

Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2008/08/27 v0.3b

Abstract

Warning! This package is experimental and subject to change without regard for backwards compatibility.

Contents

1	Introduction	2	5.3	Other things	18
2	Specification	2	6	Fundamentals	19
2.1	Using multiple fonts	3	6.1	Enlarging the number of maths families	19
2.2	Script and scriptscript fonts/features	3	6.2	<code>\DeclareMathSymbol</code> for unicode ranges	19
3	Maths input	3	6.3	User interface to <code>\DeclareSymbolFont</code>	21
4	Package options	4	6.4	Maths alphabets' character mapping	24
4.1	Math 'style'	4	6.5	(Big) operators	29
4.2	Bold switching	5	6.6	Radicals	33
4.3	Other upright vs. italic symbols	5	6.7	Delimiters	34
4.4	Math font dimensions	6	6.8	Maths accents	36
I	The unicode-math package	12	7	Font features	37
5	Things we need	12	7.1	OpenType maths font features	38
5.1	Package options	16	7.2	Script and scriptscript font options	38
5.2	Overcoming <code>\@on-lypreamble</code>	17	7.3	Range processing	38

II Maths alphabets mapping definitions	44	the NFSS	55
7.4 Bold alphabets' character mappings	48	A.1 Overview	55
III stix table data extraction	54	IV Some manner of unit testing	57
A Documenting maths support in		B The regular weight alphabets	57
		C The bold alphabets	58

1 Introduction

This document describes the unicode-math package, which is an *experimental* implementation of a macro to unicode glyph encoding for mathematical characters. Its intended use is for X_YTeX, although it is conjectured that small effect needs to be spent to create a cross-format package that would also work with LuaTeX.

As of X_YTeX v.0.995, maths characters can be accessed in unicode ranges. Now, a proper method must be invented for real unicode maths support. Fairly significant pieces of the NFSS may have to be re-written, and I'm a little unsure where to start.

2 Specification

This section will turn into 'User Interface' in time, presumably.

In the ideal case, a single unicode font will contain all maths glyphs we need. Barbara Beeton's STIX table provides the mapping between unicode maths glyphs and macro names (all 3298 — or however many — of them!). A single command

```
\setmathfont[⟨font features⟩]{⟨font name⟩}
```

would implement this for every every symbol and alphabetic variant. That means x to x , ξ to ξ , \leq to \leq , etc., \mathcal{H} to \mathcal{H} and so on, all for unicode glyphs within a single font.

Furthermore, this package should deal well with unicode characters for maths input, as well. This includes using literal Greek letters in formulae, resolving to upright or italic depending on preference.

Finally, maths versions must also be provided for. While I guess version selection in L^ATeX will remain the same, the specification for choosing the version fonts will probably be an optional argument:

```
\setmathfont[Version=Bold,⟨font features⟩]{⟨font name⟩}
```

Instances above of

```
[⟨font features⟩]{⟨font name⟩}
```

follow from my fontspec package, and therefore any additional *font features* specific to maths fonts will hook into fontspec’s methods.

2.1 Using multiple fonts

There will probably be few cases where a single unicode maths font suffices (simply due to glyph coverage). The upcoming STIX font comes to mind as a possible exception. It will therefore be necessary to delegate specific unicode ranges of glyphs to separate fonts. This syntax will also hook into the fontspec font feature processing:

`\setmathfont[Range=<unicode range>,]{}`

where *<unicode range>* is a comma-separated list of unicode slots and ranges such as {27D0-27EB,27FF,295B-297F}. Furthermore, preset names ranges could be used, such as `MiscMathSymbolsA`, with such ranges based on unicode chunks. The amount of optimisation required here to achieve acceptable performance has yet to be determined. Techniques such as saving out unicode subsets based on *<unicode range>* data to be `\input` in the next \LaTeX run are a possibility, but at this stage, performance without such measures seems acceptable.

2.2 Script and scriptscript fonts/features

Cambria Math uses OpenType font features to activate smaller optical sizes for `scriptsize` and `scriptscriptsize` symbols (the *B* and *C*, respectively, in A_{B_C}).

Other fonts will possibly use entirely separate fonts. Both of these options must be taken into account. I hope this will be mostly automatic from the users’ points of view. The `+ssty` feature can be detected and applied automatically, and appropriate optical size information embedded in the fonts will ensure this latter case. Fine tuning should be possible automatically with fontspec options. We might have to wait until MnMath, for example, before we really know.

3 Maths input

\XTeX ’s unicode support allows maths input through two methods. Like classical \TeX , macros such as `\alpha`, `\sum`, `\pm`, `\leq`, and so on, provide verbose access to the entire repertoire of characters defined by unicode. The literal characters themselves may be used instead, for more readable input files.

: TODO : describe alphabet inputs

Table 1: Effects of the `math-style` package option.

Package option	Example	
	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=ISO</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=TeX</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=French</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$

4 Package options

4.1 Math ‘style’

Classically, \LaTeX uses italic lowercase Greek letters and *upright* uppercase Greek letters for variables in mathematics. This is contrary to the ISO standards of using italic forms for both upper- and lowercase. Furthermore, the French (contrary again, *quelle surprise*) have been known to use upright uppercase *Latin* letters as well as upright upper- and lowercase Greek.

The `unicode-math` package accommodates these possibilities with an interface heavily inspired by Walter Schmidt’s `lucimatx` package: a package option `math-style` that takes one of three arguments: `TeX`, `ISO`, or `French` (case *in*-sensitive).

The philosophy behind the interface to the mathematical alphabet symbols lies in \LaTeX ’s attempt of separating content and formatting. Because input source text may come from a variety of places, the upright and ‘mathematical’ italic Latin and Greek alphabets are *unified* from the point of view of having a specified meaning in the source text. That is, to get a mathematical ‘ x ’, either the `ascii` (‘keyboard’) letter `x` may be typed, or the actual unicode character may be used. Similarly for Greek letters. The upright or italic forms are then chosen based on the `math-style` package option.

If glyphs are desired that do not map as per the package option (for example, an upright ‘ g ’ is desired but typing `g` yields ‘ g ’), *markup* is required to specify this; to follow from the example: `\mathup{g}`. Maths alphabets commands such as `\mathup` are detailed later.

Alternative interface However, some users may not like this convention. For them, an upright `x` is an upright ‘`x`’ and that’s that. (This will be the case when obtaining source text from copy/pasting PDF or Microsoft Word documents, for example.) For these users, the `literal` option to `math-style` will effect this behaviour.

The `math-style` options’ effects are shown in brief in table 1. Table ?? on page ?? shows every character under the effect of this package option.

Table 2: Effects of the bold-style package option.

Package option	Example	
	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>bold-style=ISO</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=TeX</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=French</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$

4.2 Bold switching

Similar as in the previous section, ISO standards differ somewhat to $\mathrm{T}_{\mathrm{E}}\mathrm{X}$'s conventions (and classical typesetting) for 'boldness' in mathematics. In the past, it has been customary to use bold *upright* letters to denote things like vectors and matrices. For example, $\mathbf{M} = (M_x, M_y, M_z)$. Presumably, this was due to the relatively scarcity of bold italic fonts in the pre-digital typesetting era. It has been suggested that *italic* bold symbols are used nowadays instead.

Bold Greek letters have simply been bold variant glyphs of their regular weight, as in $\boldsymbol{\xi} = (\xi_r, \xi_\varphi, \xi_\theta)$. Confusingly, the syntax in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ has been different for these two examples: `\mathbf{bf}` in the former (\mathbf{M}), and `\bm` (or `\boldsymbol`, deprecated) in the latter ($\boldsymbol{\xi}$).

In `unicode-math`, the `\mathbf{bf}` command works directly with both Greek and Latin maths alphabet characters and depending on package option either switches to upright for Latin letters (`bold-style=TeX`) as well or keeps them italic (`bold-style=ISO`).

To match the package options for non-bold characters, for `bold-style=French` all bold characters are upright, and `bold-style=literal` does not change the upright/italic shape of the letter.

Upright and italic bold mathematical letters input as direct unicode characters are normalised with the same rules. For example, with `bold-style=TeX`, a literal bold italic latin character will be typeset upright.

Note that `bold-style` is independent of `math-style`, although if the former is not specified then sensible defaults are chosen based on the latter. The `bold-style` options' effects are shown in brief in table 2. Table ?? on page ?? shows every character under the effect of this package option.

4.3 Other upright vs. italic symbols

Nabla The symbol ∇ comes in the six forms shown in table 3. We want an individual option to specify whether we want upright or italic nabla by default (when either upright or italic nabla is used in the source). $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ classically uses an upright nabla, but ISO standards differ (I think). The package options `nabla=upright`

Table 3: The various forms of nabla.

Description		Glyph
Upright	Serif	∇
	Bold serif	$\boldsymbol{\nabla}$
	Bold sans	∇
Italic	Serif	∇
	Bold serif	$\boldsymbol{\nabla}$
	Bold sans	∇

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρρςστυφφχψω

(a) Package option `[math-style=ISO]`

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρρςστυφφχψω

(b) Package option `[math-style=TeX]`

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρρςστυφφχψω

(c) Package option `[math-style=French]`

Figure 1: Example maths output demonstrating the `math-style` package option.

and `nabla=italic` switch between the two choices. This is then inherited through `\mathbf`; `\mathit` and `\mathup` can be used to force one way or the other.

`nabla=italic` is implicit when using `math-style=ISO` and `nabla=upright` follows both `math-style=TeX` and `math-style=French`.

Partial Ditto with ∂ : `partial=upright` and `partial=italic` package options. Similarly with the `math-style` defaults.

4.4 Math font dimensions

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

(a) Package option [bold-style=ISO]

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

(b) Package option [bold-style=TeX]

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

(c) Package option [bold-style=French]

Figure 2: Example maths output demonstrating the bold-style package option.

\fontdimen	Dimension name	Description
10	SCRIPTPERCENTSCALEDOWN	Percentage of scaling down for script level 1. Suggested value: 80%.
11	SCRIPTSCRIPTPERCENTSCALEDOWN	Percentage of scaling down for script level 2 (ScriptScript). Suggested value: 60%.
12	DELIMITEDSUBFORMULA-MINHEIGHT	Minimum height required for a delimited expression to be treated as a subformula. Suggested value: normal line height $\times 1.5$.
13	DISPLAYOPERATORMIN-HEIGHT	Minimum height of n-ary operators (such as integral and summation) for formulas in display mode.
14	MATHLEADING	White space to be left between math formulas to ensure proper line spacing. For example, for applications that treat line gap as a part of line ascender, formulas with ink going above (os2.sTypoAscender + os2.sTypoLineGap – MathLeading) or with ink going below os2.sTypoDescender will result in increasing line height.
15	AXISHEIGHT	Axis height of the font.

\fontdimen	Dimension name	Description
16	ACCENTBASEHEIGHT	Maximum (ink) height of accent base that does not require raising the accents. Suggested: x-height of the font (os2.sxHeight) plus any possible overshoots.
17	FLATTENEDACCENTBASE- HEIGHT	Maximum (ink) height of accent base that does not require flattening the accents. Suggested: cap height of the font (os2.sCapHeight).
18	SUBSCRIPTSHIFTDOWN	The standard shift down applied to subscript elements. Positive for moving in the downward direction. Suggested: os2.ySubscriptYOffset.
19	SUBSCRIPTTOPMAX	Maximum allowed height of the (ink) top of subscripts that does not require moving subscripts further down. Suggested: /5 x-height.
20	SUBSCRIPTBASELINEDROP- MIN	Minimum allowed drop of the baseline of subscripts relative to the (ink) bottom of the base. Checked for bases that are treated as a box or extended shape. Positive for subscript baseline dropped below the base bottom.
21	SUPERSCRIPSHIFTUP	Standard shift up applied to superscript elements. Suggested: os2.ySuperscriptYOffset.
22	SUPERSCRIPSHIFTUP- CRAMPED	Standard shift of superscripts relative to the base, in cramped style.
23	SUPERSCRIPBOTTOMMIN	Minimum allowed height of the (ink) bottom of superscripts that does not require moving subscripts further up. Suggested: ¼ x-height.
24	SUPERSCRIPBASELINEDROP- MAX	Maximum allowed drop of the baseline of superscripts relative to the (ink) top of the base. Checked for bases that are treated as a box or extended shape. Positive for superscript baseline below the base top.
25	SUBSUPERSCRIPGAPMIN	Minimum gap between the superscript and subscript ink. Suggested: 4×default rule thickness.

\fontdimen	Dimension name	Description
26	SUPERSCRIPTBOTTOMMAX-WITHSUBSCRIPT	The maximum level to which the (ink) bottom of superscript can be pushed to increase the gap between superscript and subscript, before subscript starts being moved down. Suggested: $\frac{1}{5}$ x-height.
27	SPACEAFTERScript	Extra white space to be added after each subscript and superscript. Suggested: 0.5pt for a 12 pt font.
28	UPPERLIMITGAPMIN	Minimum gap between the (ink) bottom of the upper limit, and the (ink) top of the base operator.
29	UPPERLIMITBASELINERISE-MIN	Minimum distance between baseline of upper limit and (ink) top of the base operator.
30	LOWERLIMITGAPMIN	Minimum gap between (ink) top of the lower limit, and (ink) bottom of the base operator.
31	LOWERLIMITBASELINE DROP-MIN	Minimum distance between baseline of the lower limit and (ink) bottom of the base operator.
32	STACKTOPSHIFTUP	Standard shift up applied to the top element of a stack.
33	STACKTOPDISPLAYSTYLE-SHIFTUP	Standard shift up applied to the top element of a stack in display style.
34	STACKBOTTOMSHIFTDOWN	Standard shift down applied to the bottom element of a stack. Positive for moving in the downward direction.
35	STACKBOTTOMDISPLAY-STYLESHIFTDOWN	Standard shift down applied to the bottom element of a stack in display style. Positive for moving in the downward direction.
36	STACKGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element. Suggested: $3 \times$ default rule thickness.
37	STACKDISPLAYSTYLEGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element in display style. Suggested: $7 \times$ default rule thickness.
38	STRETCHSTACKTOPSHIFTUP	Standard shift up applied to the top element of the stretch stack.

\fontdimen	Dimension name	Description
39	STRETCHSTACKBOTTOM-SHIFTDOWN	Standard shift down applied to the bottom element of the stretch stack. Positive for moving in the downward direction.
40	STRETCHSTACKGAPABOVE-MIN	Minimum gap between the ink of the stretched element, and the (ink) bottom of the element above. Suggested: UpperLimitGapMin
41	STRETCHSTACKGAPBELOW-MIN	Minimum gap between the ink of the stretched element, and the (ink) top of the element below. Suggested: LowerLimitGapMin.
42	FRACTIONNUMERATOR-SHIFTUP	Standard shift up applied to the numerator.
43	FRACTIONNUMERATOR-DISPLAYSTYLESHIFTUP	Standard shift up applied to the numerator in display style. Suggested: StackTopDisplayStyleShiftUp.
44	FRACTIONDENOMINATOR-SHIFTDOWN	Standard shift down applied to the denominator. Positive for moving in the downward direction.
45	FRACTIONDENOMINATOR-DISPLAYSTYLESHIFTDOWN	Standard shift down applied to the denominator in display style. Positive for moving in the downward direction. Suggested: StackBottomDisplayStyleShiftDown.
46	FRACTIONNUMERATORGAP-MIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar. Suggested: default rule thickness
47	FRACTIONNUMDISPLAY-STYLEGAPMIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
48	FRACTIONRULETHICKNESS	Thickness of the fraction bar. Suggested: default rule thickness.
49	FRACTIONDENOMINATOR-GAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar. Suggested: default rule thickness

\fontdimen	Dimension name	Description
50	FRACTIONDENOMDISPLAY- STYLEGAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
51	SKEWEDFRACTION- HORIZONTALGAP	Horizontal distance between the top and bottom elements of a skewed fraction.
52	SKEWEDFRACTIONVERTICAL- GAP	Vertical distance between the ink of the top and bottom elements of a skewed fraction.
53	OVERBARVERTICALGAP	Distance between the overbar and the (ink) top of the base. Suggested: 3×default rule thickness.
54	OVERBARRULETHICKNESS	Thickness of overbar. Suggested: default rule thickness.
55	OVERBAREXTRAASCENDER	Extra white space reserved above the overbar. Suggested: default rule thickness.
56	UNDERBARVERTICALGAP	Distance between underbar and (ink) bottom of the base. Suggested: 3×default rule thickness.
57	UNDERBARRULETHICKNESS	Thickness of underbar. Suggested: default rule thickness.
58	UNDERBAREXTRA- DESCENDER	Extra white space reserved below the underbar. Always positive. Suggested: default rule thickness.
59	RADICALVERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: 1¼ default rule thickness.
60	RADICALDISPLAYSTYLE- VERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: default rule thickness + ¼ x-height.
61	RADICALRULETHICKNESS	Thickness of the radical rule. This is the thickness of the rule in designed or constructed radical signs. Suggested: default rule thickness.
62	RADICALEXTRAASCENDER	Extra white space reserved above the radical. Suggested: RadicalRuleThickness.
63	RADICALKERNBEFORE- DEGREE	Extra horizontal kern before the degree of a radical, if such is present. Suggested: 5/18 of em.

<code>\fontdimen</code>	Dimension name	Description
64	<code>RADICALKERNAFTERDEGREE</code>	Negative kern after the degree of a radical, if such is present. Suggested: $-10/18$ of em.
65	<code>RADICALDEGREEBOTTOM-RAISEPERCENT</code>	Height of the bottom of the radical degree, if such is present, in proportion to the ascender of the radical sign. Suggested: 60%.

File I

The unicode-math package

This is the package.

```

1 \ProvidesPackage{unicode-math}
2 [2008/08/27 v0.3b Unicode maths in XeLaTeX]
```

5 Things we need

Packages

```

3 \RequirePackage{fontspec}
```

Counters and conditionals

```

4 \newcounter{um@fam}
5 \newif\if@um@fontspec@feature
6 \newif\if@um@ot@math@
7 \newif\if@um@init
```

For math-style:

```

8 \newif\if@um@literal
9 \newif\if@um@upGreek
10 \newif\if@um@upgreek
11 \newif\if@um@upLatin
12 \newif\if@um@uplatin
```

For bold-style:

```

13 \newif\if@um@bfliteral
14 \newif\if@um@bfupGreek
15 \newif\if@um@bfupgreek
16 \newif\if@um@bfupLatin
17 \newif\if@um@bfuplatin
```

For nabla and partial:

```

18 \newif\if@um@upNabla
19 \newif\if@um@uppartial
```

Programming niceties

```
\def@cn
20 \providecommand\def@cn[2]{%
21 \expandafter\def\csname#1\endcsname{#2}}

\um@Loop See Kees van der Laan's various articles on TEX programming:
\um@Break 22 \def\um@Loop#1\um@Pool{#1\um@Loop#1\um@Pool}
23 \def\um@Break#1\um@Pool{}

\um@FOR A simple 'for' loop implemented with the above. Takes a (predefined) counter
control sequence and increments it between two integers, iterating as we go.
24 \long\def\um@FOR#1=[#2:#3]\do#4{%
25 #1=#2\relax
26 \um@Loop #4%
27 \advance#1@ne
28 \ifnum#1>#3\relax
29 \expandafter\um@Break
30 \fi
31 \um@Pool}
```

[g/h/i/j/k/l/m/](#)

```
\newcount\@ii
\um@FOR\@ii=[7:13]\do{\@alph\@ii/}
```

Shortcuts

```
32 \newcommand\um@PackageError[2]{\PackageError{unicode-math}{#1}{#2}}
33 \newcommand\um@PackageWarning[1]{\PackageWarning{unicode-math}{#1}}
34 \newcommand\um@PackageInfo[1]{\PackageInfo{unicode-math}{#1}}
```

Alphabet unicode positions Before we begin, let's define the positions of the various unicode alphabets so that our code is a little more readable.¹

```
35 \def\um@usv@num{\`0}
36 \def\um@usv@upLatin{\`A}
37 \def\um@usv@uplatin{\`a}
38 \def\um@usv@itLatin{"1D434}
39 \def\um@usv@itlatin{"1D44E}
40 \def\um@usv@upGreek{"391}
41 \def\um@usv@upgreek{"3B1}
42 \def\um@usv@itGreek{"1D6E2}
43 \def\um@usv@itgreek{"1D6FC}
44 \def\um@usv@bbnum{"1D7D8}
```

¹'U.S.V.' stands for 'unicode scalar value'.

```

45 \def\um@usv@bbLatin{"1D538}
46 \def\um@usv@bbLatin{"1D552}
47 \def\um@usv@scrLatin{"1D49C}
48 \def\um@usv@scrLatin{"1D4B6}
49 \def\um@usv@frakLatin{"1D504}
50 \def\um@usv@frakLatin{"1D51E}
51 \def\um@usv@sfnun{"1D7E2}
52 \def\um@usv@sflatin{"1D5A0}
53 \def\um@usv@sflatin{"1D5BA}
54 \def\um@usv@sfitLatin{"1D608}
55 \def\um@usv@sfitLatin{"1D622}
56 \def\um@usv@ttnum{"1D7F6}
57 \def\um@usv@ttLatin{"1D670}
58 \def\um@usv@ttLatin{"1D68A}

```

Bold:

```

59 \def\um@usv@bfnum{"1D7CE}
60 \def\um@usv@bfLatin{"1D400}
61 \def\um@usv@bfLatin{"1D41A}
62 \def\um@usv@bfGreek{"1D6A8}
63 \def\um@usv@bfGreek{"1D6C2}
64 \def\um@usv@bfitLatin{"1D468}
65 \def\um@usv@bfitLatin{"1D482}
66 \def\um@usv@bfitGreek{"1D71C}
67 \def\um@usv@bfitGreek{"1D736}
68 \def\um@usv@bffrakLatin{"1D56C}
69 \def\um@usv@bffrakLatin{"1D586}
70 \def\um@usv@bfscrLatin{"1D4D0}
71 \def\um@usv@bfscrLatin{"1D4EA}
72 \def\um@usv@bfsfnun{"1D7EC}
73 \def\um@usv@bfsfLatin{"1D5D4}
74 \def\um@usv@bfsfLatin{"1D5EE}
75 \def\um@usv@bfsfGreek{"1D756}
76 \def\um@usv@bfsfGreek{"1D770}
77 \def\um@usv@bfsfitLatin{"1D63C}
78 \def\um@usv@bfsfitLatin{"1D656}
79 \def\um@usv@bfsfitGreek{"1D790}
80 \def\um@usv@bfsfitGreek{"1D7AA}

```

Greek variants:

```

81 \def\um@usv@varTheta{"3F4}
82 \def\um@usv@Digamma{"3DC}
83 \def\um@usv@varepsilon{"3F5}
84 \def\um@usv@vartheta{"3D1}
85 \def\um@usv@varkappa{"3F0}
86 \def\um@usv@varphi{"3D5}
87 \def\um@usv@varrho{"3F1}
88 \def\um@usv@varpi{"3D6}

```

89 \def\um@usv@digamma{"3DD}

Bold:

90 \def\um@usv@bfvarTheta{"1D6B9}
 91 \def\um@usv@bfDigamma{"1D7CA}
 92 \def\um@usv@bfvarepsilon{"1D6DC}
 93 \def\um@usv@bfvartheta{"1D6DD}
 94 \def\um@usv@bfvarkappa{"1D6DE}
 95 \def\um@usv@bfvarphi{"1D6DF}
 96 \def\um@usv@bfvarrho{"1D6E0}
 97 \def\um@usv@bfvarpi{"1D6E1}
 98 \def\um@usv@bfdigamma{"1D7CB}

Italic Greek variants:

99 \def\um@usv@ith{"210E}
 100 \def\um@usv@itvarTheta{"1D6F3}
 101 \def\um@usv@itvarepsilon{"1D716}
 102 \def\um@usv@itvartheta{"1D717}
 103 \def\um@usv@itvarkappa{"1D718}
 104 \def\um@usv@itvarphi{"1D719}
 105 \def\um@usv@itvarrho{"1D71A}
 106 \def\um@usv@itvarpi{"1D71B}

Bold:

107 \def\um@usv@bfuph{"1D421}
 108 \def\um@usv@bfith{"1D489}
 109 \def\um@usv@bfitvarTheta{"1D72D}
 110 \def\um@usv@bfitvarepsilon{"1D750}
 111 \def\um@usv@bfitvartheta{"1D751}
 112 \def\um@usv@bfitvarkappa{"1D752}
 113 \def\um@usv@bfitvarphi{"1D753}
 114 \def\um@usv@bfitvarrho{"1D754}
 115 \def\um@usv@bfitvarpi{"1D755}

Nabla:

116 \def\um@usv@Nabla{"2207}
 117 \def\um@usv@itNabla{"1D6FB}
 118 \def\um@usv@bfNabla{"1D6C1}
 119 \def\um@usv@bfitNabla{"1D735}
 120 \def\um@usv@bfsfNabla{"1D76F}
 121 \def\um@usv@bfsfitNabla{"1D7A9}

Partial:

122 \def\um@usv@partial{"2202}
 123 \def\um@usv@itpartial{"1D715}
 124 \def\um@usv@bfpartial{"1D6DB}
 125 \def\um@usv@bfitpartial{"1D74F}
 126 \def\um@usv@bfsfpartial{"1D789}
 127 \def\um@usv@bfsfitpartial{"1D7C3}

5.1 Package options

xkeyval's package support is used here.

math-style

```
128 \define@choicekey*{unicode-math.sty}
129   {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{%
130   \ifcase\@tempb\relax
131     \@um@upGreekfalse
132     \@um@upgreekfalse
133     \@um@upLatinfalse
134     \@um@uplatinfalse
135     \@um@bfupGreekfalse
136     \@um@bfupgreekfalse
137     \@um@bfupLatinfalse
138     \@um@bfuplatinfalse
139     \@um@upNablafalse
140     \@um@uppartialfalse
141   \or
142     \@um@upGreektrue
143     \@um@upgreekfalse
144     \@um@upLatinfalse
145     \@um@uplatinfalse
146     \@um@bfupGreektrue
147     \@um@bfupgreekfalse
148     \@um@bfupLatintrue
149     \@um@bfuplatintrue
150     \@um@upNablatrue
151     \@um@uppartialtrue
152   \or
153     \@um@upGreektrue
154     \@um@upgreektrue
155     \@um@upLatintrue
156     \@um@uplatinfalse
157     \@um@bfupGreektrue
158     \@um@bfupgreektrue
159     \@um@bfupLatintrue
160     \@um@bfuplatintrue
161     \@um@upNablatrue
162     \@um@uppartialtrue
163   \or
164     \@um@literaltrue
165   \fi}
```

bold-style

```
166 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,french,literal}{%
```



```

167 \ifcase\@tempb\relax
168   \um@bfupGreekfalse
169   \um@bfupgreekfalse
170   \um@bfupLatinfalse
171   \um@bfuplatinfalse
172 \or
173   \um@bfupGreektrue
174   \um@bfupgreekfalse
175   \um@bfupLatintrue
176   \um@bfuplatintrue
177 \or
178   \um@bfupGreektrue
179   \um@bfupgreektrue
180   \um@bfupLatintrue
181   \um@bfuplatintrue
182 \or
183   \um@bfliteraltrue
184 \fi}

```

Symbol obliqueness

```

185 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright,italic}{%
186   \ifcase\@tempb\relax
187     \um@upNablatrue
188   \or
189     \um@upNablafalse
190   \fi}
191 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{%
192   \ifcase\@tempb\relax
193     \um@uppartialtrue
194   \or
195     \um@uppartialfalse
196   \fi}
197 \ExecuteOptionsX{math-style=iso}
198 \ProcessOptionsX

```

5.2 Overcoming \@onlypreamble

This will be refined later! Sort out which macros actually have to be removed from the \@preamblecmds token list.

```

199 \def\@preamblecmds{}
200
201 % this is from gmutils.sty:
202 \newcommand\not@onlypreamble[1]{%
203   \def\do##1{\ifx#1##1\else\noexpand\do\noexpand##1\fi}%
204   \xdef\@preamblecmds{\@preamblecmds}}

```

```

205 \def\gm@notprerr{ can be used only in preamble (\on@line)}
206 \AtBeginDocument{%
207   \def\do#1{\noexpand\do\noexpand#1}%
208   \edef\@preamblecmds{%
209     \def\noexpand\do##1{%
210       \def##1{\noexpand\PackageError{gmutils/LaTeX}%
211         {\noexpand\string##1 \noexpand\gm@notprerr}\noexpand\@eha}}%
212     \@preamblecmds}}
213 \def\nocite#1{%
214   \@bsphack{\setbox0=\hbox{\cite{#1}}}\@esphack}
215

```

5.3 Other things

`\um@fontdimen@percent` #1 : Font dimen number

`\fontdimens` 10, 11, and 65 aren't actually dimensions, they're percentage values given in units of sp. This macro takes a font dimension number and outputs the decimal value of the associated parameter.

0.73	<code>\font\tmpfont="Cambria Math"</code>
0.60	<code>\um@fontdimen@percent{10}{\tmpfont}\</code>
0.65	<code>\um@fontdimen@percent{11}{\tmpfont}\</code>
	<code>\um@fontdimen@percent{65}{\tmpfont}</code>

```

216 \def\um@fontdimen@percent#1#2{%
217   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax}

```

`\um@scaled@apply` #1 : A math style

#2 : Macro that takes a non-delimited length argument (like `\kern`)

#3 : Length control sequence to be scaled according to the math style

This macro is used to scale the lengths reported by `\fontdimen` according to the scale factor for script- and scriptscript-size objects.

```

218 \def\um@scaled@apply#1#2#3{%
219   \ifx#1\scriptstyle
220     #2\um@fontdimen@percent{10}\um@font#3%
221   \else
222     \ifx#1\scriptscriptstyle
223       #2\um@fontdimen@percent{11}\um@font#3%
224     \else
225       #2#3%
226     \fi
227   \fi}

```

6 Fundamentals

6.1 Enlarging the number of maths families

To start with, we've got a power of two as many `\fams` as before. So (from `ltxssbas.dtx`) we want to redefine

```
228 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cclvi}  
229 \let\newfam\new@mathgroup
```

This is sufficient for L^AT_EX's `\DeclareSymbolFont`-type commands to be able to define 256 named maths fonts. Now we need a new `\DeclareMathSymbol`.

6.2 `\DeclareMathSymbol` for unicode ranges

This command is a bit funny at the moment; it doesn't define the actual macro for almost all of the symbols passed to it, but it does assign the `\XeTeXmathchar`.

```
\um@mathsymbol #1 : Symbol, e.g., \alpha  
                #2 : Type, e.g., \mathalpha  
                #3 : Math font name, e.g., operators  
                #4 : Slot, e.g., "221E  
230 \def\um@mathsymbol#1#2#3#4{%  
231   \expandafter\um@set@mathsymbol\csname sym#3\endcsname#1#2{#4}}
```

The final macros that actually define the maths symbol with X_YL^AT_EX primitives.

```
\um@set@mathsymbol #1 : Symbol font number  
                   #2 : Symbol macro, e.g., \alpha  
                   #3 : Type, e.g., \mathalpha  
                   #4 : Slot, e.g., "221E  
If the symbol definition is for a macro. There are a bunch of tests to perform to  
process the various characters.  
232 \def\um@set@mathsymbol#1#2#3#4{%
```

Operators In the examples following, say we're defining for the symbol `\sum` (Σ).

```
233 \ifx\mathop#3\relax
```

In order for literal unicode characters to be used in the source and still have the correct limits behaviour, big operators are made math-active. `\unicodemathgobble` is the same as `\scantokens` but needs to not have `@` in its name because the argument goes inside a `\scantokens`.

The active math char is `\let` to the macro `\sum@op`.

```
234   \begingroup  
235   \catcode#4=\active  
236   \global\mathcode#4="8000\relax
```

```

237     \um@scanactivedef#4\@nil{\csname\string#2@op\endcsname}%
238     \endgroup

```

Some of these require a `\nolimits` suffix. This is controlled by the `\um@nolimits` macro, which contains a list of such characters. This list is checked dynamically because we're not interested in efficiency. Or something. This allows the list to be updated in the middle of a document.

Declare the plain old `mathchardef` for the control sequence `\sum@sym`.

```

239     \expandafter\global\expandafter\XeTeXmathchardef
240     \csname\string#2@sym\endcsname
241     =" \mathchar@type#3 #1 #4\relax

```

Now define `\sum@op` as `\sum@sym`, followed by `\nolimits` if necessary.

```

242     \expandafter\gdef\csname\string#2@op\endcsname{%
243     \csname\string#2@sym\endcsname
244     \expandafter\in@\expandafter#2\expandafter{\um@nolimits}%
245     \ifin@
246     \expandafter\nolimits
247     \fi}%

```

Don't forget that the actual `\sum` macro is simply defined in terms of the literal unicode symbol!

```

248     \else

```

Radicals Needs to be before the delimiters because the radical is, for some reason, `\mathopen`.

```

249     \expandafter\in@\expandafter#2\expandafter{\um@radicals,}%
250     \ifin@
251     \expandafter\gdef\csname
252     \expandafter\@gobble\string#2sign\endcsname
253     {\XeTeXradical#1 #4\relax}%
254     \else

```

Delimiters : TODO : sort out which of these three declarations are necessary! (Definitely the first, to work with `\left/\right`.)

```

255     \ifx\mathopen#3\relax
256     \gdef#2{\XeTeXdelimiter "\mathchar@type#3 #1 #4}%
257     \global\XeTeXdelcode#4=#1 #4\relax
258     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
259     \else
260     \ifx\mathclose#3\relax
261     \gdef#2{\XeTeXdelimiter "\mathchar@type#3 #1 #4}%
262     \global\XeTeXdelcode#4=#1 #4\relax
263     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
264     \else

```

Accents

```

265         \ifx\mathaccent#3\relax
266         \xdef#2{\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}%
267         \else

```

And finally, the general case. We define the unicode mathcode for the character. The macro is defined generically in terms of the unicode character.

```

268         \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
269         \fi
270     \fi
271 \fi
272 \fi
273 \fi}

```

`\SetMathCode` [For later] or if it's for a character code (just a wrapper around the primitive). Note that this declaration *isn't* global so that it can be constrained by grouping.

```

274 \newcommand\SetMathCode[4]{%
275     \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax}

```

A

```

\zf@fontspec{}\Cambria Math}
\let\glb@currsz\relax
\DeclareSymbolFont{test2}{EU1}{\zf@family}{m}{n}
\SetMathCode{65}{\mathalpha}{test2}{119860}
$A$

```

6.3 User interface to `\DeclareSymbolFont`

Here's the simplest usage:

$Ax \stackrel{\text{def}}{=} \nabla \times \mathcal{Z}$

```

\setmathfont{Cambria Math}
$Ax \eqdef \nabla \times \mathscrZ$

```

TODO And an example of the Range feature:

$(a, a, \boldsymbol{a}, \alpha)$

```

\setmathfont{Cambria Math}
$(a, \mita, \mathbf{a}, \mathalpha)$

```

An interesting example of the Range feature:

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

```

\setmathfont[Colour=000000]{Cambria Math}
\setmathfont[Range={\mathop}, Colour=FF0000]{Cambria Math}
\setmathfont[Range={\equal}, Colour=009900]{Cambria Math}
\setmathfont[Range={\mathopen,\mathclose},
              Colour=0000FF]{Cambria Math}
\l[
F(s)=\mscrL\{f(t)\}=\int_0^{\infty} \mathup{e}^{-st}f(t)\,\mathup{d} t
\l]

```

Using a Range including large character sets such as `\mathrel`, `\mathalpha`, *etc.*, is *very slow*! I hope to improve the performance somehow.

`\setmathfont` [#1]: font features

#2 : font name

276 `\newcommand\setmathfont[2][]{}` %

Init

- Erase any conception \LaTeX has of previously defined math symbol fonts; this allows `\DeclareSymbolFont` at any point in the document.

277 `\let\glb@currsizel\relax`

- To start with, assume we're defining the font for every math symbol character.

278 `\let\um@char@range\@empty`

279 `\let\um@char@num@range\@empty`

280 `\@um@initfalse`

- Tell `fontspec` that maths font features are actually allowed.

281 `\@um@fontspec@featuretrue`

- Grab the current size information (is this robust enough? Maybe it should be preceded by `\normalsize`...).

282 `\csname S@\f@size\endcsname`

- Set the name of the math version being defined

283 `\def\um@mversion{normal}%`

284 `\DeclareMathVersion{\um@mversion}%`

Define default font features for the script and scriptscript font.

285 `\def\um@ScriptFeatures{ScriptStyle}%`

286 `\def\um@ScriptScriptFeatures{ScriptScriptStyle}%`

287 `\def\um@ScriptFont{#2}%`

288 `\def\um@ScriptScriptFont{#2}%`

Use fontspec to select a font to use. The macro `\S@<size>` contains the definitions of the sizes used for maths letters, subscripts and subsubscripts in `\tf@size`, `\sf@size`, and `\ssf@size`, respectively.

```

289 \setkeys*[um]{options}{#1}%
290 %\rule{1.5ex}{1.5ex}%
291 \edef\@tempa{\noexpand\zf@fontspec{%
292   Script=Math,SizeFeatures={%
293     {Size=\tf@size-},%
294     {Size=\sf@size-\tf@size,%
295       Font=\um@ScriptFont,%
296       \um@ScriptFeatures},%
297     {Size=-\ssf@size,%
298       Font=\um@ScriptScriptFont,%
299       \um@ScriptScriptFeatures}},%
300   \XKV@rm}{#2}}\@tempa
301 %\rule{1.5ex}{1.5ex}%

```

Probably want to check there that we're not creating multiple symbol fonts with the same NFSS declaration.

Check for the correct number of `\fontdimens`:

```

302 \font\um@font="#2"\relax
303 \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt
304   \@um@ot@math@true
305 \else
306   \um@PackageWarning{%
307     The font '#2' is not a valid OpenType maths font.
308     Some maths features will not be available or behaviour
309     in a substandard manner.}%
310 \fi

311 \ifx\um@char@range\@empty
312   \def\um@symfont{\um@allsym}%
313   \um@PackageInfo{Defining the default maths font as '#2'}%
314   \let\UnicodeMathSymbol\um@mathsymbol@noparse
315 \else
316   \stepcounter{um@fam}%
317   \edef\um@symfont{\um@fam\theum@fam}%
318   \let\UnicodeMathSymbol\um@mathsymbol@parse
319 \fi
320 \DeclareSymbolFont{\um@symfont}
321 {\encodingdefault}{\zf@family}{\mddefault}{\updefault}%

```

And now we input every single maths char. See File III for the source to `unicode-math.tex`.

```

322 \if@um@init
323   \input unicode-math.tex\relax
324   \input unicode-math-add.tex\relax

```

```

325 \else
326   \unless\ifx\um@char@range\@empty
327     \input unicode-math.tex\relax
328     \input unicode-math-add.tex\relax
329   \fi
330 \fi

```

If `\um@char@range` is empty, we are defining maths from scratch. So we empty all of the macros used to switch maths alphabets. Otherwise, the `\um@math..` macros are appended to.

```

331 \ifx\um@char@range\@empty
332   \let\um@mathbb\@empty
333   \let\um@mathbf\@empty
334   \let\um@mathfrak\@empty
335   \let\um@mathup\@empty
336   \let\um@mathit\@empty
337   \let\um@mathscr\@empty
338   \let\um@mathsf\@empty
339   \let\um@mathsfitt\@empty
340   \let\um@mathhtt\@empty
341   \let\um@mathbf\@empty
342   \let\um@mathbfup\@empty
343   \let\um@mathbfitt\@empty
344   \let\um@mathbffrak\@empty
345   \let\um@mathbfscr\@empty
346   \let\um@mathbfsf\@empty
347   \let\um@mathbfsfitt\@empty
348   \let\um@setsinglemathalph\um@mathmap@noparse
349 \else
350   \let\um@setsinglemathalph\um@mathmap@parse
351 \fi

```

6.4 Maths alphabets' character mapping

We want it to be convenient for users to actually type in maths. The ASCII Latin characters should be used for italic maths, and the text Greek characters should be used for upright/italic (depending on preference) Greek, if desired.

Numbers:

```

352 \ifx\um@char@range\@empty
353   \um@def@numbers

```

Normal weight

```

354 \if@um@literal
355   \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}%
356   \um@setmathcode[26]{\um@usv@itLatin}{\um@usv@itLatin}%

```



```

357 \um@setmathcode[26]{\um@usv@itlatin}{\um@usv@itlatin}%
358 \um@setmathcode{\um@usv@ith}{\um@usv@ith}%
359 \um@setmathcode[26]{\um@usv@uplatin}{\um@usv@uplatin}%
360 \um@setmathcode[25]{\um@usv@upGreek}{\um@usv@upGreek}%
361 \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}%
362 \um@setmathcode[25]{\um@usv@itGreek}{\um@usv@itGreek}%
363 \um@setmathcode[25]{\um@usv@upgreek}{\um@usv@upgreek}%

```

: TODO : other literal symbols? or are these redundant?

```

364 \um@setmathcode{\um@usv@Nabla}{\um@usv@Nabla}%
365 \um@setmathcode{\um@usv@itNabla}{\um@usv@itNabla}%
366 \um@setmathcode{\um@usv@partial}{\um@usv@partial}%
367 \um@setmathcode{\um@usv@itpartial}{\um@usv@itpartial}%
368 \else

```

Latin letters:

```

369 \if@um@upLatin\um@def@upLatin\else\um@def@itLatin\fi
370 \if@um@uplatin\um@def@uplatin\else\um@def@itlatin\fi

```

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

```

\setmathfont{Cambria Math}
$0123456789$ \\\
$ABCDEFGHIJKLMNOPQRSTUVWXYZ$ \\\
$abcdefghijklmnopqrstuvwxyz$ \\\

```

Normal weight Greek, italic uppercase and lowercase respectively:

```

371 \if@um@upGreek\um@def@upGreek\else\um@def@itGreek\fi
372 \if@um@upgreek\um@def@upgreek\else\um@def@itgreek\fi

```

Nabla and partial:

```

373 \if@um@upNabla
374 \um@setmathcode{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
375 \else
376 \um@setmathcode{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}%
377 \fi
378 \if@um@uppartial
379 \um@setmathcode{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
380 \else
381 \um@setmathcode{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}%
382 \fi
383 \fi

```

Bold

```

384 \if@um@bfliteral
385 \um@setmathcode[26]{\um@usv@bfLatin}{\um@usv@bfLatin}%
386 \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}%
387 \um@setmathcode[26]{\um@usv@bfitLatin}{\um@usv@bfitLatin}%
388 \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}%

```

```

389 \um@setmathcode[25]{\um@usv@bfGreek}{\um@usv@bfGreek}%
390 \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}%
391 \um@setmathcode[25]{\um@usv@bfitGreek}{\um@usv@bfitGreek}%
392 \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}%
393 \else
394 \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfitLatin}{%
395 \ifum@bfupLatin\um@usv@bfLatin\else\um@usv@bfitLatin\fi}%
396 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{%
397 \ifum@bfuplatin\um@usv@bflatin\else\um@usv@bfitlatin\fi}%
398 \ifum@bfupGreek
399 \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfitGreek}{\um@usv@bfGreek}%
400 \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfvarTheta}%
401 \else
402 \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfitGreek}{\um@usv@bfitGreek}%
403 \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfitvarTheta}%
404 \fi
405 \ifum@bfupgreek
406 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfgreek}%
407 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfvarepsilon}%
408 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}%
409 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}%
410 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}%
411 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}%
412 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}%
413 \else
414 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}%
415 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}%
416 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}%
417 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}%
418 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}%
419 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}%
420 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}%
421 \fi

```

Bold nabla and partial symbols:

```

422 \ifum@upNabla
423 \um@setmathcode{\um@usv@bfNabla}{\um@usv@bfNabla}%
424 \um@setmathcode{\um@usv@bfitNabla}{\um@usv@bfNabla}%
425 \um@setmathcode{\um@usv@bfsfNabla}{\um@usv@bfsfNabla}%
426 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfNabla}%
427 \else
428 \um@setmathcode{\um@usv@bfNabla}{\um@usv@bfitNabla}%
429 \um@setmathcode{\um@usv@bfitNabla}{\um@usv@bfitNabla}%
430 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}%
431 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}%
432 \fi
433 \ifum@uppartial

```

```

434 \um@setmathcode{\um@usv@bfpartial}{}{\um@usv@bfpartial}%
435 \um@setmathcode{\um@usv@bfitpartial}{}{\um@usv@bfitpartial}%
436 \um@setmathcode{\um@usv@bfsfpartial}{}{\um@usv@bfsfpartial}%
437 \um@setmathcode{\um@usv@bfsfitpartial}{}{\um@usv@bfsfitpartial}%
438 \else
439 \um@setmathcode{\um@usv@bfpartial}{}{\um@usv@bfitpartial}%
440 \um@setmathcode{\um@usv@bfitpartial}{}{\um@usv@bfitpartial}%
441 \um@setmathcode{\um@usv@bfsfpartial}{}{\um@usv@bfsfitpartial}%
442 \um@setmathcode{\um@usv@bfsfitpartial}{}{\um@usv@bfsfitpartial}%
443 \fi
444 \fi
445 \else
: TODO : implement behaviour when char@range is NOT empty
446 \fi

```

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ
αβγδεζηθικλμνξοπρστυφχψ ω εθκφρω

```
\setmathfont{Cambria Math}
$AB\Gamma EZH\Theta IK\Lambda MN\Xi O\P\S T\U\Phi\chi\Omega$\quad$\$ \quad
$\pi$\quad$\pi$\quad$\pi$\quad$\pi$\quad$\pi$\quad$\pi$\quad$\pi$\quad$\pi$\quad$\pi$ \quad
```

Set up the maths alphabets:

```
447 \um@setup@alphabets
```

End of the \setmathfont macro.

448 }

\um@mathsymbol@noparse

```
449 \newcommand\um@mathsymbol@noparse[4]{%
```

```
450 \um@mathsymbol{#2}{#3}{\um@symfont}{#1}}
```

`\um@mathsymbol@parse` If the Range font feature has been used, then only a subset of the unicode glyphs are to be defined. See section §7.3 for the code that enables this.

```
451 \newcommand\um@mathsymbol@parse[4]{%
```

```
452 \um@parse@term{#1}{#2}{#3}{%
```

```
453 %\um@PackageInfo{Defining \string#2 as mathchar #1}%
```

454 \um@mathsymbol{#2}{#3}{\um@symfont}{#1}}

`\um@mk@alph` Wrapper to define maths alphabets.

```
455 \newcommand\um@mk@math[1]{%
```

```
456 \expandafter\def\cscname math#1\endcscname##1{%
```

```
457 \begingroup
```

458 \csname um@math#1\endcsname

459 ##1

```
460 \endgroup}}
```

Maths alphabets' base definition. See section §6.4 for the internal definitions.

461 \um@mk@math{up}

```

462 \um@mk@math{it}
463 \um@mk@math{scr}
464 \um@mk@math{bb}
465 \um@mk@math{frak}
466 \um@mk@math{sf}
467 \um@mk@math{sfit}
468 \um@mk@math{tt}

```

And bold maths alphabets. See section §7.4 for the internal definitions.

```

469 \um@mk@math{bf}
470 \um@mk@math{bfup}
471 \um@mk@math{bfif}
472 \um@mk@math{bfscr}
473 \um@mk@math{bffrak}
474 \um@mk@math{bfsf}
475 \um@mk@math{bfsfit}

```

`\mathcal`

```

476 \let\mathcal\mathscr

```

`\um@mathmap@noparse` #1 : Maths alphabet, *e.g.*, `\mathbb`
 #2 : Input slot(s), *e.g.*, the slot for ‘A’ (comma separated)
 #3 : Output slot, *e.g.*, the slot for ‘A’
 Adds `\SetMathCode` declarations to the specified maths alphabet’s definition (*e.g.*, `\um@mathscr`). Uses `\um@addto@mathmap` (below) to expand the name of the current symbol font.

```

477 \newcommand\um@mathmap@noparse[3]{%
478   \@for\@ii:=#2\do{%
479     \expandafter\expandafter
480     \expandafter\um@addto@mathmap
481     \expandafter\expandafter
482     \expandafter{%
483     \expandafter\um@symfont
484     \expandafter}%
485     \expandafter{\@ii}{#1}{#3}%
486   }}%

```

`\um@mathmap@parse` #1 : Maths alphabet, *e.g.*, `\mathbb`
 #2 : Input slot(s), *e.g.*, the slot for ‘A’ (comma separated)
 #3 : Output slot, *e.g.*, the slot for ‘A’
 When `\um@parse@term` is executed, it populates the `\um@char@num@range` macro with slot numbers corresponding to the specified range. This range is used to conditionally add `\SetMathCode` declarations to the maths alphabet definition (*e.g.*, `\um@mathscr`).

```

487 \newcommand\um@mathmap@parse[3]{%
488   \@for\@ii:=\um@char@num@range\do{%

```

```

489 \ifnum\@ii=#3\relax
490 \@for\@jj:=#2\do{%
491 \expandafter\expandafter
492 \expandafter\um@addto@mathmap
493 \expandafter\expandafter
494 \expandafter{%
495 \expandafter\um@symfont
496 \expandafter}%
497 \expandafter{\@jj}{#1}{#3}}%
498 \fi}%

```

`\um@addto@mathmap` #1 : Math symbol font, always/usually the expansion of `\um@symfont`

#2 : Input slot, *e.g.*, the slot for ‘A’

#3 : Maths alphabet, *e.g.*, `\mathbb`

#4 : Output slot, *e.g.*, the slot for ‘A’

This macro is used so that `\um@symfont` can be expanded before entering the `\g@addto@macro` command.

```

499 \newcommand\um@addto@mathmap[4]{%
500 \expandafter\g@addto@macro
501 \csname um@\expandafter\@gobble\string#3\endcsname{%
502 \SetMathCode{#2}{\mathalpha}{#1}{#4}}

```

6.5 (Big) operators











Turns out that \XeTeX is clever enough to deal with big operators for us automatically with `\XeTeXmathchardef`. Amazing!







However, the limits aren’t set automatically; that is, we want to define, a la Plain \TeX *etc.*, `\def\int{\intop\nolimits}`, so there needs to be a transformation from `\int` to `\intop` during the expansion of `\UnicodeMathSymbol` in the appropriate contexts.


Following is a table of every math operator (`\mathop`) defined in `unicode-maths.tex`, from which a subset need to be flagged for `\nolimits` adjustments. The limits behaviour as specified by `unicode-math` are shown (with grey ‘scripts’).

USV	Ex.	Macro	Description
U+02140	$\sum\limits_0^1$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod\limits_0^1$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod\limits_0^1$	<code>\coprod</code>	COPRODUCT OPERATOR

U+02211	$\sum\limits_0^1$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int\limits_0^1$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint\limits_0^1$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint\limits_0^1$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	$\oint\limits_0^1$	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oiint\limits_0^1$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oiiint\limits_0^1$	<code>\oiiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\int\limits_0^1$	<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232	$\oint\limits_0^1$	<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\oint\limits_0^1$	<code>\ointctrackwise</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0	$\bigwedge\limits_0^1$	<code>\bigwedge</code>	LOGICAL OR OPERATOR
U+022C1	$\bigvee\limits_0^1$	<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2	$\bigcap\limits_0^1$	<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3	$\bigcup\limits_0^1$	<code>\bigcup</code>	UNION OPERATOR
U+027D5	$\left[\right]_0^1$	<code>\left[\right]_0^1</code>	LEFT OUTER JOIN
U+027D6	$\left[\right]_0^1$	<code>\left[\right]_0^1</code>	RIGHT OUTER JOIN
U+027D7	$\left[\right]_0^1$	<code>\left[\right]_0^1</code>	FULL OUTER JOIN
U+027D8	$\uparrow\limits_0^1$	<code>\biguparrow</code>	LARGE UP TACK
U+027D9	$\downarrow\limits_0^1$	<code>\bigdownarrow</code>	LARGE DOWN TACK
U+029F8	$\bigodot\limits_0^1$	<code>\xsol</code>	BIG SOLIDUS
U+029F9	$\bigodot\limits_0^1$	<code>\xbsol</code>	BIG REVERSE SOLIDUS
U+02A00	$\bigodot\limits_0^1$	<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR

U+02A01		\bigoplus	N-ARY CIRCLED PLUS OPERATOR
U+02A02		\bigotimes	N-ARY CIRCLED TIMES OPERATOR
U+02A03		\bigcupdot	N-ARY UNION OPERATOR WITH DOT
U+02A04		\biguplus	N-ARY UNION OPERATOR WITH PLUS
U+02A05		\bigsqcap	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06		\bigsqcup	N-ARY SQUARE UNION OPERATOR
U+02A07		\conjquant	TWO LOGICAL AND OPERATOR
U+02A08		\disjquant	TWO LOGICAL OR OPERATOR
U+02A09		\bigtimes	N-ARY TIMES OPERATOR
U+02A0B		\sumint	SUMMATION WITH INTEGRAL
U+02A0C		\iiint	QUADRUPLE INTEGRAL OPERATOR
U+02A0D		\intbar	FINITE PART INTEGRAL
U+02A0E		\intBar	INTEGRAL WITH DOUBLE STROKE
U+02A0F		\fint	INTEGRAL AVERAGE WITH SLASH
U+02A10		\cirfnint	CIRCULATION FUNCTION
U+02A11		\awint	ANTICLOCKWISE INTEGRATION
U+02A12		\rppointint	PATH AROUND POLE LINE INTEGRATION WITH RECTANGULAR
U+02A13		\scpointint	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A14		\npointint	POLE INTEGRAL AROUND A POINT OPERATOR
U+02A15		\sqint	QUATERNION INTEGRAL OPERATOR
U+02A16			INTEGRAL WITH LEFTWARDS ARROW
U+02A17		\intlarhk	WITH HOOK
U+02A18		\intx	INTEGRAL WITH TIMES SIGN
U+02A19		\intcap	INTEGRAL WITH INTERSECTION
U+02A1A		\intcup	INTEGRAL WITH UNION
U+02A1B		\upint	INTEGRAL WITH OVERBAR
U+02A1C		\lowint	INTEGRAL WITH UNDERBAR
U+02A1D		\Join	JOIN

U+02A1E		<code>\bigtriangleleft</code>	LARGE LEFT TRIANGLE OPERATOR
U+02A1F		<code>\zcmp</code>	Z NOTATION SCHEMA COMPOSITION
U+02A20		<code>\zpipe</code>	Z NOTATION SCHEMA PIPING
U+02A21		<code>\zproject</code>	Z NOTATION SCHEMA PROJECTION
U+02AFC		<code>\biginterleave</code>	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFF		<code>\bigtalloblong</code>	N-ARY WHITE VERTICAL BAR

`\um@nolimits` This macro is a commalist containing those maths operators that require a `\no-limits` suffix. This list is used when processing `unicode-math.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 19). I’ve chosen essentially just the operators that look like integrals; hopefully a better mathematician can help me out here. I’ve a feeling that it’s more useful *not* to include the multiple integrals such as , but that might be a matter of preference.

```

503 \def\um@nolimits{%
504   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oint\@elt\oiint
505   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrclockwise\@elt\sumint
506   \@elt\intbar\@elt\intBar\@elt\oint\@elt\cirfnint\@elt\awint\@elt\rppoint
507   \@elt\scpolint\@elt\ntopolint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
508   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint}

```

`\addnolimits` This macro appends material to the macro containing the list of operators that don’t take limits. See example following for usage. Note at present that this command must have taken effect before `\setmathfont`.

```

509 \newcommand\addnolimits[1]{%
510   \expandafter\def
511   \expandafter\um@nolimits
512   \expandafter{\um@nolimits\@elt#1}}

```

`\removenolimits` Can this macro be given a better name? It removes (globally) an item from the `nolimits` list. See example following for usage.

```

513 \def\removenolimits#1{%
514   \begingroup
515     \def\@elt##1{%
516       \ifx##1#1\else
517         \noexpand\@elt\noexpand##1
518       \fi}
519     \xdef\um@nolimits{\um@nolimits}%
520   \endgroup}

```

$$\iiint_V \iiint_V \iiint_V$$

```
\def\dmath#1{\displaystyle #1$}
\setmathfont{Cambria Math} \dmath{\iint_V}
\removenolimits\iint
\setmathfont{Cambria Math} \dmath{\iint_V}
\addnolimits\iint
\setmathfont{Cambria Math} \dmath{\iint_V}
```

6.6 Radicals

The radical for square root is organised in `\um@set@mathsymbol` on page ?? . I think it's the only radical ever. But what about right-to-left square roots?

`\um@radicals` We organise radicals in the same way as `nolimits-operators`; that is, in a comma-list.

```
521 \def\um@radicals{\sqrt}
```

`\r@@t` #1 : A mathstyle (for `\mathpalette`)

#2 : Leading superscript for the sqrt sign

A re-implementation of L^AT_EX's hard-coded n-root sign using the appropriate `\fontdimens`.

```
522 \def\r@@t#1#2{%
523   \setbox\z@\hbox{$\math #1\sqrt{sign{#2}}$}%
524   \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}%
525   \raise \dimexpr(
526     \um@fontdimen@percent{65}{\um@font}\ht\z@-
527     \um@fontdimen@percent{65}{\um@font}\dp\z@
528   )\relax
529   \copy \rootbox%
530   \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}%
531   \box \z@}
```

$$\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + x}}}}}}$$

```
\setmathfont{Cambria Math}
\[\sqrt{1+\sqrt{1+
\sqrt{1+\sqrt{1+
\sqrt{1+\sqrt{1+
\sqrt{1+x}}}}}}}\]
```

$$\sqrt[2]{1 + \sqrt[3]{1 + x}}$$

```
\setmathfont{Cambria Math}
\[\sqrt[2]{1+\sqrt[3]{1+x}} \]
```

6.7 Delimiters

\left We redefine the primitive to be preceded by \mathopen; this gives much better spacing in cases such as \sin\left.... Courtesy of Frank Mittelbach:
<http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3853&prlatex/3754>

```

532 \let\left@primitive\left
533 \def\left{\mathopen{}\left@primitive}

```

No re-definition is made for `\right` because I don't believe it to be necessary...
:TODO: 'fences', e.g., `\vert`

$\left(\left(\left(\left(x^1\right)^2\right)^3\right)^4\right)^5$	<pre>\setmathfont{Cambria Math} \[\left(\left(\left(\left(\left(x \right)^1\right)\right)^2\right)\right)^3\right)^4\right)^5 \]</pre>
$\left[\left[\left[\left[y^1\right]^2\right]^3\right]^4\right]^5$	<pre>\[\left[\left[\left[\left(\left[y \right]^1\right)\right]^2\right)\right]^3\right]^4\right]^5 \]</pre>
$\left\{\left\{\left\{\left\{z^1\right\}^2\right\}^3\right\}^4\right\}^5$	<pre>\left\{\left\{\left\{\left(\left\{z \right\}^1\right)\right\}\right)^2\right\}\right)^3\right\}^4\right\}^5 \]</pre>

Here are all $\mathsf{mathopen}$ characters:

USV	Ex.	Macro	Description
U+00028	(<code>\lparen</code>	LEFT PARENTHESIS
U+0005B	[<code>\lbrack</code>	LEFT SQUARE BRACKET
U+0007B	{	<code>\lbrace</code>	LEFT CURLY BRACKET
			DOUBLE ANGLE QUOTATION MARK
U+000AB	«	<code>\guillemotleft</code>	(GUILLEMET), LEFT
U+02018	‘	<code>\lq</code>	SINGLE QUOTATION MARK, LEFT
U+0201A	,	<code>\quotingslbase</code>	RISING SINGLE QUOTE, LEFT (LOW)
U+0201E	„	<code>\quotdblbase</code>	RISING DOUBLE QUOTE, LEFT (LOW)
			SINGLE ANGLE QUOTATION MARK
U+02039	‹	<code>\guilsinglleft</code>	(GUILLEMET), LEFT
U+0221A	√	<code>\sqrt</code>	RADICAL
U+0221B	∛	<code>\cuberoot</code>	CUBE ROOT
U+0221C	∜	<code>\fourthroot</code>	FOURTH ROOT
U+02308	⌈	<code>\lceil</code>	LEFT CEILING
U+0230A	⌋	<code>\lfloor</code>	LEFT FLOOR
U+0231C	⤵	<code>\ulcorner</code>	UPPER LEFT CORNER
U+0231E	⤵	<code>\llcorner</code>	LOWER LEFT CORNER
			LIGHT LEFT TORTOISE SHELL BRACKET
U+02772	⌘	<code>\lbrbrak</code>	ORNAMENT
U+027C5	⏞	<code>\lbag</code>	LEFT S-SHAPED BAG DELIMITER

U+027CC	⎵	<code>\longdivision</code>	LONG DIVISION MATHEMATICAL LEFT WHITE SQUARE
U+027E6	⎵	<code>\lBrack</code>	BRACKET
U+027E8	⎵	<code>\langle</code>	MATHEMATICAL LEFT ANGLE BRACKET MATHEMATICAL LEFT DOUBLE ANGLE
U+027EA	⎵	<code>\lAngle</code>	BRACKET MATHEMATICAL LEFT WHITE TORTOISE
U+027EC	⎵	<code>\lbrbrak</code>	SHELL BRACKET
U+02983	⎵	<code>\lBrace</code>	LEFT WHITE CURLY BRACKET
U+02985	⎵	<code>\lParen</code>	LEFT WHITE PARENTHESIS
U+02987	⎵	<code>\llparenthesis</code>	Z NOTATION LEFT IMAGE BRACKET
U+02989	⎵	<code>\llangle</code>	Z NOTATION LEFT BINDING BRACKET
U+0298B	⎵	<code>\lbrackubar</code>	LEFT SQUARE BRACKET WITH UNDERBAR LEFT SQUARE BRACKET WITH TICK IN TOP
U+0298D	⎵	<code>\lbrackultick</code>	CORNER LEFT SQUARE BRACKET WITH TICK IN
U+0298F	⎵	<code>\lbracklltick</code>	BOTTOM CORNER
U+02991	⎵	<code>\langedot</code>	LEFT ANGLE BRACKET WITH DOT
U+02993	⎵	<code>\lparenless</code>	LEFT ARC LESS-THAN BRACKET
U+02997	⎵	<code>\lblkbrbrak</code>	LEFT BLACK TORTOISE SHELL BRACKET
U+029D8	⎵	<code>\lvzigzag</code>	LEFT WIGGLY FENCE
U+029DA	⎵	<code>\lvzigzag</code>	LEFT DOUBLE WIGGLY FENCE
U+029FC	⎵	<code>\lcurvyangle</code>	LEFT POINTING CURVED ANGLE BRACKET
U+03014	⎵	<code>\lbrbrak</code>	LEFT BROKEN BRACKET
U+03018	⎵	<code>\Lbrbrak</code>	LEFT WHITE TORTOISE SHELL BRACKET

And `\mathclose`:

USV	Ex.	Macro	Description
U+00029)	<code>\rparen</code>	RIGHT PARENTHESIS
U+0005D]	<code>\rbrack</code>	RIGHT SQUARE BRACKET
U+0007D	}	<code>\rbrace</code>	RIGHT CURLY BRACKET DOUBLE ANGLE QUOTATION MARK
U+000BB	»	<code>\guillemotright</code>	(GUILLEMET), RIGHT
U+02019	'	<code>\rq</code>	SINGLE QUOTATION MARK, RIGHT
U+0201B	⎵	<code>\quotsinglright</code>	RISING SINGLE QUOTE, RIGHT (HIGH)
U+0201F	⎵	<code>\quotdblright</code>	RISING DOUBLE QUOTE, RIGHT (HIGH) SINGLE ANGLE QUOTATION MARK
U+0203A	›	<code>\guilsinglright</code>	(GUILLEMET), RIGHT
U+02309	⎵	<code>\rceil</code>	RIGHT CEILING
U+0230B	⎵	<code>\rfloor</code>	RIGHT FLOOR
U+0231D	⎵	<code>\urcorner</code>	UPPER RIGHT CORNER
U+0231F	⎵	<code>\lrcorner</code>	LOWER RIGHT CORNER LIGHT RIGHT TORTOISE SHELL BRACKET
U+02773	⎵	<code>\rbrbrak</code>	ORNAMENT

U+027C6	?	\rbag	RIGHT S-SHAPED BAG DELIMITER MATHEMATICAL RIGHT WHITE SQUARE
U+027E7	⌋	\rBrack	BRACKET
U+027E9	⌋	\rangle	MATHEMATICAL RIGHT ANGLE BRACKET MATHEMATICAL RIGHT DOUBLE ANGLE
U+027EB	⌋	\rAngle	BRACKET MATHEMATICAL RIGHT WHITE TORTOISE
U+027ED	?	\Rbrbrak	SHELL BRACKET
U+02984	?	\rBrace	RIGHT WHITE CURLY BRACKET
U+02986	?	\rParen	RIGHT WHITE PARENTHESIS
U+02988	?	\rrparenthesis	Z NOTATION RIGHT IMAGE BRACKET
U+0298A	?	\rrangle	Z NOTATION RIGHT BINDING BRACKET RIGHT SQUARE BRACKET WITH
U+0298C	?	\rbrackubar	UNDERBAR RIGHT SQUARE BRACKET WITH TICK IN
U+0298E	?	\rbracklrtick	BOTTOM CORNER RIGHT SQUARE BRACKET WITH TICK IN
U+02990	?	\rbrackurtick	TOP CORNER
U+02992	?	\rangledot	RIGHT ANGLE BRACKET WITH DOT
U+02994	?	\rpangtr	RIGHT ARC GREATER-THAN BRACKET
U+02998	?	\rblbrbrak	RIGHT BLACK TORTOISE SHELL BRACKET
U+029D9	?	\rvzigzag	RIGHT WIGGLY FENCE
U+029DB	?	\Rvzigzag	RIGHT DOUBLE WIGGLY FENCE RIGHT POINTING CURVED ANGLE
U+029FD	⌋	\rcurvyangle	BRACKET
U+03015	⌋	\rbrbrak	RIGHT BROKEN BRACKET
U+03019	?	\Rbrbrak	RIGHT WHITE TORTOISE SHELL BRACKET

6.8 Maths accents

Maths accents should just work *if they are available in the font*.

USV	Ex.	Macro	Description
U+00300	˘	\grave	GRAVE ACCENT
U+00301	ˆ	\acute	ACUTE ACCENT
U+00302	ˆ	\hat	CIRCUMFLEX ACCENT
U+00303	˜	\tilde	TILDE
U+00304	ˉ	\bar	MACRON
U+00305	ˆ	\overbar	OVERBAR EMBELLISHMENT
U+00306	˘	\breve	BREVE
U+00307	˙	\dot	DOT ABOVE
U+00308	¨	\ddot	DIERESIS
U+00309	ˆ	\ovhook	COMBINING HOOK ABOVE
U+0030A	ˆ	\ocirc	RING

U+0030C	Š	\check	CARON
U+00310	Ṡ	\candra	CANDRABINDU (NON-SPACING)
U+00312	Ṣ	\oturnedcomma	COMBINING TURNED COMMA ABOVE
U+00313	Ṥ	\osmooth	GREEK PSILI (SMOOTH BREATHING) (NON-SPACING)
U+00314	Ṧ	\orough	GREEK DASIA (ROUGH BREATHING) (NON-SPACING)
U+00315	Ṩ	\ocommatopright	COMBINING COMMA ABOVE RIGHT
U+0031A	Ṫ	\droang	LEFT ANGLE ABOVE (NON-SPACING)
U+020D0	᳀	\leftharpoonaccent	COMBINING LEFT HARPOON ABOVE
U+020D1	᳁	\rightharpoonaccent	COMBINING RIGHT HARPOON ABOVE
U+020D2	᳂	\vertoverlay	COMBINING LONG VERTICAL LINE OVERLAY
U+020D6	᳆	\overleftarrow	COMBINING LEFT ARROW ABOVE
U+020D7	᳇	\vec	COMBINING RIGHT ARROW ABOVE
U+020DB	ᳫ	\dddot	COMBINING THREE DOTS ABOVE
U+020DC	ᳬ	\ddddot	COMBINING FOUR DOTS ABOVE
U+020E1	ᳱ	\overleftrightharpoon	COMBINING LEFT RIGHT ARROW ABOVE
U+020E7	᳷	\annuity	COMBINING ANNUITY SYMBOL
U+020E8	᳸	\threeunderdot	COMBINING TRIPLE UNDERDOT
U+020E9	᳹	\widebridgeabove	COMBINING WIDE BRIDGE ABOVE
U+020EC	᳼	\underrightharpoondown	COMBINING RIGHTWARDS HARPOON WITH BARB DOWNWARDS
U+020ED	᳽	\underleftharpoondown	COMBINING LEFTWARDS HARPOON WITH BARB DOWNWARDS
U+020EE	᳾	\underleftarrow	COMBINING LEFT ARROW BELOW
U+020EF	᳿	\underrightarrow	COMBINING RIGHT ARROW BELOW
U+020F0	ᳺ	\asteraccent	COMBINING ASTERISK ABOVE

7 Font features

`\um@zf@feature` Use the same method as `fontspec` for feature definition (*i.e.*, using `xkeyval`) but with a conditional to restrict the scope of these features to `unicode-math` commands.

```

534 \newcommand\um@zf@feature[2]{%
535   \define@key[zf]{options}{#1}[]{}%
536   \ifum@fontspec@feature
537     #2
538   \else
539     \PackageError{fontspec/unicode-math}
540       {The ‘#1’ font feature can only be used for maths fonts}
541       {The feature you tried to use can only be in commands
542        like \protect\setmathfont}%
543   \fi}}

```

7.1 OpenType maths font features

```

544 \um@zf@feature{ScriptStyle}{%
545   \zf@update@ff{+ssty=0}}
546 \um@zf@feature{ScriptScriptStyle}{%
547   \zf@update@ff{+ssty=1}}

```

7.2 Script and scriptscript font options

```

548 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
549 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
550 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
551 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

7.3 Range processing

```

552 \define@choicekey+[um]{options}{Range}[\@tempa\@tempb]{ALL}{%
553   \ifcase\@tempb\relax
554     \@um@inittrue
555   \fi}{% else:
556     \xdef\um@char@range{\zap@space#1 \@empty}}

```

Pretty basic comma separated range processing. Donald Arseneau’s selectp package has a cleverer technique.

`\um@parse@term`

- #1 : unicode character slot
- #2 : control sequence (character macro)
- #3 : control sequence (math type)
- #4 : code to execute

This macro expands to #4 if any of its arguments are contained in the commalist `\um@char@range`. This list can contain either character ranges (for checking with #1) or control sequences. These latter can either be the command name of a specific character, *or* the math type of one (*e.g.*, `\mathbin`).

Character ranges are passed to `\um@parse@range`, which accepts input in the form shown in table 9.

Table 9: Ranges accepted by `\um@parse@range`.

Input	Range
x	$r = x$
$x-$	$r \geq x$
$-y$	$r \leq y$
$x-y$	$x \leq r \leq y$

Start by iterating over the commalist, ignoring empties, and initialising the scratch conditional:

```

557 \newcommand\um@parse@term[4]{%
558   \@for\@ii:=\um@char@range\do{%

```

```

559 \unless\ifx\@ii\@empty
560 \@tempswafalse

```

Match to either the character macro (`\alpha`) or the math type (`\mathbin`):

```

561 \expandafter\um@firstchar\expandafter{\@ii}%
562 \ifx\@tempa\um@backslash
563 \expandafter\ifx\@ii#2\relax
564 \@tempswatrue
565 \else
566 \expandafter\ifx\@ii#3\relax
567 \@tempswatrue
568 \fi
569 \fi

```

Otherwise, we have a number range, which is passed to another macro:

```

570 \else
571 \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
572 \fi

```

If we have a match, execute the code! It also populates the `\um@char@num@range` macro, which is used when defining `\mathbf` (*etc.*) `\mathchar` remappings.

```

573 \if@tempswa
574 \ifx\um@char@num@range\@empty
575 \g@addto@macro\um@char@num@range{#1}%
576 \else
577 \g@addto@macro\um@char@num@range{,#1}%
578 \fi
579 #4%
580 \fi
581 \fi}}
582 \def\um@firstof#1#2\@nil{#1}
583 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
584 \def\um@firstchar#1{\edef\@tempa{\expandafter\um@firstof\string#1\@nil}}

```

'1' or 'a' or 'b' is included '1' or 'b' or 'c' is included '3' or 'a' or 'b' is included '3' or 'a' or 'b' is included

```

\def\um@char@range{\a,2-4,\c}
\um@parse@term{1}{\a}{\b}
{ '1' or '\string'a' or '\string'b' is included}
\um@parse@term{1}{\b}{\c}
{ '1' or '\string'b' or '\string'c' is included}
\um@parse@term{3}{\a}{\b}
{ '3' or '\string'a' or '\string'b' is included}

```

`\um@parse@range` Weird syntax. As shown previously in table 9, this macro can be passed four different input types via `\um@parse@term`.

```

585 \def\um@parse@range#1-#2-#3\@nil#4\@nil{%
586 \def\@tempa{#1}%
587 \def\@tempb{#2}%

```

Range $r = x$
C-list input \@ii=X
Macro input \um@parse@range X-\@marker-\@nil#1\@nil
Arguments $\#1-\#2-\#3 = X-\@marker-\{\}$

```

588 \expandafter\ifx\expandafter\@marker\@tempb\relax
589 \ifnum#4=#1\relax
590 \@tempwattrue
591 \fi
592 \else

```

Range $r \geq x$
C-list input \@ii=X-
Macro input \um@parse@range X--\@marker-\@nil#1\@nil
Arguments $\#1-\#2-\#3 = X-\{\}-\@marker-$

```

593 \ifx\@empty\@tempb
594 \ifnum#4>\numexpr#1-1\relax
595 \@tempwattrue
596 \fi
597 \else

```

Range $r \leq y$
C-list input \@ii=-Y
Macro input \um@parse@range -Y-\@marker-\@nil#1\@nil
Arguments $\#1-\#2-\#3 = \{\}-Y-\@marker-$

```

598 \ifx\@empty\@tempa
599 \ifnum#4<\numexpr#2+1\relax
600 \@tempwattrue
601 \fi

```

Range $x \leq r \leq y$
C-list input \@ii=X-Y
Macro input \um@parse@range X-Y-\@marker-\@nil#1\@nil
Arguments $\#1-\#2-\#3 = X-Y-\@marker-$

```

602 \else
603 \ifnum#4>\numexpr#1-1\relax
604 \ifnum#4<\numexpr#2+1\relax
605 \@tempwattrue
606 \fi\fi\fi\fi\fi}

```

\um@setmathcode #1 : Starting input char(s)
#2 : Number of iterations
#3 : Starting output char
Loops through character ranges setting \mathcode.

```

607 \newcommand\um@setmathcode[3][1]{%
608 \@for\um@inp:=#2\do{%
609 \um@FOR\@tempcnta=[1:#1]\do{%
610 \SetMathCode{\numexpr\um@inp+\@tempcnta-1\relax}

```



```
611 {\mathalpha}{\um@symfont}{\numexpr#3+\@tempcnta-1\relax}}}
```

\um@setmathalph [*(Number of iterations)*] #1 : Maths alphabet
 #2 : Starting input char(s)
 #3 : Starting output char
 Loops through character ranges setting \mathcode.

```
612 \newcommand\um@setmathalph[4][1]{%
613   \@for\um@inp:=#3\do{%
614     \um@FOR\@tempcnta=[1:#1]\do{%
615       \edef\@tempa{%
616         \noexpand\um@setsinglemathalph
617         {\noexpand#2}
618         {\number\numexpr \um@inp+\@tempcnta-1 \relax}
619         {\number\numexpr #4+\@tempcnta-1 \relax}}\@tempa}}}
```

BCDBCD *ABCDEF*

```
{\um@setmathcode[3]{\A,\D}{\B}
$ABCDEF$} $ABCDEF$
```

\um@resolve@greek This macro defines \Alpha...\omega as their corresponding unicode (mathematical italic) character. Remember that the mapping to upright or italic happens with the mathcode definitions, whereas these macros just stand for the literal unicode characters.

```
620 \AtBeginDocument{\um@resolve@greek}%
621 \newcommand\um@resolve@greek{%
622   \def\Alpha{\mitAlpha}%
623   \def\Beta{\mitBeta}%
624   \def\Gamma{\mitGamma}%
625   \def\Delta{\mitDelta}%
626   \def\Epsilon{\mitEpsilon}%
627   \def\Zeta{\mitZeta}%
628   \def\Eta{\mitEta}%
629   \def\Theta{\mitTheta}%
630   \def\Iota{\mitIota}%
631   \def\Kappa{\mitKappa}%
632   \def\Lambda{\mitLambda}%
633   \def\Mu{\mitMu}%
634   \def\Nu{\mitNu}%
635   \def\Xi{\mitXi}%
636   \def\Omicron{\mitOmicron}%
637   \def\Pi{\mitPi}%
638   \def\Rho{\mitRho}%
639   \def\varTheta{\mitvarTheta}%
640   \def\Sigma{\mitSigma}%
```

```

641 \def\Tau{\mitTau}%
642 \def\Upsilon{\mitUpsilon}%
643 \def\Phi{\mitPhi}%
644 \def\Chi{\mitChi}%
645 \def\Psi{\mitPsi}%
646 \def\Omega{\mitOmega}%

```

Lowercase:

```

647 \def\alpha{\mitalpha}%
648 \def\beta{\mitbeta}%
649 \def\gamma{\mitgamma}%
650 \def\delta{\mitdelta}%
651 \def\varepsilon{\mitvarepsilon}%
652 \def\zeta{\mitzeta}%
653 \def\eta{\miteta}%
654 \def\theta{\mittheta}%
655 \def\iota{\mitiota}%
656 \def\kappa{\mitkappa}%
657 \def\lambda{\mitlambda}%
658 \def\mu{\mitmu}%
659 \def\nu{\mitnu}%
660 \def\xi{\mitxi}%
661 \def\omicron{\mitomicron}%
662 \def\pi{\mitpi}%
663 \def\rho{\mitrho}%
664 \def\varsigma{\mitvarsigma}%
665 \def\sigma{\mitsigma}%
666 \def\tau{\mittau}%
667 \def\upsilon{\mitupsilon}%
668 \def\phi{\mitphi}%
669 \def\chi{\mitchi}%
670 \def\psi{\mitpsi}%
671 \def\omega{\mitomega}%
672 \def\varepsilon{\mitvarepsilon}%
673 \def\vartheta{\mitvartheta}%
674 \def\varkappa{\mitvarkappa}%
675 \def\varphi{\mitvarphi}%
676 \def\varrho{\mitvarrho}%
677 \def\varrho{\mitvarsigma}%
678 \def\varpi{\mitvarpi}}

```

\um@def@numbers

```

679 \newcommand\um@def@numbers{%
680 \um@setmathcode[10]{\um@usv@num}{\um@usv@num}}

```

\um@def@upLatin

```

681 \newcommand\um@def@upLatin{%

```

```

682 \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}}

\um@def@itLatin
683 \newcommand\um@def@itLatin{%
684 \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}}

\um@def@itlatin Don't overlook 'h', which maps to U+210E: PLANCK CONSTANT instead of the ex-
pected U+1D455: MATHEMATICAL ITALIC SMALL H.
685 \newcommand\um@def@itlatin{%
686 \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}%
687 \um@setmathcode{\`h,\um@usv@ith}{\um@usv@ith}%
688 }

\um@def@uplatin
689 \newcommand\um@def@uplatin{%
690 \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}%
691 \um@setmathcode{\um@usv@ith}{\`h}%
692 }

\um@def@upGreek
693 \newcommand\um@def@upGreek{%
694 \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}%
695 \um@setmathcode{\um@usv@varTheta,"1D6F3}{\um@usv@varTheta}%
696 }

\um@def@itGreek
697 \newcommand\um@def@itGreek{%
698 \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}%
699 \um@setmathcode{\um@usv@varTheta}{\um@usv@itvarTheta}%
700 }

\um@def@upgreek
701 \newcommand\um@def@upgreek{%
702 \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}%
703 \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
704 \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
705 \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
706 \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
707 \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
708 \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
709 }

\um@def@itgreek
710 \newcommand\um@def@itgreek{%
711 \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}%
712 \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}%

```



```

23 \um@setmathalph{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
24 \um@setmathalph{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
25 \um@setmathalph{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}%
26 \um@setmathalph{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
27 \um@setmathalph{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
28 \um@setmathalph{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
29 \um@setmathalph{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
30 \um@setmathalph{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
31 \um@setmathalph{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
32 }

```

`\mathrm` Can't call it `\mathrm` any more because it contains Greek as well! But we define `\mathrm` as an alias for backwards compatibility.

```

33 \def\mathrm{\mathup}

```

7.3.2 Italic: `\mathit`

<i>ABCDEFGHIJKLMNOPQRSTUVWXYZ</i>	<code>\$\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
<i>abcdefghijklmnopqrstuvwxyz</i>	<code>\$\mathit{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>
<i>ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ</i>	<code>\$\mathit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$\quad\$\mathit{Θ}\$ \\\</code>
<i>αβγδεζηθικλμνξοπρστυφχψω εθκφρϖ</i>	<code>\$\mathit{αβγδεζηθικλμνξοπρστυφχψω}\$\quad\$\mathit{εθκφρϖ}\$ \\\</code>

Roman:

```

34 \def\um@setup@mathit{%
35 \um@setmathalph[26]{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}%
36 \um@setmathalph[26]{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}%
37 \um@setmathalph{\mathit}{\`h,\um@usv@ith}{\um@usv@ith}%

```

Greek:

```

38 \um@setmathalph[25]{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}%
39 \um@setmathalph[25]{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}%
40 \um@setmathalph{\mathit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}%
41 \um@setmathalph{\mathit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}%
42 \um@setmathalph{\mathit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarTheta}%
43 \um@setmathalph{\mathit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}%
44 \um@setmathalph{\mathit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}%
45 \um@setmathalph{\mathit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}%
46 \um@setmathalph{\mathit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}%
47 \um@setmathalph{\mathit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}%
48 \um@setmathalph{\mathit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}%
49 }

```

7.3.3 Blackboard or double-struck: `\mathbb`

	<code>0123456789</code>	<code>\$\mathbb{0123456789}\$ \\\</code>
<code>ABCDEFGHIJKLMNOPQRSTUVWXYZ</code>		<code>\$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
<code>abcdefghijklmnopqrstuvwxyz</code>		<code>\$\mathbb{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>

Numbers:

```
50 \def\um@setup@mathbb{%
51   \um@setmathalph[10]{\mathbb}{\um@usv@num}{\um@usv@bbnum}%
```

Roman uppercase:

```
52   \um@setmathalph[26]{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bbLatin}%
53   \um@setmathalph{\mathbb}{`\C,"1D60A}{`"2102}%
54   \um@setmathalph{\mathbb}{`\H,"1D60F}{`"210D}%
55   \um@setmathalph{\mathbb}{`\N,"1D60F}{`"2115}%
56   \um@setmathalph{\mathbb}{`\P,"1D617}{`"2119}%
57   \um@setmathalph{\mathbb}{`\Q,"1D618}{`"211A}%
58   \um@setmathalph{\mathbb}{`\R,"1D619}{`"211D}%
59   \um@setmathalph{\mathbb}{`\Z,"1D621}{`"2124}%
```

Roman lowercase:

```
60   \um@setmathalph[26]{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bblatin}%
61 }
```

7.3.4 Script or caligraphic: `\mathscr` and `\mathcal`

`\mathcal` and `\mathscr` are aliases.

<code>ABCDEFGHIJKLMNOPQRSTUVWXYZ</code>	<code>\$\mathscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
<code>abcdefghijklmnopqrstuvwxyz</code>	<code>\$\mathscr{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>

```
62 \def\um@setup@mathscr{%
63   \um@setmathalph[26]{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}%
64   \um@setmathalph{\mathscr}{`\B,"1D435}{`"212C}%
65   \um@setmathalph{\mathscr}{`\E,"1D438}{`"2130}%
66   \um@setmathalph{\mathscr}{`\F,"1D439}{`"2131}%
67   \um@setmathalph{\mathscr}{`\H,"1D43B}{`"210B}%
68   \um@setmathalph{\mathscr}{`\I,"1D43C}{`"2110}%
69   \um@setmathalph{\mathscr}{`\L,"1D43F}{`"2112}%
70   \um@setmathalph{\mathscr}{`\M,"1D440}{`"2133}%
71   \um@setmathalph{\mathscr}{`\R,"1D445}{`"211B}%
72   \um@setmathalph[26]{\mathscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@scrlatin}%
73   \um@setmathalph{\mathscr}{`\e,"1D452}{`"212F}%
74   \um@setmathalph{\mathscr}{`\g,"1D454}{`"210A}%
75   \um@setmathalph{\mathscr}{`\o,"1D45C}{`"2134}%
76 }
```

7.3.5 Fraktur or fraktur or blackletter: \mathfrak

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$	<code>\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
$\mathfrak{abcdefghijklmnopqrstuvwxyz}$	<code>\mathfrak{abcdefghijklmnopqrstuvwxyz}\$ \</code>

Letters, with exceptions {C, G, S, R, Z}:

```

77 \def\um@setup@mathfrak{%
78   \um@setmathalph[26]{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}%
79   \um@setmathalph{\mathfrak}{`C,"1D436}{`"212D}%
80   \um@setmathalph{\mathfrak}{`H,"1D43B}{`"210C}%
81   \um@setmathalph{\mathfrak}{`I,"1D43C}{`"2111}%
82   \um@setmathalph{\mathfrak}{`R,"1D445}{`"211C}%
83   \um@setmathalph{\mathfrak}{`Z,"1D44D}{`"2128}%
84   \um@setmathalph[26]{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}%
85 }

```

7.3.6 Sans serif: \mathsf

0123456789	<code>\mathsf{0123456789}\$ \</code>
ABCDEFGHIJKLMNOPQRSTUVWXYZ	<code>\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
abcdefghijklmnopqrstuvwxyz	<code>\mathsf{abcdefghijklmnopqrstuvwxyz}\$ \</code>

```

86 \def\um@setup@mathsf{%
87   \um@setmathalph[10]{\mathsf}{\um@usv@num}{\um@usv@sfnun}%
88   \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sflatin}%
89   \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sflatin}%
90 }

```

7.3.7 Sans serif italic: \mathsf

0123456789	<code>\mathsf{0123456789}\$ \</code>
<i>ABCDEFGHIJKLMNOPQRSTUVWXYZ</i>	<code>\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
<i>abcdefghijklmnopqrstuvwxyz</i>	<code>\mathsf{abcdefghijklmnopqrstuvwxyz}\$ \</code>

```

91 \def\um@setup@mathsf{%
92   \um@setmathalph[10]{\mathsf}{\um@usv@num}{\um@usv@sfnun}%
93   \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLatin}%
94   \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLatin}%
95 }

```

7.3.8 Typewriter or monospaced: `\mathtt`

0123456789	
ABCDEFGHIJKLMNOPQRSTUVWXYZ	
abcdefghijklmnopqrstuvwxyz	

	<code>\$\mathtt{0123456789}\$ \\\</code>
	<code>\$\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
	<code>\$\mathtt{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>

```

96 \def\um@setup@mathtt{%
97   \um@setmathalph[10]{\mathtt}{\um@usv@num}{\um@usv@ttnum}%
98   \um@setmathalph[26]{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}%
99   \um@setmathalph[26]{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}%
100 }
```

7.4 Bold alphabets' character mappings

7.4.1 Bold: `\mathbf`

0123456789	
ABCDEFGHIJKLMNOPQRSTUVWXYZ	
abcdefghijklmnopqrstuvwxyz	
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ	
θ	
αβγδεζηθικλμνξοπρστυφχψω	
εθκφρϖ	

	<code>\$\mathbf{0123456789}\$ \\\</code>
	<code>\$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
	<code>\$\mathbf{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>
	<code>\$\mathbf{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$ \quad \$\mathbf{θ}\$ \\\</code>
	<code>\$\mathbf{αβγδεζηθικλμνξοπρστυφχψω}\$ \quad \$\mathbf{εθκφρϖ}\$ \\\</code>

```

101 \def\um@setup@mathbf{%
102   \um@setmathalph[10]{\mathbf}{\um@usv@num}{\um@usv@bfnum}%
103   \um@setmathalph{\mathbf}{\um@usv@Digamma}{1D7CA}%
104   \um@setmathalph{\mathbf}{\um@usv@digamma}{1D7CB}%
105   \if@um@bfliteral
106     \um@setmathalph[26]{\mathbf}{\um@usv@upLatin}{\um@usv@bfLatin}%
107     \um@setmathalph[26]{\mathbf}{\um@usv@itLatin}{\um@usv@bfItLatin}%
108     \um@setmathalph[26]{\mathbf}{\um@usv@uplatin}{\um@usv@bflatin}%
109     \um@setmathalph[26]{\mathbf}{\um@usv@itlatin}{\um@usv@bfItlatin}%
110     \um@setmathalph[25]{\mathbf}{\um@usv@upGreek}{\um@usv@bfGreek}%
111     \um@setmathalph[25]{\mathbf}{\um@usv@itGreek}{\um@usv@bfItGreek}%
112     \um@setmathalph[25]{\mathbf}{\um@usv@upgreek}{\um@usv@bfgreek}%
113     \um@setmathalph[25]{\mathbf}{\um@usv@itgreek}{\um@usv@bfItgreek}%
114     \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfith}%
115     \um@setmathalph{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}%
116     \um@setmathalph{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}%
117     \um@setmathalph{\mathbf}{\um@usv@Digamma}{\um@usv@bfDigamma}%
118     \um@setmathalph{\mathbf}{\um@usv@partial}{\um@usv@bfpartial}%
119     \um@setmathalph{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}%

```



```

120 \um@setmathalph{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}%
121 \um@setmathalph{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}%
122 \um@setmathalph{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}%
123 \um@setmathalph{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}%
124 \um@setmathalph{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}%
125 \um@setmathalph{\mathbf}{\um@usv@digamma}{\um@usv@bfdigamma}%
126 \um@setmathalph{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}%
127 \um@setmathalph{\mathbf}{\um@usv@itNabla}{\um@usv@bfitNabla}%
128 \um@setmathalph{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}%
129 \um@setmathalph{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}%
130 \um@setmathalph{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}%
131 \um@setmathalph{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}%
132 \um@setmathalph{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}%
133 \um@setmathalph{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}%
134 \um@setmathalph{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}%
135 \else
136 \ifum@bfupLatin
137 \um@setmathalph[26]{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bLatin}%
138 \else
139 \um@setmathalph[26]{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}%
140 \fi
141 \ifum@bfuplatin
142 \um@setmathalph[26]{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@blatin}%
143 \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfuph}%
144 \else
145 \um@setmathalph[26]{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}%
146 \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfith}%
147 \fi
148 \ifum@bfupGreek
149 \um@setmathalph[25]{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfGreek}%
150 \um@setmathalph{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvarTheta}%
151 \else
152 \um@setmathalph[25]{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}%
153 \um@setmathalph{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitvarTheta}%
154 \fi
155 \ifum@bfupgreek
156 \um@setmathalph[25]{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfgreek}%
157 \um@setmathalph{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfvarepsilon}%
158 \um@setmathalph{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfvartheta}%
159 \um@setmathalph{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfvarkappa}%
160 \um@setmathalph{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}%
161 \um@setmathalph{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}%
162 \um@setmathalph{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}%
163 \else
164 \um@setmathalph[25]{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}%
165 \um@setmathalph{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}%

```

```

166 \um@setmathalph{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitvartheta}%
167 \um@setmathalph{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitvarkappa}%
168 \um@setmathalph{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi}%
169 \um@setmathalph{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}%
170 \um@setmathalph{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}%
171 \fi
172 \fi
173 }
174 % \um@setmathalph{\mathbf}{\um@usv@Nabla}{\um@usv@Nabla}{\um@usv@bfitNabla}%
175 % \um@setmathalph{\mathbf}{\um@usv@partial}{\um@usv@partial}{\um@usv@bfitpartial}%
176 % \um@setmathalph{\mathbf}{\um@usv@itNabla}{\um@usv@itNabla}{\um@usv@bfititNabla}%
177 % \um@setmathalph{\mathbf}{\um@usv@itpartial}{\um@usv@itpartial}{\um@usv@bfititpartial}%

```

7.4.2 Bold Italic: \mathbfit

<p style="text-align: center;"> 0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ αβγδεζηθικλμνξοπρστυφχψω εθκφρϖ </p>	<pre> \$ \mathbfit{0123456789}\$ \ \$ \mathbfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \ \$ \mathbfit{abcdefghijklmnopqrstuvwxyz}\$ \ \$ \mathbfit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$ \quad \$ \mathbfit{ }\$ \ \$ \mathbfit{00000000000000000000000000000000}\$ \quad \$ \mathbfit{000000}\$ \ </pre>
---	--

```

178 \def\um@setup@mathbfit{%
179 \um@setmathalph[10]{\mathbfit}{\um@usv@num}{\um@usv@bfnun}%
180 \um@setmathalph[26]{\mathbfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}%
181 \um@setmathalph[26]{\mathbfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}%
182 \um@setmathalph[25]{\mathbfit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}%
183 \um@setmathalph[25]{\mathbfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}%
184 \um@setmathalph[26]{\mathbfit}{\um@usv@bflatin}{\um@usv@bfitlatin}%
185 \um@setmathalph[26]{\mathbfit}{\um@usv@bflatin}{\um@usv@bfitlatin}%
186 \um@setmathalph[25]{\mathbfit}{\um@usv@bfgreek}{\um@usv@bfitgreek}%
187 \um@setmathalph[25]{\mathbfit}{\um@usv@bfgreek}{\um@usv@bfitgreek}%
188 \um@setmathalph{\mathbfit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitvarTheta}%
189 \um@setmathalph{\mathbfit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfitNabla}%
190 \um@setmathalph{\mathbfit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpartial}%
191 \um@setmathalph{\mathbfit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}%
192 \um@setmathalph{\mathbfit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitvartheta}%
193 \um@setmathalph{\mathbfit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitvarkappa}%
194 \um@setmathalph{\mathbfit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi}%
195 \um@setmathalph{\mathbfit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}%
196 \um@setmathalph{\mathbfit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}%
197 }

```

7.4.3 Bold Italic: `\mathbfup`

0123456789
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ
 αβγδεζηθικλμνξοπρστυφχψω εδκφορσ

$$\begin{aligned} & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{0123456789}\}\text{\$} \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\}\text{\$} \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{abcdefghijklmnopqrstuvwxyz}\}\text{\$} \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{ABGA\Delta EZH\Omega K\Lambda MN\epsilon O\overline{P}T\Sigma\Phi\chi\Psi}\}\text{\$}\quad \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{\emptyset}\}\text{\$} \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{000000000000000000\pi00000000}\}\text{\$}\quad \\ & \text{\$}\backslash\mathrm{mathbf{fup}}\{\mathtt{000000}\}\text{\$} \end{aligned}$$

```

198 \def\um@setup@mathbfup{%
199   \um@setmathalph[10]{\mathbfup}{\um@usv@num}{\um@usv@bnum}%
200   \um@setmathalph[26]{\mathbfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bLatin}%
201   \um@setmathalph[26]{\mathbfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bLatin}%
202   \um@setmathalph[25]{\mathbfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bGreek}%
203   \um@setmathalph[25]{\mathbfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bGreek}%
204   \um@setmathalph[26]{\mathbfup}{\um@usv@bLatin}{\um@usv@bLatin}%
205   \um@setmathalph[26]{\mathbfup}{\um@usv@bLatin}{\um@usv@bLatin}%
206   \um@setmathalph[25]{\mathbfup}{\um@usv@bGreek}{\um@usv@bGreek}%
207   \um@setmathalph[25]{\mathbfup}{\um@usv@bGreek}{\um@usv@bGreek}%
208   \um@setmathalph{\mathbfup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bvarTheta}%
209   \um@setmathalph{\mathbfup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bNabla}%
210   \um@setmathalph{\mathbfup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bpartial}%
211   \um@setmathalph{\mathbfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@b varepsilon}%
212   \um@setmathalph{\mathbfup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bvartheta}%
213   \um@setmathalph{\mathbfup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bvarkappa}%
214   \um@setmathalph{\mathbfup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bvarphi}%
215   \um@setmathalph{\mathbfup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bvarrho}%
216   \um@setmathalph{\mathbfup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bvarpi}%
217 }

```

7.4.4 Bold fractur or fraktur or blackletter: `\mathbf{frak}`

ឧបទេស្តន្ទក្រណាត់ព័ន្ធហិរញ្ញវត្ថុ
abcdefghijklmnopqrstuvwxyz

```
\setmathfont{Cambria Math}

$$\frac{ABCDEFGHIJKLMNOPQRSTUVWXYZ}{abcdefghijklmnopqrstuvwxyz}$$


$$\frac{ABCDEFGHIJKLMNOPQRSTUVWXYZ}{abcdefghijklmnopqrstuvwxyz}$$

```

```

218 \def\um@setup@mathbfrac{%
219   \um@setmathalph{10}{\mathbfrac}{\um@usv@num}{\um@usv@bnum}%
220   \um@setmathalph{26}{\mathbfrac}{\um@usv@upLatin, \um@usv@itLatin, \um@usv@frakLatin}{\um@usv@
221   \um@setmathalph{26}{\mathbfrac}{\um@usv@uplatin, \um@usv@itlatin, \um@usv@fraklatin}{\um@usv@
222 }

```

7.4.5 Bold script or calligraphic: `\mathbf{scr}`

*ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz*

```
\setmathfont{Cambria Math}
$\mathbf{fscr}{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \
$\mathbf{fscr}{abcdefghijklmnopqrstuvwxyz}$ \
```

```

223 \def\um@setup@mathbfscsr{%
224   \um@setmathalph[10]{\mathbfscsr}{\um@usv@num}{\um@usv@bfnum}%
225   \um@setmathalph[26]{\mathbfscsr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfscsrLatin}%
226   \um@setmathalph[26]{\mathbfscsr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfscsrLatin}%
227 }

```

7.4.6 Bold sans serif: \mathbfsf

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

[illegible][illegible]

Numbers: (always upright)

```

228 \def\um@setup@mathbfsf{%
229   \um@setmathalph[10]{\mathbfsf}{\um@usv@num}{\um@usv@bfnum}%
230   \um@setmathalph[26]{\mathbfsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfLatin}%
231   \um@setmathalph[26]{\mathbfsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfLatin}%
232   \um@setmathalph[25]{\mathbfsf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfGreek}%
233   \um@setmathalph[25]{\mathbfsf}{\um@usv@uparek,\um@usv@itarek}{\um@usv@bfsfGreek}%

```

Theta symbol:

234 \um@setmathalph{\mathbfsf}{\um@usv@varTheta}{"1D767}%

Nabla:

235 \um@setmathalph{\mathbfsf}{\um@usv@Nabla}{\{"1D76F}%

Partial, epsilon symbol, theta symbol, kappa symbol, phi symbol, rho symbol, pi symbol:

```

236 \um@setmathalph{\mathbfsf}{\um@usv@partial}{\"1D789}%
237 \um@setmathalph{\mathbfsf}{\um@usv@varepsilon}{\"1D78A}%
238 \um@setmathalph{\mathbfsf}{\um@usv@vartheta}{\"1D78B}%
239 \um@setmathalph{\mathbfsf}{\um@usv@varkappa}{\"1D78C}%
240 \um@setmathalph{\mathbfsf}{\um@usv@varphi}{\"1D78D}%
241 \um@setmathalph{\mathbfsf}{\um@usv@varrho}{\"1D78E}%
242 \um@setmathalph{\mathbfsf}{\um@usv@varpi}{\"1D78F}%
243 }

```

[illegible]

Other symbols:

Here we define every unicode math codepoint an equivalent macro name. The two are equivalent, in a `\let\xyz=^^^1234` kind of way.

To do this, turn ^ into a regular ‘other’ character and define the macro to perform the lowercasing and \let. \scantokens changes the carets back into their original meaning after the group has ended and ^’s catcode returns to normal.

53

```
268 \let\unicodemathgobble\@gobble
```

Now give `\UnicodeMathSymbol` a definition in terms of `\um@scancharlet` and we're good to go.

```
269 \begingroup
270 \def\UnicodeMathSymbol#1#2#3#4{%
271   \um@scancharlet#2=#1\@nil}
272 \input unicode-math.tex\relax
273 \input unicode-math-add.tex\relax
274 \endgroup
```

Undo the damage made to `\sqrt`:

```
275 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}
```

We need to change L^AT_EX's idea of the font used to typeset things like `\sin` and `\cos`:

```
276 \def\operator@font{\um@mathup}
277 \AtBeginDocument{\setmathfont[Range=ALL]{Cambria Math}}
```

File III

stix table data extraction

The source for the T_EX names for the very large number of mathematical glyphs are provided via Barbara Beeton's table file for the STIX project (ams.org/STIX). A version is located at <http://www.ams.org/STIX/bnb/stix-tbl.asc> but check <http://www.ams.org/STIX/> for more up-to-date info.

A single file is produced containing all (more than 3298) symbols. Future optimisations might include generating various (possibly overlapping) subsets so not all definitions must be read just to redefine a small range of symbols. Performance for now seems to be acceptable without such measures.

```
1 #!/bin/sh
2
3 cat stix-tbl.txt |
4 awk '
```

If the USV isn't repeated (TODO: check this is valid!) and the entry isn't one of the weird ones in the big block at the end of the STIX table (TODO: check that out!)

```
5 {if (usv != substr($0,2,5) && substr($0,2,1) != " ")
6   {usv = substr($0,2,5);
7     texname = substr($0,84,25);
8     class = substr($0,57,1);
9     description = tolower(substr($0,233,350));
```

If the USV has a macro name, which isn't `\text...`, and isn't a single character macro (e.g., `\#`, `\S`, ...), and has a class, and it isn't reserved (*i.e.*, doubled up with a previously assigned glyph):

```

10     if (texname ~ /[\\]/ &&
11         substr(texname,0,5) != "\\text" &&
12         substr(texname,0,4) != "\\ipa" &&
13         substr(texname,0,5) != "\\tone" &&
14         substr(texname,3,1) != " " &&
15         class != " " &&
16         description !~ /<reserved>/ )

```

Print the actual entry corresponding to the unicode character:

```

17     print "\\UnicodeMathSymbol{" " \
18         usv "{" " \
19         texname "{" " \
20         class "{" " \
21         description "%";
22     }}" - |

```

Now replace the `stix` class abbreviations with their \TeX macro names.

```

23 sed -e ' s/{N}/{\\mathord}/ ' \

```

A ‘fence’ defined by the `stix` table is something like `\vert`; in \XeTeX this is just a `\mathord` that will grow with the magic of `\XeTeXmathchardef`.

```

24     -e ' s/{F}/{\\mathord}/ ' \
25     -e ' s/{A}/{\\mathalpha}/ ' \
26     -e ' s/{D}/{\\mathaccent}/ ' \
27     -e ' s/{P}/{\\mathpunct}/ ' \
28     -e ' s/{B}/{\\mathbin}/ ' \
29     -e ' s/{R}/{\\mathrel}/ ' \
30     -e ' s/{L}/{\\mathop}/ ' \
31     -e ' s/{O}/{\\mathopen}/ ' \
32     -e ' s/{C}/{\\mathclose}/ ' \

```

Fixing up a couple of things in the `STIX` table.

```

33     -e ' s/^/\\string^/ ' > unicode-math.tex

```

A Documenting maths support in the NFSS

A.1 Overview

In the following, $\langle NFSS\ decl. \rangle$ stands for something like $\{T1\}\{lmr\}\{m\}\{n\}$.

Maths symbol fonts Fonts for symbols: α , \leq , \rightarrow

`\DeclareSymbolFont{ $\langle name \rangle$ }` $\langle NFSS\ decl. \rangle$

Declares a named maths font such as operators from which symbols are defined with `\DeclareMathSymbol`.

Maths alphabet fonts Fonts for $ABC-xyz$, $\mathfrak{A}\mathfrak{B}\mathfrak{C}-\mathcal{X}\mathcal{Y}\mathcal{Z}$, etc.

```
\DeclareMathAlphabet{<cmd>}{NFSS decl.}
```

For commands such as `\mathbf`, accessed through maths mode that are unaffected by the current text font, and which are used for alphabetic symbols in the ASCII range.

```
\DeclareSymbolFontAlphabet{<cmd>}{<name>}
```

Alternative (and optimisation) for `\DeclareMathAlphabet` if a single font is being used for both alphabetic characters (as above) and symbols.

Maths ‘versions’ Different maths weights can be defined with the following, switched in text with the `\mathversion{<maths version>}` command.

```
\SetSymbolFont{<name>}{<maths version>}{NFSS decl.}
\SetMathAlphabet{<cmd>}{<maths version>}{NFSS decl.}
```

Maths symbols Symbol definitions in maths for both characters (=) and macros (`\eqdef`): `\DeclareMathSymbol{<symbol>}{<type>}{<named font>}{<slot>}` This is the macro that actually defines which font each symbol comes from and how they behave.

Delimiters and radicals use wrappers around TeX’s `\delimiter`/`\radical` primitives, which are re-designed in X_YTeX. The syntax used in L^ATeX’s NFSS is therefore not so relevant here.

Delimiters A special class of maths symbol which enlarge themselves in certain contexts.

```
\DeclareMathDelimiter{<symbol>}{<type>}{<sym. font>}{<slot>}{<sym. font>}{<slot>}
```

Radicals Similar to delimiters (`\DeclareMathRadical` takes the same syntax) but behave ‘weirdly’. `\sqrt` might very well be the only one.

In those cases, glyph slots in *two* symbol fonts are required; one for the small (‘regular’) case, the other for situations when the glyph is larger. This is not the case in X_YTeX.

Accents are not included yet.

Summary For symbols, something like:

```
\def\DeclareMathSymbol#1#2#3#4{%
\global\mathchardef#1"\mathchar@type#2
\expandafter\hexnumber@\csname sym#2\endcsname
{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}}
```


For characters, something like:

```
\def\DeclareMathSymbol#1#2#3#4{%
  \global\mathcode`#1"\mathchar@type#2
  \expandafter\hexnumber@\csname sym#2\endcsname
  {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}}
```

File IV

Some manner of unit testing

Some of the examples in the documentation are actually set up as unit tests, where multiple maths alphabets are placed on top of each other to ensure that various input methods result in the same output.

B The regular weight alphabets

For regular weight alphabets, we test the resolution from upright/italic math source to unified-shape output.

```
1 <(*test)
2 \documentclass{article}
3 \usepackage[a6paper]{geometry}
4 \usepackage{fontspec}
5 \setmainfont{FPL Neu}
6 \usepackage{unicode-math}
7 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
8 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
9 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
10 \def\upgreek{αβγδεζηθικλμνξοπρστυφχψω}
11 \def\itLatin{abcdefghijklmnopqrstuvwxyz}
12 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
13 \def\itGreek{αβγδεζηθικλμνξοπρστυφχψω}
14 \def\itgreek{αβγδεζηθικλμνξοπρστυφχψω}
15 \def\testmath#1{%
16   \makebox[\linewidth][l]{%
17     \makebox[0pt][l]{\csname up#1\endcsname$}%
18     \makebox[0pt][l]{\csname it#1\endcsname$}}
19 \begin{document}
20 \setmathfont[Colour=2255FF99]{Cambria Math}
21 \parindent=0pt
22 \voffset=-1in
23 \hoffset=-1in
```

```

24 \setbox0=\vbox{%
25 \testmath{Latin}\\
26 \testmath{latin}\\
27 \testmath{Greek}\\
28 \testmath{greek}}
29 \dimen0=\ht0
30 \advance\dimen0\dp0
31 \edef\papersize{papersize=\the\wd0,\the\dimen0}
32 \setbox255=\vbox{\special{\papersize}\box0}
33 \shipout\box255
34 \end{document}
35 </test>

```

We need three unit tests to produce the three variations of the `math-style` option. I'm guessing `literal` is working just fine, but it really needs a different test.

C The bold alphabets

For bold alphabets, it's a bit more complex. We also test `literal bold` to the bold produced from markup.

```

36 <*testbf>
37 \documentclass{article}
38 \usepackage[a6paper]{geometry}
39 \usepackage{fontspec}
40 \setmainfont{FPL Neu}
41 \usepackage{unicode-math}
42 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
43 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
44 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
45 \def\upgreek{0000000000000000π0000000000000000}
46 \def\itLatin{00000000000000000000000000000000}
47 \def\itlatin{00000000000000000000000000000000}
48 \def\itGreek{00000000000000000000000000000000}
49 \def\itgreek{00000000000000000000000000000000}
50 \def\bfupLatin{00000000000000000000000000000000}
51 \def\bfuplatin{00000000000000000000000000000000}
52 \def\bfupGreek{00000000000000000000000000000000}
53 \def\bfupgreek{00000000000000000000000000000000}
54 \def\bfitalatin{00000000000000000000000000000000}
55 \def\bfitalatin{00000000000000000000000000000000}
56 \def\bfitalGreek{00000000000000000000000000000000}
57 \def\bfitalgreek{00000000000000000000000000000000}
58 \providecommand\mathalphabet{\mathbf}
59 \def\testmath#1{%

```

```

60 \makebox[\linewidth][l]{%
61   \makebox[0pt][l]{\mathalphabet{\csname up#1\endcsname}}}%
62   \makebox[0pt][l]{\mathalphabet{\csname it#1\endcsname}}}%
63   \makebox[0pt][l]{\csname bfup#1\endcsname}}%
64   \makebox[0pt][l]{\csname bfit#1\endcsname}}%
65 }%
66 \begin{document}
67 \setmathfont[Colour=2255FF55]{Cambria Math}
68 \parindent=0pt
69 \voffset=-1in
70 \hoffset=-1in
71 \setbox0=\vbox{%
72   \testmath{Latin}\\
73   \testmath{latin}\\
74   \testmath{Greek}\\
75   \testmath{greek}}
76 \dimen0=\ht0
77 \advance\dimen0\dp0
78 \edef\papersize{\papersize=\the\wd0,\the\dimen0}
79 \setbox255=\vbox{\special{\papersize}\box0}
80 \shipout\box255
81 \end{document}
82 </testbf>

```

Change History

v0.01	
General: Tidied up awk code	54
\addnolimits: Implemented for \nolimits processing	31
\um@nolimits: Implemented for \nolimits processing	31
\um@radicals: Implemented for more general radicals processing.	32
v0.1	
General: Ignore \text.. STIX commands.	53
v0.2	
\removenolimits: Implemented for \nolimits processing	32
\um@setmathalph: Beginning to sort out the input encoding.	40
\um@setmathcode: Beginning to sort out the input encoding.	39
v0.3	
General: Ignore \ipa.. and \tone.. accents.	53
\um@mathsymbol: Guttied and simplified. TODO: re-add robust checking.	18
\um@set@mathsymbol: Added \mathaccentsupport.	20

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	
\"	17
\@cclvi	202
\@elt	476–480, 484, 487, 489
\@empty	250, 251, 283, 298, 303–319, 324, 518, 521, 536, 555, 560
\@for	450, 460, 462, 520, 570, 575
\@gobble	224, 268, 473
\@ii	450, 457, 460, 461, 520, 521, 523, 525, 528, 533
\@jj	462, 469
\@marker	533, 550
\@ne	27
\@nil	211, 262, 265, 271, 533, 544–547
\@preamblecmds	199
\@tempa	129, 166, 185, 191, 263, 272, 514, 524, 546, 548, 560, 577, 581
\@tempb	129, 130, 166, 167, 185, 186, 191, 192, 514, 515, 549, 550, 555
\@tempcnta	571–573, 576, 580, 581
\@tempfont	274, 275
\@tempswafalse	522
\@tempswattrue	526, 529, 552, 557, 562, 567
\@um@bfliteraltrue	183
\@um@bfupGreekfalse	135, 168
\@um@bfupGreektrue	146, 157, 173, 178
\@um@bfupLatinfalse	137, 170
\@um@bfupLatintrue	148, 159, 175, 180
\@um@bfupgreekfalse	136, 147, 169, 174
\@um@bfupgreektrue	158, 179
\@um@bfuplatinfalse	138, 171
\@um@bfuplatintrue	149, 160, 176, 181
\@um@fontspec@featuretrue	253
\@um@initfalse	252
\@um@inittrue	516
\@um@literaltrue	164
\@um@ot@math@true	276
\@um@upGreekfalse	131
\@um@upGreektrue	142, 153
\@um@upLatinfalse	133, 144
\@um@upLatintrue	155
\@um@upNablafalse	139, 189
\@um@upNablatrue	150, 161, 187
\@um@upgreekfalse	132, 143
\@um@upgreektrue	154
\@um@uplatinfalse	134, 145, 156
\@um@uppartialfalse	140, 195
\@um@uppartialtrue	151, 162, 193
\@	10–13, 17, 23–33, 69–71
\^	33, 261, 264
Numbers	
\0	35
\	17–20, 23–32
A	
\A	36
\a	37
\active	209
\addnolimits	<u>481</u>
\advance	27, 29, 74
\alloc@	202
\Alpha	584
\alpha	609
\AtBeginDocument	276, 582
\awint	478
B	
\B	64
\begin	18, 63
\begingroup	208, 260, 269, 429, 486
\Beta	585
\beta	610
\bfitGreek	53
\bfitgreek	54
\bfitLatin	51
\bfitlatin	52
\bfupGreek	49
\bfupgreek	50

<code>\bfuplatin</code>	47	<code>\endcsname</code>	
<code>\bfuplatin</code>	48	. 16, 17, 21, 58–61, 205, 211, 214,	
<code>\box</code>	31, 32, 76, 77	216, 217, 224, 247, 254, 428, 430, 473	
C			
<code>\C</code>	53, 79	<code>\endgroup</code>	212, 267, 274, 432, 492
<code>\catcode</code>	209, 261, 264	<code>\Epsilon</code>	588
<code>\chardef</code>	202	<code>\Eta</code>	590
<code>\Chi</code>	606	<code>\eta</code>	615
<code>\chi</code>	631	<code>\ExecuteOptionsX</code>	197
<code>\cirfnint</code>	478	<code>\expandafter</code> 21, 29, 205, 213, 216, 218,	
<code>\csname</code> 16, 17, 21, 58–61, 205, 211, 214,		219, 221, 223, 224, 428, 451–457,	
216, 217, 223, 247, 254, 428, 430, 473		463–469, 472, 473, 482–484,	
D			
<code>\DeclareMathVersion</code>	256	523, 525, 528, 533, 545, 546, 550	
<code>\DeclareSymbolFont</code>	292	F	
<code>\def</code> 6–14, 18, 21–24, 33, 35–127,		<code>\F</code>	66
178, 198–200, 202, 204, 206, 218,		<code>\f@size</code>	254
223, 228, 244, 255, 257–260, 266,		<code>\fi</code>	30, 140,
270, 275, 284, 428, 475, 482, 485,		147, 154, 165, 171, 172, 184, 190,	
487, 493, 495, 544, 546–549, 584–640		196, 219, 241–245, 282, 291, 301,	
<code>\def@cn</code>	20	302, 323, 341–344, 349, 354, 355,	
<code>\define@choicekey</code> 128, 166, 185, 191, 514		367, 369, 376, 393, 404, 415, 416,	
<code>\define@cmdkey</code>	510–513	418, 470, 490, 505, 517, 530, 531,	
<code>\define@key</code>	497	534, 540, 542, 543, 553, 558, 563, 568	
<code>\Delta</code>	587	<code>\fint</code>	478
<code>\delta</code>	612	<code>\font</code>	274
<code>\dimen</code>	28–30, 73–75	<code>\fontdimen</code>	201, 275
<code>\dimexpr</code>	201, 275	G	
<code>\do</code> 24, 450, 460, 462, 520, 570, 571, 575, 576		<code>\g</code>	74
<code>\documentclass</code>	1, 34	<code>\g@addto@macro</code>	472, 537, 539
<code>\dp</code>	29, 74	<code>\Gamma</code>	586
E		<code>\gamma</code>	611
<code>\E</code>	65	<code>\gdef</code>	216, 223, 228, 233, 262, 265
<code>\e</code>	73	<code>\glb@currsiz</code>	249
<code>\edef</code> 30, 75, 263, 289, 545, 546, 577		<code>\global</code>	210,
<code>\else</code> 135, 138, 144, 151, 163,		213, 229, 230, 234, 235, 240, 263, 266	
220, 226, 231, 236, 239, 277, 287,		H	
297, 321, 340–344, 347, 352, 365,		<code>\H</code>	54, 67, 80
367, 369, 373, 385, 399, 410, 417,		<code>\h</code>	36, 649, 654
488, 500, 527, 532, 538, 554, 559, 564		<code>\hoffset</code>	22, 67
<code>\encodingdefault</code>	293	<code>\ht</code>	28, 73
<code>\end</code>	33, 78	I	
		<code>\I</code>	68, 81
		<code>\if@tempswa</code>	535
		<code>\if@um@bfliteral</code>	13, 105, 356

<code>\if@um@bfupGreek</code>	14, 148, 370	<code>\Lambda</code>	594
<code>\if@um@bfupgreek</code>	15, 155, 377	<code>\lambda</code>	619
<code>\if@um@bfupLatin</code>	16, 136, 367	<code>\left</code>	494
<code>\if@um@bfuplatin</code>	17, 141, 369	<code>\left@primitive</code>	494, 495
<code>\if@um@fontspec@feature</code>	5, 498	<code>\let</code>	203, 249–251, 263, 268, 286, 290, 304–320, 322, 448, 494
<code>\if@um@init</code>	7, 294	<code>\linewidth</code>	15, 57
<code>\if@um@literal</code>	8, 326	<code>\long</code>	24
<code>\if@um@ot@math@</code>	6	<code>\lowercase</code>	263, 266
<code>\if@um@upGreek</code>	9, 343	<code>\lowint</code>	480
<code>\if@um@upgreek</code>	10, 344	M	
<code>\if@um@upLatin</code>	11, 341	<code>\M</code>	70
<code>\if@um@uplatin</code>	12, 342	<code>\makebox</code>	15–17, 57–61
<code>\if@um@upNabla</code>	18, 345, 394	<code>\mathaccent</code>	237
<code>\if@um@uppartial</code>	19, 350, 405	<code>\mathalpha</code>	474, 573
<code>\ifcase</code>	130, 167, 186, 192, 515	<code>\mathalphabet</code>	55, 58, 59
<code>\ifdim</code>	275	<code>\mathbb</code>	51–60
<code>\ifin@</code>	219, 222	<code>\mathbf</code> ..	55, 102–104, 106–134, 137, 139, 142, 143, 145, 146, 149, 150, 152, 153, 156–162, 164–170, 174–177
<code>\ifnum</code> ...	28, 461, 551, 556, 561, 565, 566	<code>\mathbffrak</code>	219–221
<code>\ifx</code>	207, 227, 232, 237, 283, 298, 303, 324, 488, 521, 524, 525, 528, 536, 550, 555, 560	<code>\mathbfit</code>	179–196
<code>\iiiint</code>	476	<code>\mathbfscr</code>	224–226
<code>\iiint</code>	476	<code>\mathbfsf</code>	229–242
<code>\iint</code>	476	<code>\mathbfsfit</code>	245–258
<code>\in@</code>	218, 221	<code>\mathbfup</code>	199–216
<code>\input</code>	272, 273, 295, 296, 299, 300	<code>\mathcal</code>	448
<code>\int</code>	476	<code>\mathchar@type</code>	215, 228, 230, 233, 235, 238, 240, 247
<code>\intBar</code>	478	<code>\mathclose</code>	232
<code>\intbar</code>	478	<code>\mathcode</code>	210
<code>\intcap</code>	480	<code>\mathfrak</code>	78–84
<code>\intclockwise</code>	477	<code>\mathgroup</code>	202
<code>\intcup</code>	480	<code>\mathit</code>	34–48
<code>\intlarhk</code>	479	<code>\mathop</code>	207
<code>\intx</code>	479	<code>\mathopen</code>	227, 495
<code>\Iota</code>	592	<code>\mathscr</code>	63–75, 448
<code>\iota</code>	617	<code>\mathsf</code>	87–89
<code>\itGreek</code>	12, 45	<code>\mathsf fit</code>	92–94
<code>\itgreek</code>	13, 46	<code>\mathstt</code>	97–99
<code>\itLatin</code>	10, 43	<code>\mathup</code>	19–31
<code>\itlatin</code>	11, 44	<code>\mddefault</code>	293
K		<code>\mitAlpha</code>	584
<code>\Kappa</code>	593	<code>\mitalpha</code>	609
<code>\kappa</code>	618	<code>\mitBeta</code>	585
L		<code>\mitbeta</code>	610
<code>\L</code>	69		

\backslash Phi	605	\backslash string	211,
\backslash phi	630		214, 216, 217, 224, 425, 473, 545, 546
\backslash Pi	599	\backslash strip@pt	201
\backslash pi	624	\backslash sumint	477
\backslash pointint	479		
\backslash ProcessOptionsX	198		
\backslash protect	504		
\backslash providecommand	20, 55		
\backslash ProvidesPackage	1		
\backslash Psi	607		
\backslash psi	632		
	Q		T
\backslash Q	57	\backslash Tau	603
		\backslash tau	628
		\backslash testmath	14, 24–27, 56, 69–72
		\backslash tf@size	265, 266
		\backslash the	30, 75
		\backslash Theta	591
		\backslash theta	616
		\backslash theum@fam	289
	R		U
\backslash R	58, 71, 82	\backslash um@addto@mathmap	452, 464, <u>471</u>
\backslash relax	25,	\backslash um@backslash	524, 545
	28, 130, 167, 186, 192, 201, 207,	\backslash um@Break	<u>22</u> , 29
	210, 215, 225, 227, 229, 230, 232,	\backslash um@char@num@range	251, 460, 536, 537, 539
	234, 235, 237, 238, 240, 247, 249,	\backslash um@char@range	
	261, 264, 272–275, 295, 296, 299,		... 250, 283, 298, 303, 324, 518, 520
	300, 461, 515, 525, 528, 550, 551,	\backslash um@def@itGreek	343, <u>661</u>
	556, 561, 565, 566, 572, 573, 580, 581	\backslash um@def@itgreek	344, <u>674</u>
\backslash removenolimits	<u>485</u>	\backslash um@def@itLatin	341, <u>645</u>
\backslash RequirePackage	3	\backslash um@def@itlatin	342, <u>647</u>
\backslash Rho	600	\backslash um@def@numbers	325, <u>641</u>
\backslash rho	625	\backslash um@def@upGreek	343, <u>656</u>
\backslash rrppolint	478	\backslash um@def@upgreek	344, <u>665</u>
\backslash rule	262, 273	\backslash um@def@upLatin	341, <u>643</u>
		\backslash um@def@uplatin	342, <u>652</u>
		\backslash um@firstchar	523, 546
		\backslash um@firstof	544–546
		\backslash um@fontdimen@percent	200
		\backslash um@FOR	<u>24</u> , 571, 576
		\backslash um@inp	570, 572, 575, 580
		\backslash um@Loop	<u>22</u> , 26
		\backslash um@mathbb	304
		\backslash um@mathbf	305, 313
		\backslash um@mathbffrak	316
		\backslash um@mathbfit	315
		\backslash um@mathbfscr	317
		\backslash um@mathbfsf	318
		\backslash um@mathbfsfit	319
		\backslash um@mathbfup	314
	S		
\backslash scantokens	263, 266		
\backslash scpolint	479		
\backslash setbox	23, 31, 68, 76		
\backslash setkeys	261		
\backslash setmainfont	4, 37		
\backslash SetMathCode	<u>246</u> , 474, 572		
\backslash setmathfont	19, 64, <u>248</u> , 276, 504		
\backslash sf@size	266, 269		
\backslash shipout	32, 77		
\backslash Sigma	602		
\backslash sigma	627		
\backslash special	31, 76		
\backslash sqint	479		
\backslash sqrt	493		
\backslash stepcounter	288		

<code>\um@mathfrak</code>	306	642, 644, 646, 648–650, 653, 654,	
<code>\um@mathit</code>	308	657–659, 662, 663, 666–672, 675–681	
<code>\um@mathmap@noparse</code>	320, <u>449</u>	<code>\um@setsinglemathalph</code> ... 320, 322, 578	
<code>\um@mathmap@parse</code>	322, <u>459</u>	<code>\um@setup@alphabets</code>	1, 419
<code>\um@mathscr</code>	309	<code>\um@setup@mathbb</code>	4, 50
<code>\um@mathsf</code>	310	<code>\um@setup@mathbf</code>	10, 101
<code>\um@mathsfitt</code>	311	<code>\um@setup@mathbffrak</code>	14, 218
<code>\um@mathsymbol</code>	<u>204</u> , 422, 426	<code>\um@setup@mathbfit</code>	12, 178
<code>\um@mathsymbol@noparse</code>	286, <u>421</u>	<code>\um@setup@mathbfscr</code>	13, 223
<code>\um@mathsymbol@parse</code>	290, <u>423</u>	<code>\um@setup@mathbfsf</code>	15, 228
<code>\um@mathtt</code>	312	<code>\um@setup@mathbfsfit</code>	16, 244
<code>\um@mathup</code>	275, 307	<code>\um@setup@mathbfup</code>	11, 198
<code>\um@mk@alph</code>	<u>427</u>	<code>\um@setup@mathfrak</code>	6, 77
<code>\um@mk@math</code>	427, 433–447	<code>\um@setup@mathit</code>	3, 33
<code>\um@mversion</code>	255, 256	<code>\um@setup@mathscr</code>	5, 62
<code>\um@nolimits</code>	218, <u>475</u> , 483, 484, 491	<code>\um@setup@mathsf</code>	7, 86
<code>\um@PackageError</code>	32	<code>\um@setup@mathsfitt</code>	8, 91
<code>\um@PackageInfo</code>	34, 285, 425	<code>\um@setup@mathtt</code>	9, 96
<code>\um@PackageWarning</code>	33, 278	<code>\um@setup@mathup</code>	2, 18
<code>\um@parse@range</code>	533, <u>547</u>	<code>\um@symfont</code>	
<code>\um@parse@term</code>	424, <u>519</u>		284, 289, 292, 422, 426, 455, 467, 573
<code>\um@Pool</code>	22, 23, 31	<code>\um@usv@bbLatin</code>	45, 52
<code>\um@radicals</code>	221, <u>493</u>	<code>\um@usv@bblatin</code>	46, 60
<code>\um@resolve@greek</code>	<u>582</u>	<code>\um@usv@bbnum</code>	44, 51
<code>\um@scanactivedef</code>	211, <u>260</u>	<code>\um@usv@bfDigamma</code>	91, 117
<code>\um@scancharlet</code>	<u>260</u> , 271	<code>\um@usv@bfdigamma</code>	98, 125
<code>\um@ScriptFeatures</code>	257, 268	<code>\um@usv@bffrakLatin</code>	68, 220
<code>\um@ScriptFont</code>	259, 267	<code>\um@usv@bffraklatin</code>	69, 221
<code>\um@ScriptScriptFeatures</code> ...	258, 271	<code>\um@usv@bfGreek</code>	62,
<code>\um@ScriptScriptFont</code>	260, 270		110, 149, 186, 202, 206, 361, 371, 374
<code>\um@set@mathsymbol</code>	205, <u>206</u>	<code>\um@usv@bfgreek</code>	63,
<code>\um@setmathalph</code>			112, 156, 187, 203, 207, 362, 378, 386
...	19–31, 34–48, 51–60, 63–75,	<code>\um@usv@bfitGreek</code>	
	78–84, 87–89, 92–94, 97–99,		66, 111, 152, 182, 186, 363, 371, 374
	102–104, 106–134, 137, 139, 142,	<code>\um@usv@bfitgreek</code>	
	143, 145, 146, 149, 150, 152,		67, 113, 164, 183, 187, 364, 378, 386
	153, 156–162, 164–170, 174–177,	<code>\um@usv@bfith</code>	108, 114, 146
	179–196, 199–216, 219–221,	<code>\um@usv@bfitLatin</code>	
	224–226, 229–242, 245–258, <u>574</u>		64, 107, 139, 180, 184, 359, 366, 367
<code>\um@setmathcode</code>		<code>\um@usv@bfitlatin</code>	
...	327–339, 346, 348, 351, 353,		65, 109, 145, 181, 185, 360, 368, 369
	357–364, 366, 368, 371, 372, 374,	<code>\um@usv@bfitNabla</code>	
	375, 378–384, 386–392, 395–398,		119, 127, 189, 396, 400, 401
	400–403, 406–409, 411–414, 569,	<code>\um@usv@bfitpartial</code>	
			125, 128, 190, 407, 411, 412

$\backslash\mathrm{um@usv@bfitvarepsilon}$	$\backslash\mathrm{um@usv@bfvarrho}$
..... 110, 129, 165, 191, 379, 387 96, 123, 161, 215, 383, 391
$\backslash\mathrm{um@usv@bfitvarkappa}$	$\backslash\mathrm{um@usv@bfvarTheta}$
..... 112, 131, 167, 193, 381, 389 90, 115, 150, 208, 372, 375
$\backslash\mathrm{um@usv@bfitvarphi}$	$\backslash\mathrm{um@usv@bfvartheta}$
..... 113, 132, 168, 194, 382, 390 93, 120, 158, 212, 380, 388
$\backslash\mathrm{um@usv@bfitvarpi}$	$\backslash\mathrm{um@usv@Digamma}$
..... 115, 134, 170, 196, 384, 392 82, 103, 117
$\backslash\mathrm{um@usv@bfitvarrho}$	$\backslash\mathrm{um@usv@digamma}$
..... 114, 133, 169, 195, 383, 391 89, 104, 125
$\backslash\mathrm{um@usv@bfitvarTheta}$	$\backslash\mathrm{um@usv@frakLatin}$
..... 109, 126, 153, 188, 372, 375 49, 78, 220
$\backslash\mathrm{um@usv@bfitvartheta}$	$\backslash\mathrm{um@usv@fraklatin}$
..... 111, 130, 166, 192, 380, 388 50, 84, 221
$\backslash\mathrm{um@usv@bfLatin}$	$\backslash\mathrm{um@usv@itGreek}$
..... 60,	21, 38, 42, 111, 149,
106, 137, 184, 200, 204, 357, 366, 367	152, 182, 202, 232, 248, 334, 657, 662
$\backslash\mathrm{um@usv@bflatin}$	$\backslash\mathrm{um@usv@itgreek}$
..... 61,	22, 39, 43, 113,
108, 142, 185, 201, 205, 358, 368, 369	156, 164, 183, 203, 233, 249, 666, 675
$\backslash\mathrm{um@usv@bfNabla}$ 116, 118, 209, 395, 396, 400	$\backslash\mathrm{um@usv@ith}$
$\backslash\mathrm{um@usv@bfnum}$	36, 37,
59, 102, 179, 199, 219, 224, 229, 245	99, 114, 143, 146, 330, 649, 650, 654
$\backslash\mathrm{um@usv@bfpartial}$	$\backslash\mathrm{um@usv@itLatin}$..
..... 118, 124, 210, 406, 407, 411	19, 34, 38, 52, 63,
$\backslash\mathrm{um@usv@bfscrLatin}$	78, 88, 93, 98, 107, 137, 139, 180,
..... 70, 225	200, 220, 225, 230, 246, 328, 644, 646
$\backslash\mathrm{um@usv@bfscrLatin}$	$\backslash\mathrm{um@usv@itlatin}$..
..... 71, 226	20, 35, 39, 60, 72,
$\backslash\mathrm{um@usv@bfsgreek}$	84, 89, 94, 99, 109, 142, 145, 181,
..... 75, 232	201, 221, 226, 231, 247, 329, 648, 653
$\backslash\mathrm{um@usv@bfsgreek}$	$\backslash\mathrm{um@usv@itNabla}$
..... 76, 233	23, 40,
$\backslash\mathrm{um@usv@bfsgreek}$	117, 127, 176, 189, 209, 337, 346, 348
..... 79, 248	$\backslash\mathrm{um@usv@itpartial}$
$\backslash\mathrm{um@usv@bfsgreek}$	24, 41,
..... 80, 249	123, 128, 177, 190, 210, 339, 351, 353
$\backslash\mathrm{um@usv@bfsgreek}$	$\backslash\mathrm{um@usv@itvarepsilon}$
..... 77, 246	26, 43,
$\backslash\mathrm{um@usv@bfsgreek}$	101, 129, 157, 165, 191, 211, 667, 676
..... 78, 247	$\backslash\mathrm{um@usv@itvarkappa}$
$\backslash\mathrm{um@usv@bfsgreek}$..	28, 45,
121, 398, 402, 403	103, 131, 159, 167, 193, 213, 669, 678
$\backslash\mathrm{um@usv@bfsgreek}$ 127, 409, 413, 414	$\backslash\mathrm{um@usv@itvarphi}$
$\backslash\mathrm{um@usv@bfsgreek}$	29, 46,
..... 73, 230	104, 132, 160, 168, 194, 214, 670, 679
$\backslash\mathrm{um@usv@bfsgreek}$	$\backslash\mathrm{um@usv@itvarpi}$
..... 74, 231	31, 48,
$\backslash\mathrm{um@usv@bfsgreek}$	106, 134, 162, 170, 196, 216, 672, 681
..... 120, 397, 398	$\backslash\mathrm{um@usv@itvarrho}$
$\backslash\mathrm{um@usv@bfsgreek}$	30, 47,
..... 72	105, 133, 161, 169, 195, 215, 671, 680
$\backslash\mathrm{um@usv@bfsgreek}$..	$\backslash\mathrm{um@usv@itvarTheta}$
126, 408, 409, 413	25,
$\backslash\mathrm{um@usv@bfuph}$	42, 100, 126, 150, 153, 188, 208, 663
..... 107, 143	$\backslash\mathrm{um@usv@itvartheta}$
$\backslash\mathrm{um@usv@bfvarepsilon}$	27, 44,
..... 92, 119, 157, 211, 379, 387	102, 130, 158, 166, 192, 212, 668, 677
$\backslash\mathrm{um@usv@bfvarkappa}$	$\backslash\mathrm{um@usv@Nabla}$
..... 94, 121, 159, 213, 381, 389	23, 40, 116,
$\backslash\mathrm{um@usv@bfvarphi}$	174, 189, 209, 235, 251, 336, 346, 348
..... 95, 122, 160, 214, 382, 390	$\backslash\mathrm{um@usv@num}$
$\backslash\mathrm{um@usv@bfvarpi}$ 97, 124, 162, 216, 384, 392	35, 51, 87, 92, 97,
	102, 179, 199, 219, 224, 229, 245, 642
	$\backslash\mathrm{um@usv@partial}$...
	24, 41, 118, 122,
	175, 190, 210, 236, 252, 338, 351, 353

