentation markup.

Style sheets for formatting are standard in TeX (e.g. amssym). The analogue for HTML is CSS (Cascading Style Sheets). There is no need to learn CSS separately, the necessary commands to get started are in the header of the template for a MathML typeset research article at http://www.math.wisc.edu/~ram/MathGlossary/template.xml [tmp]. Appendix 1 contains a list of the most useful HTML commands and (all) MathML commands. The authoritative sources for commands and syntax are the W3C (World Wide Web Consortium) recommendations:

- HTML (Hypertext Markup Language) http://www.w3.org/TR/html401/
- CSS (Cascading Style Sheets) http://www.w3.org/TR/CSS21/
- XHTML (Extensible hypertext markup language) http://www.w3.org/TR/xhtml1/
- XML (Extensible markup language) http://www.w3.org/TR/xml/
- XSL (Extensible stylesheet language) http://www.w3.org/TR/xsl/
- MathML (Mathematics markup language) http://www.w3.org/TR/MathML2/
- SVG (Scalable vector graphics) http://www.w3.org/TR/SVG/

The W3C specifications are often more useful than books from the library or the bookstore. If you do go read the MathML specification, read Chapter 3 first.

3. Formatting an article with HTML and MathML

The following gives some basics for creating pages with HTML and MathML. The code for most of the are often examples in this article is included in the template article at http://www.math.wisc.edu/~ram/MathGlossary/template.xml [tmp]. If you are viewing this article on the web use **View Source** in the **View** menu of your browser to see the code or download this page and edit it in your favourite editor.

Though MathML is simpler to learn than TeX, it has one great annoyance (which is, very likely, the reason it hasn't caught on more quickly): *nearly every* printing character is supposed to put inside `<mi> </mi>` or `<mo> </mo>` or `<mn> </mn>` (for "math identifier", "math operator" and "math number"). This means that the equation $x^2 + y^2 = 1$ is typeset with the code

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<msup><mi>x</mi><mn>2</mn></msup>
<mo>+</mo>
<msup><mi>y</mi><mn>2</mn></msup>
<mo>=</mo><mn>1</mn>
</math>
```

The consequence is that it is tedious to type in MathML by hand and some software that helps is desirable. Amaya is still in beta-version but it can be useful. Hopefully, Amaya will be improved quickly. Javascript can do amazing things: there is a javascript ASCII to MathML translator [AT] written by Peter Jipsen [Jp] and a javascript LateX to MathML translator [TT] written by Douglas Woodall and Jeff Knisley [Kn]. Perhaps translation is too extreme, a MathML dialect that retains the advantages of web markup and is not tedious to enter might be the ideal. Although I have not yet implemented this myself, a bit of javascript at the beginning of your web page that automatically puts in the tedious `mi`, `mo`, and `mn` tags would enable the same equation, $x^2 + y^2 = 1$, to be typeset with the code

```
<mymath> <msup>x2</msup> + <msup>y2</msup> =1 </mymath>
```

3.1 Features from HTML and CSS

- For *displayed equations* the simplest option is to use `<blockquote> put equation here </blockquote>`. Alternatively, the header in the template article at http://www.math.wisc.edu/~ram/MathGlossary/template.xml [tmp] includes the CSS lines

  ```
  table.dispeq{ width:100%; }
  td.eq{ text-align:center }
  td.eqno.after { content: "("counter(section)"."counter(eqn)")";
       counter-increment: eqn;
       font-style: normal; }
  ```

With these header lines displayed equations are inserted with

```
<table class="dispeq"> <a name="eqname"></a>
<tr>
<td class="eq">  Insert equation here  </td>
<td class="eqno"></td>
</tr>
</table>
```

where the line `<td class="eqno"></td>` is needed only for a numbered equation and the `<a name="eqname"></a>` is only needed to refer back to this equation later in the text.

- The header material in the template article at http://www.math.wisc.edu/~ram/MathGlossary/template.xml [tmp] includes CSS commands (inside the `<style> ... </style>` tags) which provide *formatting and automatic numbering for title, author, sections, subsections, equation numbers, theorems, propositions and lemmas*. With these lines in the header, the title and author are put in with

  ```
  <h2 class="title">My title</h2>

  <p class="author">
  My Name <br />
  Department of My Stuff <br />
  University of My Place <br />
  City, ST 99999 SUE
  </p>
  ```

  the section and subsection headers are put in with

  `<h2 class="section">A section</h2>` and `<h3 class="subsection">A subsection</h3>`

  respectively, and a theorem with a link back to equation (2.1) is coded with

  ```
  <div class="theorem">
  My theorem with a link back to equation (<a href="#firsteqn">2.1</a>).
  </div>
  ```

- A standard *bibliographic entry* might look something like

  [BG] A. Braverman and D. Gaitsgory, *Crystals via the affine Grassmanian*, Duke Math. J. **107** no. 3, (2001), 561-575.

  Dressed up a bit, it has links to the authors' web pages, a link to the Math Review, a link to the journal, link to the arXiv version, and an anchor so that the reader can click on [BG] wherever it happens to be in the text and immediately jump to this reference entry.

  [BG] A. Braverman and D. Gaitsgory, *Crystals via the affine Grassmanian*, Duke Math. J. **107** no. 3, (2001), 561-575; arXiv:math/9909077v2, MR1828302 (2002e:20083)

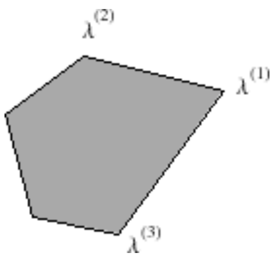
- Because most web pages are continuous media, not paged media, the conventions for *footnotes* and *acknowledgments* may differ. A reasonable convention would be to place acknowledgments at the end of the introduction. Footnotes at the end of the document with anchors will have easy point and click link access. For information on including specific formatting rules for the printed version see Section 7 of the CSS Specification. Page breaks are discussed in Section 13.

- A *link* to the Mozilla MathML torture test [Tt] is coded with

  ```
   <a href="http://www.mozilla.org/projects/mathml/demo/texvsmml.xhtml">
  Mozilla MathML torture test</a>.
  ```

- The *graphic* in the line

$$\mathrm{span}\{v_1, \ldots, v_k\} \overset{\mu}{\mapsto} \mathrm{Conv}(\lambda^{(1)}, \ldots, \lambda^{(k)}) = \qquad (3.1)$$



is coded with

```
<img xmlns="http://www.w3.org/1999/xhtml"
align="middle" alt="convex hull"
src="http://www.math.wisc.edu/~ram/MathGlossary/convexhull.png" />
```

Nearly any graphic you make with your favourite picture drawing program can be included in this fashion (in this example, the `.png` file containing the graphic was made with the picture drawing program Fireworks [Fw]).

- In the example in (3.1), the 'Conv' is coded with `<mi title="convex hull">Conv</mi>` which has the effect that a tootip containing the words "convex hull" appears on mouse over.

3.2 MathML Features

There are plenty of math symbols available in MathML.

| ‖ †‡ℏ℧ℵ ← ↑ ↓ → ↔ ∤ ⇐ ⇑ ⇓ ⇒ ⇔ ⇕ ↖ ↗ ↙ ↘ ↦ ↤ ↥↧ ↩ ↪ ⇄ ⇆ ⇅ ⇵ ⇤ ⇥ ∀ ∂∃
∄ ∈ ∉ ∍ ∌ ∏ ∐ ∑ ± ∓ ∝ ∞∧∨∪∩∫∬∮ ∴ ∼ ≀ ≃ ≄ ≅ ≇ ≈ ≉ ≈ ≉ ≠ ≡ < > ≤ ≥ ≪ ≫
≮ ≯ ≰ ≱ ≺ ≻ ≼ ≽ ⊂ ⊃ ⊄ ⊅ ⊆ ⊇ ⊈ ⊉ ⊊⊋ ⊎ ⊏ ⊐ ⊑ ⊒ ⊔ ⊓ ⊕ ⊖ × ⊗ ⊢ ⊣ ⊤ ⊥ ⊧
⊲ ⊳ ⊴ ⊵ ⋄ ⋅ ⋆ ⋈⋉⋊…⋮⋯⋰⋱⌈C′⌉⊆⌊C″⌋⟨⟩♭♮♯♠♣

The Mozilla MathML character table [CT] is extremely helpful, both for finding available characters and for determining character names. The following links will help for navigating these tables:

1. Logic table http://www.mozilla.org/projects/mathml/fonts/chars/table00xx.html
2. Arrows table http://www.mozilla.org/projects/mathml/fonts/chars/table21xx.html
3. Operators and Relations Table http://www.mozilla.org/projects/mathml/fonts/chars/table22xx.html

- *Greek* letters are coded with `&alpha;   &beta;   ...   &Phi;   &nabla;`,

  ΓΛΣΨΔΞΥΩΘΠΦ∇ αβγδεϵ∈ζηθϑικλμνξοπϖρϱσςτυϕφχψω

- *blackboard bold* characters with `&Aopf;   &Bopf;   ...   &yopf;   &zopf;`,

  𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ 𝕒𝕓𝕔𝕕𝕖𝕗𝕘𝕙𝕚𝕛𝕜𝕝𝕞𝕟𝕠𝕡𝕢𝕣𝕤𝕥𝕦𝕧𝕨𝕩𝕪𝕫

- *calligraphic* characters with `&Ascr;   &Bscr;   ...   &yscr;   &zscr;`,

  𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵
  𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀ℓ𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏

- *fraktur* characters with `&Afr;   &Bfr;   ...   &yfr;   &zfr;`, and

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ $\mathfrak{abcdefghijklmnopqrstuvwxyz}$

- *cyrllic* characters with `&Acy;  &Bcy;  ...   &ycy;   &dzcy;`.

  ЁЂЃЄЅІЇЈЉЊЋЌЎЏАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ
  абвгдежзийклмнопрстуфхцчшщъыьэюяёђѓєѕіїјљњћќўџ

Math formulas and notation are typeset inside
`<math xmlns="http://www.w3.org/1998/Math/MathML">` math stuff here `</math>`.
This is analogous to $ in TeX.

- For *superscripts* use `<msup>` and for *subscripts* use `<msub>`. For *multisubscripts and superscripts and prescripts* use `<mmultiscripts>`.

$$b_n, \qquad x^2, \qquad R_{i\ kl}^{\ j}, \qquad \text{and} \qquad {}_{\ell\ np}^{m}R_{i\ kl}^{\ j}. \tag{3.2}$$

- *Sums (`&sum;`), products (`&prod;`), integrals (`&int;`) and other similar "large" operators* should resize appropriately if they are entered in the format `<mo>&sum;</mo>`. Similarly, *parentheses, braces and brackets* should resize appropriately if they are entered as `<mo>(</mo> ... <mo>)</mo>`.

$$G = \bigsqcup_{w\in\tilde{W}} IwI, \qquad \left(\sum_{i=1}^{\infty}\left\langle\frac{m}{n}\right\rangle\right) \qquad \left\langle\prod_{i=1}^{\infty}\left\{\frac{m}{n}\right\}\right\rangle \qquad \left\{\int_{-\infty}^{G}\left(\frac{m}{n}\right)\right\} \tag{3.3}$$

  If parentheses don't automatically size correctly enclose the content inside `<mrow> ... </mrow>`. Parentheses are not required to be paired so a "cases" is easily accomplished with no extra effort.

$$T_w T_{s_i} = \left\{\begin{array}{ll} T_{ws_i}, & \text{if } ws_i > w, \\ 0, & \text{otherwise.} \end{array}\right. \tag{3.4}$$

- *Overbars, wide tildes, wide hats, vector symbols, overbraces, and underbraces* should be done with `<mover>` and `<munder>`.

$$\overbrace{x+y+z} \div \tilde{x} \otimes \widehat{x} \ \ltimes x \pm \dot{x} \circ \ddot{x} < \mathring{x} \in \overleftarrow{x} \neq \overrightarrow{x} \wedge \overleftrightarrow{x} \tag{3.5}$$

  The arrows stretch well in Mozilla and the `&Tilde;`, `&OverBar;`, `&Hat;` are also supposed to stretch, but the only one that does is `&Hat;`.

$$\tilde{xxx} \xrightarrow{\text{goes to}} \overrightarrow{\text{vector}} \quad \text{and} \quad \bar{xxx} \xrightarrow{\qquad\sim\qquad} \widehat{xxxx} \xleftarrow{\ \varphi\ } \underbrace{\{\overbrace{a, \ldots, a}^{k\ a\text{'s}}, \overbrace{b, \ldots, b}^{\ell\ b\text{'s}}\}}_{k+\ell\ \text{elements}} \tag{3.6}$$

  where the last example is stolen from the Mozilla MathML torture test [Tt].

- For *fractions* use

  `<mfrac linethickness="1">` `<mrow>` numerator `</mrow>` `<mrow>` denominator `</mrow></mfrac>`,

  where the `linethickness="1"` should be left out for ordinary fractions and set to 0, instead of 1, for binomial coefficients.

$$\sum_{\substack{0\le i\le m\\ 0<j<n}} \binom{n}{k/2} \qquad \left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) \qquad \frac{x+\frac{y}{2}}{k+\frac{1}{x}} \qquad \dfrac{x+\frac{y}{2}}{k+\dfrac{1}{x}} \tag{3.7}$$

are examples stolen from the Mozilla MathML torture test [Tt]. Resizing in fractions done with the `displaystyle` and `scriptlevel` attributes of the `<mstyle>` tag.

- *Matrices, math tables and aligned equations* are done with `<mtable>`. The equation alignment in the second example below is done with `<mtable columnalign="right center left">`.

$$\begin{pmatrix} \begin{pmatrix} \ddots & b \\ \iddots & d \end{pmatrix} & \begin{pmatrix} \cdots & \vdots \\ g & h \end{pmatrix} \\ 0 & \begin{pmatrix} i & j \\ k & l \end{pmatrix} \end{pmatrix} \qquad \begin{array}{rcl} h_{\lambda^\vee}(f)h_{\mu^\vee}(g) & = & h_{\mu^\vee}(f^{-1})h_{\lambda^\vee+\mu^\vee}(f)h_{\lambda^\vee+\mu^\vee}(g)h_{\lambda^\vee}(g^{-1}) \\ & = & h_{\mu^\vee}(f^{-1})h_{\lambda^\vee+\mu^\vee}(fg)h_{\lambda^\vee}g^{-1}) \\ & = & h_{\mu^\vee}(f^{-1})h_{\mu^\vee}(fg)h_{\lambda^\vee}(fg)h_{\lambda^\vee}(g^{-1}) \\ & = & h_{\mu^\vee}(g)h_{\lambda^\vee}(f). \end{array} \tag{3.8}$$

The `malignmark`, `maligngroup` and `groupalign` are not yet implemented in Mozilla. Check the page status report of MathML tags in Mozilla [st] for the latest updates.

- *Resizing* is best done by enclosing the content in an `<mstyle scriptlevel="-1"> ... </mstyle>` where the -1 is changed to -2 or +1 or +2, as appropriate.

- *Positioning* (up/down/left/right) of an object is controlled with `<mpadded>`. This doesn't render exactly right in Mozilla as

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mpadded width="+1em"><mi>&alpha;</mi></mpadded>
<mpadded height="-2em"><mi>&beta;</mi></mpadded>
<mpadded depth="+4em"><mi>&gamma;</mi></mpadded>
<mpadded lspace="+1em"><mi>&delta;</mi></mpadded>
</math>
```

renders as $\alpha \quad \beta\gamma \quad \delta$

and the example, from 3.3.6.5 of the MathML specification, of a (strongly discouraged) overstruck C to try to make a sign for the complex numbers renders as $C|$

- For *text inside math* use `<mtext> &nbsp;` text here `&nbsp; </mtext>`. The `&nbsp;` puts a space.

- For inserting *space inside math mode* use `<mspace width="2em"/>`, where the 2 can be adjusted as desired. Other options are the space characters `&nbsp;`, `&ensp;`, `&emsp;`, `&emsp13;`, `&emsp14;`, `&numsp;`, `&puncsp;`, `&thinsp;`, `&hairsp;`, illustrated by
nbsp ensp emsp   emsp13 emsp14 numsp puncsp thinsp hairsp ,respectively.

- *Phantom content* is inserted with `<mphantom> ... </mphantom>` (analogous to the command \phantom in TeX).

- *Comments* are inserted as in HTML, with `<!--` comment here `-->`.

- *Boxes* can be placed around content using `<mtable frame="solid">`, *color* with `<mstyle color="your color"> your colored content </mstyle>`, and a *colored background* is accomplished with `<mstyle background="yellow"> stuff </mstyle>`.

$$\boxed{x^2 + y^2 = 1} \qquad x^3 + y^3 = 2 \qquad x + 7 + 3 + z + q + q \tag{3.9}$$

where the last example is Example miScolorname15 from the MathML testsuite [TS]. The MathML tag `<menclose>` is not yet implemented in Mozilla. Check the page status report of MathML tags in Mozilla [st] for the latest updates.

- The `<maction>` tag provides a nice option of "toggling" content, by clicking on it 3 + 2 = 32 (try it, click on it).

4. Using Amaya for MathML authoring

Amaya is very useful for putting in all those `<mi>`, `<mo>`, setting up subscripts, superscripts, matrices and fractions. Amaya, at the moment, is painful for entering math symbols and it has the annoying problem that, even if you get it entered right, when you save your document, Amaya will automatically change character entities that have been entered manually. For example, `&langle;` will change into some Chinese character.

Amaya crashes often. Until Amaya gets more robust the best option is to use Amaya in tandem with your favourite editor. Do the layout of the text and the equations with Amaya. Type a few paragraphs into Amaya then copy the code into your favourite editor and fix up the character names. Use a slightly different name for Amaya file and your document file. When using Amaya

- Always start a new document by opening an existing template [tmp].
- Double click the lower edge of the display window (on mouse over it turns pink) to display in split mode and see the code that Amaya produces.
- Type into the display window to create text. I have found Amaya most useful when I *don't* use the math tool palette.
- Use the keyboard shortcuts to start math tags and use the arrow keys to navigate the boxes Amaya produces for matrices, superscripts and subscripts.
- Use the right arrow key to move the cursor out of math mode.
- Save often.
- Select **Display all shortcuts** in the **Preferences > General** dialog and restart Amaya.
- If you really want to start a blank document with Amaya do **New XHTMLdocument** from the **File Menu**, make sure that the **Documentprofile** is **XHTML1.1** and that the filename ends in `.xml` and type math into it. Amaya will automatically put in the correct DTD (Document Type Definition).

4.1 Amaya MathML keyboard shortcuts for Mac OS X

Command	Tag	Shortcut
New Formula	<math>	Ctrl-m Ctrl-m
Subscript	<msub>	Ctrl-m Ctrl-v
Superscript	<msup>	Ctrl-m Ctrl-6
Subscript and Superscript	<msubsup>	Ctrl-m Ctrl-b
Multiscripts	<mmultiscripts>	Ctrl-m Ctrl-s
Under	<munder>	Ctrl-m Ctrl-u
Over	<mover>	Ctrl-m Ctrl-o
Under and over	<munderover>	Ctrl-m Ctrl-k

Fraction	<mfrac>	Ctrl-m Ctrl-f
New Matrix	<mtable>	Ctrl-m Ctrl-h
Root	<mroot>	Ctrl-m Ctrl-r
Square root	<msqrt>	Ctrl-m Ctrl-q

4.2 Problems with Amaya

- The most annoying problem is that Amaya automatically changes character entities that have been entered manually. For example `&langle;` will change into some Chinese character.
- Amaya needs an efficient way to enter characters.
- Amaya tends to crash when it gets confused.
- The rows and columns are reversed in the matrix dialog box.
- Looking at example pages shows that Mozilla is superior for MathML rendering.

5. Tips

- The following useful information and example pages are provided by Mozilla and W3C:
    - W3C MathML test suite 2.0 http://www.w3.org/Math/testsuite/
    - Main Mozilla MathML page http://www.mozilla.org/projects/mathml/
    - Mozilla MathML Demos page http://www.mozilla.org/projects/mathml/demo/
    - Mozilla MathML Character tables http://www.mozilla.org/projects/mathml/fonts/chars/table.html
    - Mozilla MathML Fonts page http://www.mozilla.org/projects/mathml/fonts/
    - Mozilla MathML Font encodings http://www.mozilla.org/projects/mathml/fonts/encoding/
    - Mozilla Symbol Font encoding http://www.mozilla.org/projects/mathml/fonts/encoding/symbol.html

- If the usual minus sign doesn't show up properly hold down the Option key and type the usual minus sign to get a minus sign that displays properly (this may be Mac specific).

- A math character will slant less if put in an `<mo>` instead of an `<mi>`.

- Since HTML prefers that the user control spacing from style files the best hack for putting spaces in seems to be to go into math mode and put in mathspaces. Using spacing to tweak the look of the output is strongly discouraged as the results are usually renderer dependent. The ultimate tweaking is shown at the MathML test suite torture test complex 3 [TS].

- CSS rules in a document are included inside `<style type="text/css"> ... </style>` inside the `<head> ... </head>`. Alternatively, they can all be put into a style sheet, and then loaded with `@import "mathsty.css";`

- Inserting media specific (for example, for printers) CSS formatting rules is explained in Section 7 of the CSS Specification [CSS]. Page breaks are discussed in Section 13 of the CSS specification [CSS].

- When printing from Mozilla be sure that "Shrink to fit page width" is *not* selected. Sometimes the shrunk MathML does not print correctly.

- The details for automatic numbering are explained in Section 12.4 and 12.2 of the CSS specification [CSS]. Alternatively, automatic numbering and referencing ought to be doable with fairly simple javascript.

- The use of namespaces makes it possible to put XHTML commands (for example, links and graphics) inside MathML.

- Using UTF-8 character encoding is always the best. For ordinary html pages use the XHTML Transitional DTD.

- The page Authoring MathML for Mozilla [Au] suggests replacing the usual DTD with

  ```
  <?xml version="1.0"?>
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
                 "http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd" [
    <!ENTITY mathml "http://www.w3.org/1998/Math/MathML">
  ]>
  <html xmlns="http://www.w3.org/1999/xhtml">
  ```

  This has the effect that the usual entrance into math mode

  ```
  <math xlmns="http://www.w3.org/1998/Math/MathML">
  ```

  only needs to be coded with `<math xlmns="&mathml;">`

- `<mi>ij</mi>` renders as ij, `<mi>i</mi><mi>j</mi>` renders as $i j$, and `<mi> <mi>i</mi><mi>j</mi></mi>` renders as *ij*.

- Doing graphics with SVG (Scalable vector graphics) should obviate the need for external graphics files and all browsers that handle MathML also render SVG. Inkscape [Ik] is a drawing program that will output graphics in SVG (Scalable Vector Graphics). It should be possible to include formulas in MathML inside an SVG picture with the use of the foreignObject element of SVG. There is also a combined XHTML1.1+MathML2.0+SVG1.1 [mm] document type definition. However, I have been unable to determine whether the foreignObject tag is implemented in Inkscape, and I have not yet succeeded in getting mixed markup to render properly in Mozilla (it renders fine in Amaya).

- Other media types, such as animations, video and sound, can be included. For these, it is most efficient to produce the appropriate code using an application like Dreamweaver (or copy it from some other web page) and then copy it into your MathML document.

- Parentheses are supposed to resize appropriately if they are put in an `<mo>` tag. See 3.2.5.8 of the MathML specification.

6. References

[Am] Amaya, http://www.w3.org/Amaya/

[Ao] Amaya overview, http://www.w3.org/Amaya/Amaya.html

[AT] Peter Jipsen, *ASCII to MathML translator*, http://www1.chapman.edu/~jipsen/mathml/asciimath.html

[Au] Authoring MathML for Mozilla, http://www.mozilla.org/projects/mathml/authoring.html

[Bm] Beamer, http://en.wikipedia.org/wiki/Beamer_(LaTeX)

[Cm] Camino, http://www.caminobrowser.org/

[CSS] CSS (Cascading Style Sheets) Specification, http://www.w3.org/TR/CSS21/

[CT] MathML character table, http://www.mozilla.org/projects/mathml/fonts/chars/table.html

[DS] Design Science MathPlayer, http://www.dessci.com/en/products/mathplayer/

[ft] fonts for MathML enabled Mozilla, http://www.mozilla.org/projects/mathml/fonts/.

[Fw] Fireworks, http://www.adobe.com/products/fireworks/

[Fx] Mozilla Firefox, http://www.mozilla.org/products/

[HT] HyperTeX, http://arxiv.org/hypertex/

[Ik] Inkscape, http://www.inkscape.org/index.php?lang=en

[Jp] Peter Jipsen, http://www1.chapman.edu/~jipsen/

[Kn] Jeff Knisley, http://faculty.etsu.edu/knisleyj/.

[mm] XHTML1.1+MathML2.0+SVG1.1 Document Type Definition, http://www.w3.org/TR/XHTMLplusMathMLplusSVG/

[Mmz] MathML for Mozilla, http://www.mozilla.org/projects/mathml/authoring.html

[MT] MathML at MIT, http://web.mit.edu/is/products/mathml/

[Op] Opera, http://www.opera.com/

[st] Status report of MathML tags in Mozilla, http://www.mozilla.org/projects/mathml/update.html

[SX] STIX fonts, http://www.stixfonts.org/

[TS] The W3C MathML test suite, http://www.w3.org/Math/testsuite/

[tmp] Arun Ram, *Template for a research article in MathML*, http://www.math.wisc.edu/~ram/MathGlossary/template.xml

[TT] Douglas Woodall and Jeff Knisley, *LateX to MathML translator*, http://math.etsu.edu/LaTeXMathML/

[Tt] MathML torture test, http://www.mozilla.org/projects/mathml/demo/texvsmml.xhtml.

Appendix 1. HTML and MathML commands

A1.1 Basic HTML

- `<em>` *italics* `<em>`, `<strong>` **bold** `</strong>`, `<dfn>` *term* being defined `</dfn>`
- `<h2>` heading size 2 `</h2>`, `<h3>` heading size 3 `</h3>`, etc.
- `<p>` paragraph here `</p>`, `<br />` for a line break, `&nbsp;` for a space, `<pre>`

  ```
  preformatted
  ```

```
  text
```

`</pre>`, and `<hr />` for an hrule

- Make a bulleted list (unordered list) with

```
<ul>
  <li> list item </li>
  <li> list item </li>
</ul>
```

  - list item
  - list item

- Make a numbered list (ordered list) with

```
<ol>
  <li> list item one </li>
  <li> list item two </li>
</ol>
```

  1. list item
  2. list item

- Make a definition list with

```
<dl>
 <dt> definition term one</dt>
  <dd> definition description of term one</dd>
 <dt> definition term two </dt>
  <dd> definition description of term two</dd>
</dl>
```

definition term one
    definition description of term one
definition term two
    definition description of term two

- Make a table with:

```
<table>
<thead>
     <tr> header </tr>
</thead>
<tfoot>
     <tr> footer </tr>
</tfoot>
<tbody>
     <tr>
         <td>first</td>
         <td>second</td>
         <td>third</td>
     </tr>
     <tr>
         <td>fourth</td>
         <td>fifth</td>
         <td>sixth</td>
     </tr>
</tbody>
<tbody>
     <tr>
         <td>First</td>
         <td>Second</td>
         <td>Third</td>
     </tr>
     <tr>
         <td>Fourth</td>
         <td>Fifth</td>
         <td>Sixth</td>
     </tr>
</tbody>
</table>
```

header		
first	second	third
fourth	fifth	sixth
First	Second	Third
Fourth	Fifth	Sixth
footer		

- `<div class="mytype" name="myname"> ... </div>` is a way to group a set of lines and attach names, classes, special formats and other attributes to this collection of lines.

A1.2 MathML Presentation Markup tags

- element
  1. attribute = "value"
- the nine common attributes are
  1. `mathbackground="`*#rgb* | *#rrggbb* | *html-color name*`"`
  2. `mathcolor="`*#rgb* | *#rrggbb* | *html-color name*`"`
  3. `mathsize="small` | `normal` | `big` | *number* | *v-unit*`"`
  4. `mathvariant="number | bold | italic | script | bold-fraktur | double-struck | bold-script | fraktur | sans-serif | bold-sans-serif | sans-serif-italic | sans-serif-bold-italic | monospace"`
  5. `fontsize="`*number* `v-unit"`
  6. `fontweight=" normal` | `bold"`
  7. `fontstyle="normal` | `italic"`
  8. `fontfamily="`*string* | *css-fontfamily*`"`
  9. `color="`*#rgb* | *#rrggbb* | *html-color name*`"`
- all MathML presentation elements accept the following attributes
  1. `class`
  2. `style`
  3. `id`
  4. `xlink:href:`
  5. `other:`
- `<mn>` number
  1. the nine common attributes
- `<mo>` operator
  1. the nine common attributes
  2. `form="prefix` | `infix` | `postfix"`
  3. `lspace="`*number h-unit* | *namedspace*"
  4. `rspace=`"*number h-unit* | *namedspace*"
  5. `stretchy="true` | `false"`
  6. `symmetric="true` | `false"`
  7. `maxsize="`*number*[*v-unit* | *h-unit*] | *namedspace*`"`
  8. `minsize="`*number* | *namedspace*`"`
  9. `largeop="true` | `false"`
  10. `movablelimits="true` | `false"`
  11. `fence="true` | `false"`
  12. `separator="true` | `false"`
  13. `accent="true` | `false"`
- `<mi>` identifier
  1. the nine common attributes
- `<mtext>` text
  1. the nine common attributes
- `<mspace>` blank space
  1. the nine common attributes and
  2. `width="`*number h-unit* | *namedspace*`"`
  3. `height="`*number v-unit*`"`
  4. `depth="`*number v-unit*`"`
- `<ms>` string
  1. the nine common attributes
  2. `lquote="`*character*`"` default is `&quot`
  3. `rquote="`*character*`"` default is `&quot`
- `<mglyph>` character glyph

    1. the nine common attributes and
    2. `fontfamily="`*string* | *css-fontfamily "*
    3. `index="` *integer* `"`
    4. `alt="`*string*`"`
- `<mrow>` horizontal row
- `<mfrac>` fraction
    1. `linethickness="`*number* [*v-unit*] | `thin` | `medium` | `thick"`
    2. `numalign=" left` | `center` | `right"`
    3. `denomalign="(left` | `center` | `right`)+"
    4. `bevelled="true` | `false"`
- `<msqrt>` square root
- `<mroot>` root
- `<mstyle>` apply style
    1. all attributes of all presentation elements except for those that have a required value
    2. `scriptlevel="`[+|-] *unsigned integer*`"`
    3. `displaystyle="true` | `false"`
    4. `scriptsizemultiplier="`*number*`"` default is 0.71
    5. `scriptminsize="`*number v-unit*`"` default is 8pt
    6. `color="`*#rgb* | *#rrggbb* | `transparent` | *html-color-name*`"`
    7. `background="`*#rgb* | *#rrggbb* | `transparent` | *html-color-name*`"` default is `transparent`
    8. `veryverythinmathspace="`*number h-unit*`"` default is 1/18 em
    9. `verythinmathspace="`*number h-unit*`"`default is 1/9 em
    10. `thinmathspace="`*number h-unit*`"`default is 1/6 em
    11. `mediummathspace="`*number h-unit*`"` default is 2/9 em
    12. `thickmathspace="`*number h-unit*`"`default is 5/18 em
    13. `verythickmathspace="`*number h-unit*`"` default is 1/3 em
    14. `veryverythickmathspace="`*number h-unit*`"` default is 7/18 em
- `<mspace>` blank space
    1. the nine common attributes
    2. `width="`*number h-unit* | *namedspace*`"`
    3. `height="`*number v-unit*`"`
    4. `depth="`*number v-unit*`"`
- `<mphantom>` make content invisible
- `<mfenced>` add fences around content
    1. `open="`*string*`"` default is (
    2. `close="`*string*" default is)
    3. `separators="`*character**`"` default is ,
- `<menclose>` enclose content in a stretchy symbol
    1. `notation="longdiv` | `actuarial` | `radical"` default is `radical`
- `<merror>` error messages
- `<msub>` subscript
    1. `subscriptshift="`*number v-unit*`"`
- `<msup>` superscript
    1. `superscriptshift="`*number v-unit*`"`
- `<msubsup>` subscript and superscript
    1. `subscriptshift="`*number v-unit*`"`
    2. `superscriptshift="`*number v-unit*`"`
- `<munder>` underbrace
    1. `accentunder="true` | `false"`
- `<mover>` overbrace
    1. `accent="true` | `false"`
- `<munderover>` under and overbrace
    1. `accentunder="true` | `false"`
    2. `accent="true` | `false"`
- `<mmultiscripts>`

   1. subscriptshift="*number v-unit*"
   2. superscriptshift="*number v-unit*"
- <none/> is used only as a child element of the mmultiscripts element
- <mtable> table or matrix
   1. align="(top | bottom | center | baseline | axis)" default is axis
   2. rowalign="(top | bottom | center | baseline | axis)+" default is baseline
   3. columnalign="(left | center | right)+" default is center
   4. columnwidth="(auto | *number h-unit* | *namedspace* | fit)" default is auto
   5. width="auto | *number h-unit*" default is auto
   6. rowspacing="(*number v-unit*)+" default is 1.0 ex
   7. columnspacing="(*number h-unit* | *namedspace*)+" default is 0.8em
   8. rowlines="(none | solid | dashed)+" default is none
   9. columnlines="(none | solid | dashed)+" default is none
   10. equalrows="true | false" default is false
   11. equalcolumns="true | false" default is false
   12. frame="(none | solid | dashed)+" default is none
   13. framespacing="(*number h-unit* | *namedspace*) (*number v-unit* | *namedspace*)" default is 0.4 em 0.5 ex
   14. displaystyle="true | false" default is false
   15. side="left | right | leftoverlap | rightoverlap" default is overlap
   16. minlabelspacing="*number h-unit*"
   17. groupalign="*groupalignment-list-list*"
   18. alignmentscop="true | false"
- <mtr> row of a table
   1. rowalign="(top | bottom | center | baseline | axis)+" default is baseline
   2. columnalign="(left | center | right)+" default is center
   3. groupalign="*groupalignment-list-list*"default is {left}
- <mlabeledtr> labeled row of a table
   1. rowalign="(top | bottom | center | baseline | axis)+" default is baseline
   2. columnalign="(left | center | right)+" default is center
   3. groupalign="*groupalignment-list-list*"default is {left}
- <mtd> cell in a table
   1. rowalign="(top | bottom | center | baseline | axis)+" default is baseline
   2. columnalign="(left | center | right)+" default is center
   3. groupalign="*groupalignment-list-list*"default is {left}
   4. rowspan="*number*" default is 1
   5. columnspan="*number*" default is 1
- <maligngroup> alignment group
- <malignmark> alignment marker
   1. edge="left | right" default is {left}
- <maction> interactive actions
   1. selection="*number*" default is 1
   2. actiontype="depends on application"
- <mpadded> precise positioning
   1. width="[+ | -] *number* | (%[*pseudo-unit*] | *pseudounit* | *h-unit* | *namedspace*)"
   2. lspace="[+ | -] *number* | (%[*pseudo-unit*] | *pseudounit* | *h-unit*)"
   3. height="[+ | -] *number* | (%[*pseudo-unit*] | *pseudounit* | *h-unit*)"
   4. depth="[+ | -] *number* | (%[*pseudo-unit*] | *pseudounit* | *h-unit*)"
- <mprescripts> used in mmultiscripts to indicate folowing elements in prescript position