Two Linear Formats Interoperable with MathML

Murray Sargent III
Microsoft
murrays@microsoft.com

Abstract

This talk compares two linear-format input methods for keying in mathematical equations along with other approaches involving handwriting, menus, toolbars and ribbons. The linear format methods are 1) that used in Microsoft Office 2007, and 2) a version of [La]TeX's math input with appropriate extensions and conventions for interoperating with presentation MathML and MS Office's OMML. Demonstrations will be given revealing how formula autobuildup together with WYSIWYG editing simplify and streamline equation entry.

1. Introduction

MathML[1] has been designed for machine representation of mathematics and is useful for interchange between mathematical applications as well as for rendering the math in technical documents. While very good for these purposes, MathML is awkward for direct human input. Hence it's desirable to have more user friendly ways of inputting mathematical expressions and equations. In this paper we describe two linear-format input methods for this purpose. The first is the Unicode linear format[2] used by Word 2007 and the second is a version of [La]TeX[3,4] mathematical notation that maps well to Presentation MathML. For the purposes of this paper, we refer to the first as UnicodeMath and the second as MathTeX. This paper only provides quick overviews of the formats. More detailed specifications are given in the references. Many other input formats for mathematics exist some of which are discussed in the workshop The Evolution of Mathematical Communication in the Age of Digital Libraries.

UnicodeMath is an outgrowth of the PS Technical Word Processor input format, which started with a C-like syntax back in 1980. It's more like a real math notation than TeX, and borrows from [La]TeX when more mathematical notation not obvious.

Both linear formats are primarily concerned with presentation, but they have some semantic features that might seem to be only content oriented, e.g., *n*-aryands and function-apply arguments. These are included to aid in displaying built-up functions with proper typography and to aid in interoperating with math-oriented programs. UnicodeMath explicitly includes these concepts, whereas MathTeX includes special conventions for TeX's syntax. Although Presentation MathML doesn't have all of the semantics of Content MathML, it does have more semantics than [La]TeX. Accordingly we add some control words and conventions to the usual TeX descriptions of mathematics. For example, we need to know the base of a superscript or subscript, the integrand of an integral and the argument(s) of a mathematical func-

tion. In addition to facilitating interchange with computational applications, this extra knowledge reduces the need for user tweaks to achieve correct mathematical spacing.

A variety of syntax choices can be used for a linear format. UnicodeMath favors a number of criteria: efficient input of mathematical formulae, sufficient generality to support high-quality mathematical typography, the ability to round trip elegant mathematical text at least in a rich-text environment, and a format that resembles a real mathematical notation. Obviously compromises between these goals had to be made. MathTeX favors [La]TeX syntax, which is widely used. Usually UnicodeMath is more compact and easy to read than MathTeX, but less common mathematical constructs force the use of more complicated syntax.

UnicodeMath doesn't attempt to include all typographical embellishments. Instead it delegates some embellishments in the higher-level layer that handles rich text properties like text and background colors, font size, footnotes, comments, hyperlinks, etc. In particular, math italic and boldface attributes are handled by the application's tools[7] for that purpose. Officially UnicodeMath is defined by Unicode character codes, not by ASCII aliases such as \beta for β. This choice implies that UnicodeMath is globalized, whereas the standard ASCII aliases have a strong English bias, thanks to their TeX origins.

MathTeX doesn't attempt to include all of [La]TeX's mathematical notation. In particular it doesn't include a general TeX macro facility since macros impede or even preclude interoperability. Simple substitution macros can be expanded and the expanded results used in information interchange. The input characters for both formats can be all ASCII, but we assume that the input is targeted for use by Unicode[5,6] rendering and computational systems. In fact, typical interactive implementations translate to Unicode as soon as the input control words are recognized. Also LaTeX environments aren't used, since they're relatively complicated from an input method point of view.

A key concept that dramatically reduces the effort of math input is formula autobuildup.[8] Build up occurs when the user enters a character that is unambiguously not part of the previous mathematical expression(s) according to the linear-format rules. As such the character itself is not part of the resulting built-up expression. With this feature, the user seldom sees much linear format when entering math; most of the mathematics is built up on screen. The user can edit the built-up mathematics directly or with additional linear format input.

2. Encoding Simple Math Expressions

This section introduces UnicodeMath and MathTeX with discussions on general syntax, fractions, subscripts, superscripts, delimiters, integrals and other common 2D mathematical constructs. It includes a subsection on how the ASCII space character U+0020 is used. First it's helpful to understand some basic notation.

Both approaches assume math occurs inside math zones. In TeX these are traditionally delimited by $'s when in line with nonmathematical text and by double $$'s for displayed equations. For UnicodeMath and MathTeX, we assume some way of delimiting math zones is used, but allow it to be abstracted as a character format property in place of the usual $ and $$. Hence we assume the math expressions in this document are in math zones and we don't display the math zone delimiters explicitly. The keyboard input is in ASCII, with nonASCII characters written as control words that start with a backslash. For example, the Greek letter β is input using the control word \beta.

An argument of a TeX math object consists either of a single character (which can be given by a control word) or multiple characters enclosed in curly braces {}. The only exception to this rule is the fraction \over kind of construct discussed next. Such curly braces are not displayed in built-up form. Since the ASCII curly braces are used for grouping, they need special control words \{ and \} to be used as math characters. Some other ASCII operators like _ and ^ are used as part of the syntax and cannot be input directly.

UnicodeMath defines a simple operand to consist of all consecutive letters and decimal digits, i.e., a span of alphanumeric characters, those belonging to the L*x* and Nd General Categories (see *The Unicode Standard* 5.0,[1] Table 4-2. General Category). As such, a simple numerator or denominator is terminated by most nonalphanumeric characters, including, for example, arithmetic operators, the blank (U+0020), and Unicode characters in the ranges U+2200—U+23FF, U+2500—U+27FF, and U+2900—U+2AFF.

2.1 Fractions

The original way to specify a fraction in TeX is using the \over control word. This has the syntax

{*numerator*\over *denominator*}

If the fraction is the only expression in the math zone, the enclosing {} can be omitted. Alternatively one can use LaTeX's \frac{*numerator*}{*denominator*}. With either fraction syntax, the { } are not printed when the fraction is built up. For example, the simple built-up fraction

$$\frac{abc}{d}$$

is given by {abc\over d} or equivalently by \frac{abc}{d} in MathTeX. It is given by abc/d in UnicodeMath.

For more complex operands (such as those that include operators), UnicodeMath needs parentheses ( ), brackets [ ], or braces { } to enclose the desired character combinations. If parentheses are used and the outermost parentheses are preceded and followed by operators, those parentheses are not displayed in built-up form, since usually one does not want to see such parentheses. So the plain text $(a + c)/d$ displays as

$$\frac{a+c}{d}.$$

while in MathTeX, it is given by {a+c\over d} or equivalently by \frac{a+c}{d}.

In Presentation MathML this reads as

```
<mfrac>
    <mrow>
        <mi>a</mi>
        <mo>+</mo>
        <mi>c</mi>
    </mrow>
    <mi>d</mi>
</mfrac>
```

The same notational syntax is used for a "stack" which is like a fraction with no fraction bar. The stack is used to create binomial coefficients and the stack control word is \atop, which works similarly to \over. For example, the binomial theorem

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$$

can be input in MathTeX by

```
(a+b)^n=\sum_{k=0}^n{\left(n\atop k\right)a^kb^{n-k}}
```

where \left(n\atop k\right) is the binomial coefficient for the combinations of n items grouped k at a time. The summation limits use the subscript/superscript notation discussed in the next subsection. Prefixing the opening parenthesis with \left and the closing parenthesis with \right causes them to grow with the size of their argument. Since binomial coefficients are quite common, TeX has the \choose control word for them. Accordingly the binomial coefficient in the binomial theorem above can be written a little more simply as "{n\choose k}". In AmSTeX, one uses \binom{n}{k} for this.

In UnicodeMath, the binomial theorem reads as

```
(a + b)^n = ∑_(k=0)^n ▒ (n ¦ k) a^k b^(n-k),
```

where the n-aryand "glue" operator ▒ (U+2592) is entered by the control word \of and ¦ is entered by \atop.

2.2 Subscripts and Superscripts

In both UnicodeMath and MathTeX, a subscript is introduced by a subscript operator, the ASCII underscore _, as in a_2 which represents a_2. Similarly, superscripts are introduced by a superscript operator, which is the ASCII ^. So a^b represents a^b. In MathTeX if the superscript consists of more than one letter, you need to enclose it

in {}. For example, a^{b+c} is represented by a^{b+c}. In UnicodeMath this is represented by a^(b+c), where as for fractions the outermost parentheses are omitted when built up.

In these examples, the base of the superscript object is the letter a. If we have ab^c, only the b would be in the base, while {ab}^c would have ab as the base. TeX doesn't care what the base of a subscript or superscript is, but MathML does, so with our input methods we need this convention. It allows us to obtain the semantics needed for Presentation MathML, while still being compatible with [La]TeX.

As an example of a slightly more complicated example, the expression $W_{\delta\rho\sigma}^{3\beta}$ is represented in MathTeX by W^{3\beta}_{\delta\rho\sigma} and in UnicodeMath as $W^3\beta\_\delta\rho\sigma$.

2.3 Use of the Blank (Space) Character

The ASCII space character U+0020 is rarely needed for explicit spacing of built-up text since the spacing around operators should be provided automatically by the math display engine. However the space character is very useful for delimiting the operands of the linear-format notation. In TeX (and MathTeX), the space character is used to delimit control words like \alpha and does not appear in built-up form. Ordinary spaces are eliminated in built-up display since the display engine is responsible for most mathematical spacing. Extra spacing can be entered by the user using special controls like \, for a thin space.

In UnicodeMath also, when the space plays this role, it is eliminated upon build up. So if you type \alpha followed by a space to get α, the space is eliminated when the α replaces the \alpha. Similarly $a\_1\ b\_2$ builds up as a_1b_2 with no intervening space.

Another example is that a space following the denominator of a fraction is eliminated, since it causes the fraction to build up. If a space precedes the numerator of a fraction, the space is eliminated since it may be necessary to delimit the start of the numerator. However if a space isn't used to build something up, it is entered as user spacing. This choice is more user friendly than in MathTeX, but it does allow the user to mar the typography inadvertently.

2.4 Delimiters

Brackets [ ] and parentheses ( ) represent themselves in TeX. They appear in ordinary text size unless you precede the opening bracket by a special prefix like \left and the closing one by \right, in which case, they grow to fit around what's inside them. In general we refer to such characters as *delimiters*. A delimited pair need not consist of the same kinds of delimiters. For example, it's fine to open with [and close with) and one sees this usage in some mathematical documents. You don't even have to have both delimiters display. For example, "\left\{a+b\right." displays as $\{a + b$. The closing delimiter can have a subscript and/or a superscript.

UnicodeMath automatically expands the size of delimiters to fit the argument within. If a delimiter is used in a nonstandard way, e.g., a closing delimiter used as an opening delimiter, it must be preceded by an override control word (\open or \close), which are similar to TeX's \left and \right.

Since the curly braces { } are used for grouping arguments in TeX, you have to precede them by \ or \left or \right or some other modifier if you want them to display when built up. For example, "\{a+b\}^2" displays as $\{a+b\}^2$.

Absolute value bars are represented by the ASCII vertical bar | (U+007C). MathTeX needs to encode the absolute value as a delimiter object. To this end, the evenness of the vertical bar count at any given bracket nesting level typically determines whether the vertical bar is a close |. Specifically, the first appearance is considered to be an open | (unless subscripted or superscripted), the next a close | (unless following an operator), the next an open |, and so forth.

Some nested absolute values can be handled unambiguously. For example, $||x| - |y||$ can be parsed without the clarifying curly braces by noting that a vertical bar | directly following an operator is an open |. But the example $|a|b-c|d|$ may need explicit prefixes like \left and \right (or \open and \close) since it can be interpreted as either $(|a|b)-(c|d|)$ or $|a(|b-c|)d|$. The usual algorithm gives the former, so if one wants the latter, one can type |{a|b−c|d}| in MathTeX.

2.5 Prescripts

A special parenthesized syntax is used to form prescripts, that is, subscripts and superscripts that precede their base. For this _c^ba creates the prescripted variable ${}_a^b c$. Variables can have both prescripts and postscripts (ordinary subscripts and superscripts). A common use of prescripts is for the confluent hypergeometric functions, such as ${}_1F_1$. This can be input as _1F_1. UnicodeMath works similarly, except that the prescript needs to be delimited by a space. So ${}_1F_1$ is given by _1 F_1.

2.6 *n*-ary Operators

n-ary operators like integral, summation and product are sub/superscripted or above/below operators that have a third argument: the "*n*-aryand". For the integral, the *n*-aryand is the integrand, and for the summation, it's the summand. For both typographical and semantic purposes, it's useful to identify these *n*-aryands. In MathTeX, the *n*-aryand is the first entity following the possibly sub/superscripted *n*-ary operator. For example, the MathTeX expression \int_0^a{x\dd x\over x^2+a^2} has the built up form

$$\int_0^a \frac{x\,dx}{x^2+a^2}$$

where x\dd x/(x^2+a^2) is the integrand and \dd (ⅆ) is the Unicode differential character U+2146. Notice that a renderer can have the convention that the ⅆ character automatically leads to a small space between the x and the dx and by default displays as a math-italic d when it appears in a math zone. A document setting can

choose upright or italic open-face. In [La]TeX, to get the spacing shown above, one can use the ASCII d preceded by the thin space \, as in \int_0^a{x\,dx\over x^2+a^2}. In UnicodeMath, this integral is given by ∫_0^a▒x𝑑x/(x^2+a^2), where the n-aryand operator ▒ is U+2592.

Sometimes one wants to control the positions of the limit expressions explicitly as in using TeX's \limits (upper limit above, lower below) and \nolimits (upper limit as superscript and lower as subscript) control words…

2.7 Mathematical Functions

Mathematical functions such as trigonometric functions like "sin" should not be italicized and special spacing may be needed between the function name and argument and other factors. Accordingly TeX has a set of control words for the common mathematical functions such as \sin and \cos.

MathTeX includes these functions and adds the convention that the entity following the function control word contains the argument(s). For example \sin x stands for $\sin x$, where x is the argument. Similarly \sin{(\omega-\omega_0)t} stands for $\sin(\omega - \omega_0)t$, where the argument is $(\omega - \omega_0)t$. Comparing these examples, we see that $\sin x$ automatically has a thin space between the sin and x, while $\sin(\omega - \omega_0)t$ has no such space. Furthermore, the expression a\sin x displays as $a \sin x$, which has an automatic thin space between the a and the sin. This is an example of where having more semantics aids the mathematical typography. It's clear that it also aids interoperating with mathematical calculation engines. One problem is that TeX puts a space between sin and {}, although it doesn't between \sin and (). So better spacing is achieved in TeX without the {}. Presumably the function macros can be redefined to eliminate the extra space.

In UnicodeMath, the functions above are automatically recognized. In Word 2007, the user can edit the choices of function names.

In addition to the most common mathematical functions, there are many others and there are variants of the common ones as well, such as asin and $\sin^{-1}$ for arcsin. To handle functions in general, MathTeX and UnicodeMath can follow an arbitration name by the Invisible Function Apply operator U+2061 (\funcapply). This is a special binary operator and the operand that follows it is the function argument. In converting to built-up form, this operator transforms its operands into a two-argument object that renders with the proper spacing for mathematical functions. For example, asin\funcapply x renders as asin x and x is the argument.

If the Function Apply operator is immediately followed by a subscript or superscript expression, that expression should be applied to the function name and the Function Apply operator moved passed the modified name to bind the operand that follows as the function argument. For example, the function $\sin^2 x$ falls into this category.

Since \funcapply isn't the most intuitive name, \of can be used in function-apply contexts. This alias is motivated by sentences like "The sine of 2x equals twice the sine of x times the cosine of x", i.e., $\sin 2x = 2 \sin x \cos x$.

2.8 Square Roots and Radicals

A square root is represented by \sqrt followed by an entity that comprises the radicand. Examples are $\sqrt{a}$ and $\sqrt{\{a+b\}}$, which display as $\sqrt{a}$ and $\sqrt{a+b}$, respectively. In general, the *n*th root radical is represented by an expression like \root *degree*\of *radicand*. For example, you can obtain $\sqrt[n]{a+b}$ using \root n\of{a+b}. In this format, the degree of the radical can be more than one character without enclosing it in curly braces. For example, $\sqrt[n+1]{b+c}$ can be input by \root n+1\of{b+c}. The same input works for UnicodeMath, except that the { } are replaced by ( ).

2.9 Matrices

Matrices are represented in MathTeX by an expression of the form

$$\text{\matrix\{}exp_1\ [\&\ exp_2]\ldots\ \text{\cr}\ \ldots\ exp_{n\text{-}1}\ [\&\ exp_n]\ldots\ \}$$

where & separates columns and \cr terminates rows. This causes exp_1 to be aligned over $exp_{n\text{-}1}$, etc., to build up an $n \times m$ matrix array, where n is the maximum number of elements in a row and m is the number of rows. The matrix is constructed with enough columns to accommodate the row with the largest number of entries, with rows having fewer entries given sufficient null entries to keep the table $n \times m$. As an example, \matrix{a&b\cr c&d\cr} displays as

$$\begin{matrix} a & b \\ c & d \end{matrix}$$

UnicodeMath represents matrices the same way except that \cr is replaced by @, the { } are replaced by ( ) and no final @ precedes the closing parenthesis. Hence the matrix above is represented by \matrix(a&b@c&d), where \matrix operator translates to ■ (U+25A0).

To enclose a matrix in parentheses, include it inside parentheses as in \left(\matrix{a&b\cr c&d\cr}\right). Because parenthesized matrices are quite common, UnicodeMath and MathTeX have the \pmatrix control word that automatically includes parentheses. So in MathTeX \pmatrix{a&b\cr c&d\cr} displays as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

2.10 Accent Operators

Mathematical text often has accented characters. Simple primed characters like a' are represented by the character followed by the ASCII apostrophe '. Double primed characters have two apostrophes, etc. Typical software automatically transforms the apostrophe(s) to superscripted Unicode prime character(s) (U+2032). The primes are special in that they need to be superscripted with appropriate use of heavier glyph variants. The ASCII asterisk is raised in ordinary text, but in a math zone it gets translated into U+2217, which is placed on the math axis as the +. To

make it a superscript or subscript, the user has to include it in a superscript or subscript expression. For example, a^{*2} is represented by a^{*2}.

Other kinds of accented characters can be represented by accent control words followed by the accent base inside { }. For example in MathTeX, $\hat{a}$ is given by \hat{a}. Since combining marks follow their base in Unicode, in UnicodeMath $\hat{a}$ is given by a\hat. If an accent should be applied to more than one character or to an expression, enclose those characters or expression in curly braces following the accent control word. For example, in MathTeX, \hat{a+b} renders as $\widehat{a+b}$. In [La]TeX, this is given by \widehat{a+b}. In UnicodeMath it's given by (a+b)\hat.

References

1. *Mathematical Markup Language (MathML) Version 2.0* (Second Edition) http://www.w3.org/TR/2003/REC-MathML2-20031021/ .
2. Murray Sargent III, Unicode Technical Note #28, "Unicode Nearly Plain-Text Encoding of Mathematics", http://www.unicode.org/notes/tn28/UTN28-PlainTextMath-v2.pdf. This linear format is used for keyboard entry of mathematical expressions in Microsoft Word 2007 and the Microsoft Math graphing calculator.
3. Donald E. Knuth, *The TeXbook*, (Reading, Massachusetts: Addison-Wesley 1984)
4. Leslie Lamport, *LaTeX: A Document Preparation System, User's Guide & Reference Manual*, 2nd edition (Addison-Wesley, 1994; ISBN 1-201-52983-1)
5. *The Unicode Standard*, Version 5.0, (Reading, MA, Addison-Wesley, 2006. ISBN 0-321-18578-1) or see online http://www.unicode.org/versions/Unicode4.0.0/
6. Barbara Beeton, Asmus Freytag, Murray Sargent III, Unicode Technical Report #25 "Unicode Support for Mathematics", http://www.unicode.org/reports/tr25.
7. M. Sargent III blog post Using Math Italic and Bold in Word 2007.
8. M. Sargent III blog posts Formula Autobuildup in Word 2007 and When Formula Autobuildup Occurs.

This document was prepared using Microsoft Word 2007 with Cambria and Cambria Math fonts.