Unicode mathematics in LaTeX: advantages and challenges

Will Robertson
School of Mechanical Engineering
University of Adelaide, SA, Australia
will.robertson@latex-project.org

Abstract

Over the last few years I've been tinkering with unicode mathematics in XeTeX. Late-2009 I spent a few weeks ironing out the significant bugs and think I've got a pretty good handle on the whole system now. In this presentation, I'd like to discuss the advantages unicode maths brings to LaTeX, challenges faced dealing with unicode, challenges with maths fonts (including the STIX fonts), challenges with compatibility with amsmath and/or MathML, and assorted related remarks. In future plans, I hope to use this system as the basis for equivalent development in LuaTeX as well.

1 Introduction

XeTeX was the first widely-used unicode extension to TeX. Several years ago Jonathan Kew [4] added OpenType maths support to XeTeX following from Microsoft's addition to the OpenType spec for Microsoft Word 2007. Around that time I built a prototype implementation of a unicode maths engine for LaTeX, but with very few OpenType maths fonts available, and other projects consuming my time, the project lost momentum and I never managed to get the package on CTAN.

Although the font situation has improved little since then, the STIX fonts have at least been shown in beta form and can be used to some degree. By the time you are reading this, an experimental version of the unicode-math package will hopefully have been released on CTAN. This extended abstract is a brief explanation of what it does and what it's good for.

2 What is unicode maths?

Barbara Beeton [1] has said it all much better than I could. In short: every known symbol to be used in technical writing has been included in the unicode specification. We now have a formal description of (some many thousand) exactly what glyphs a maths font should contain and the particulars of how they should behave [2]. (Contrast with maths fonts used with TeX documents that all contain a different set of glyphs.)

Since then, Microsoft has extended the OpenType specification to include tables of structured information for mathematics typesetting, essentially parameterising Knuth's original algorithms in TeX. For more context in this area, the historical development of unicode maths was summarised well by Ulrik Vieth [6].

Any unicode variant of TeX could always access the unicode maths glyphs. But typesetting them well is only possible with explicit layout intructions based on the Math OpenType specification.

2.1 So what does the package do?

After loading the package, users can write

```
\setmathfont{Cambria Math}
```

to select Cambria Math or any other OpenType font that has mathematics support. This can happen at any time during the document, not just in the preamble as with current LaTeX math font selection.

Control sequences are provided to access every defined unicode maths glyph, and literal input of all such characters in the source is also supported. Maths can be copied from another source (such as a web page or PDF document) and pasted directly into the LaTeX document and the content will be retained, albeit with some loss of its presentational aspects (most notably subscripts and superscripts).

With some minor exceptions, no changes to the mathematical document source should be necessary to be able to switch fonts using unicode maths.

3 Advantages

The main advantage to having a unicode maths engine is that it becomes as easy to change maths fonts as it is to change text fonts. Contrast this with what packages such as euler must do to switch from the Computer Modern Maths font. Standardising around a common math font encoding was the original goal of the Math Font Group when they invented an 8-bit math font encoding but then realised that

unicode was the future. Unicode maths brings more benefits than simply standardising the way maths fonts are loaded, however.

3.1 Pragmatic

It's never fun when typesetting mathematics to have to go hunting for a specific symbol. (Scott Pakin's 'Symbols' guide makes the task far easier, however.) Which maths font glyph package to load? Which control sequence to use? Does the glyph design even match my document fonts?

I suspect the most directly useful aspect of unicode maths will be relieving (most of) the headache around finding *and using* a particular math font glyph. The STIX fonts — hopefully released by the time you are reading this — will be extremely useful as a fallback font for many symbols. After all, most maths symbols are geometrically abstract enough that they do not need to be directly matched with the text font.

3.2 Readable source

Unicode maths provides the ability for maths symbols and characters to be input in unicode directly in the source file. It is now possible to write a literal α instead of having to type longhand `\alpha`. This actually doesn't improve input speed, but makes source documents far more readable and amenable to casual editing.

Only small changes to the regular LaTeX syntax are required to approach the simplicity of Murray Sargent's 'nearly plain-text encoding of mathematics' [5].

3.3 Flexible output

Because there does not have to be a one-to-one mapping between the unicode characters in the source and the unicode glyphs in the output, the semantic differences between upright and italic Greek (and other) letters in the input source can be abstracted away. As shown in Table 1, documents are able to be typeset as per ISO standards or in a more classical TeX-like format without changing the typed mathematics. Similarly, the output style of bold characters can also be adjusted.

As the package can load fonts for maths glyphs dynamically, multiple fonts and multiple styles can be used between various characters or families of characters. Figure 1 shows an example in which the maths was typed without presentational markup, but the different characters were assigned fonts with different shades of grey. This particular example may not be very practical, but it illustrates that the system is flexible enough to accomodate a wide range of effects.

	Example	
Package option	Latin	Greek
`math-style=ISO`	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=TeX`	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=french`	$(a, z, \mathrm{B}, \mathrm{X})$	$(\alpha, \beta, \Gamma, \Xi)$
`math-style=upright`	$(\mathrm{a}, \mathrm{z}, \mathrm{B}, \mathrm{X})$	$(\alpha, \beta, \Gamma, \Xi)$

Table 1: Same source, different styles of output.

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty \mathrm{e}^{-st} f(t)\,\mathrm{d}t$$

Figure 1: Hooks make it possible to use a variety of fonts or styles — in this case, colours — for different maths characters or families of maths characters.

4 Challenges

The biggest problem I can see with the advent of unicode maths, besides fonts — in their case I believe they'll slowly start to appear now there are tools and programs to support them — is educating people into using them well.

4.1 Using the correct characters

Table 2 shows five different maths glyphs that are all triangular. And Table 3 shows the eight different slash-like glyphs; four in each direction.

Without careful documentation and good education, it may be hard for users to use the 'correct' glyphs in many occasions. The markup in TeX and LaTeX has generally steered towards presentational aspects. But, as an example, with five different choices for which triangle to choose, different authors may choose different (but visually similar) glyphs for the same meaning.

Slot	Command	Glyph	Class
`U+25B5`	`\vartriangle`	$\vartriangle$	binary
`U+25B3`	`\bigtriangleup`	$\bigtriangleup$	binary
`U+25B3`	`\triangle`	$\triangle$	ordinary
`U+2206`	`\increment`	$\increment$	ordinary
`U+0394`	`\mathup\Delta`	$\mathup{\Delta}$	ordinary

Table 2: Four triangular glyphs with different meanings but similar shapes.

Slot	Name	Glyph	Command
U+002F	SOLIDUS	/	\slash
U+2044	FRACTION SLASH	⁄	\fracslash
U+2215	DIVISION SLASH	∕	\divslash
U+29F8	BIG SOLIDUS	⧸	\xsol
U+005C	REVERSE SOLIDUS	\	\backslash
U+2216	SET MINUS	∖	\smallsetminus
U+29F5	REVERSE SOLIDUS OPERATOR	⧵	\setminus
U+29F9	BIG REVERSE SOLIDUS	⧹	\xbsol

Table 3: The four slash-like glyphs in each direction.

My feelings are that new tools will be needed to encode mathematics more semantically. But such tools will need to be specific for each scientific field that uses different notation. This is an open problem. On the other hand, in the end how much does it really matter if my triangle is a different size to yours if the meaning is clear? (I'm not endorsing this line of thinking, just raising the question.)

4.2 Backwards compatibility vs. future compatibility

The two 'set minus' characters in Table 3 inherit their names from Plain TeX and the amssymb package, respectively. U+2216 is `\smallsetminus` and U+29F5 is `\setminus`. However, MathML does it differently: U+2216 is referred to by `setminus` *as well as* `smallsetminus` (among other synonyms); U+29F5 is as-yet unnamed [3]. This might make it difficult to move between MathML and LaTeX. Luckily these sorts of conflicts are few.

5 Summary

This extended abstract is an introduction and short summary of what I'll be talking about at TUG 2010. I'm looking forward to seeing you here.

Bibliography

[1] Barbara Beeton. Unicode and math, a combination whose time has come – Finally! *TUGboat*, 21(3):176–185, September 2000. URL http://tug.org/TUGboat/Articles/tb21-3/tb68beet.pdf.

[2] Barbara Beeton, Asmus Freytag, and Murray Sargent, III. Unicode support for mathematics. Unicode Technical Note 25 Version 9, Unicode, Inc., 2008. URL http://www.unicode.org/reports/tr25.

[3] David Carlisle and Patrick Ion. XML entity definitions for characters. Technical Report W3C Working Draft 21, W3C, 2008. URL http://www.w3.org/TR/xml-entity-names/.

[4] Jonathan Kew. XeTeX live. *TUGboat*, 29 (1):146–150, 2007. URL http://tug.org/TUGboat/Articles/tb29-1/tb91kew.pdf.

[5] Murray Sargent, III. Unicode nearly plaintext encoding of mathematics. Unicode technical note 28, Unicode, Inc., 2006. URL http://www.unicode.org/notes/tn28/.

[6] Ulrik Vieth. Do we need a 'cork' math font encoding? *TUGboat*, 29(3):426–434, 2008. URL http://tug.org/TUGboat/Articles/tb29-3/tb93vieth.pdf.