

# Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2009/09/11 v0.4

## Abstract

Warning! This package is experimental and subject to change without regard for backwards compatibility. Performance issues may be encountered until algorithms are refined.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>	<b>5.2</b>	Overcoming <code>\@on-</code> <code>lypreamble</code>	16
<b>2</b>	<b>Specification</b>	<b>2</b>	<b>5.3</b>	Other things	16
2.1	Using multiple fonts	3	<b>6</b>	<b>Fundamentals</b>	<b>17</b>
2.2	Script and scriptscript fonts/features	3	6.1	Enlarging the number of maths families	17
<b>3</b>	<b>Maths input</b>	<b>3</b>	6.2	<code>\DeclareMathSymbol</code> for unicode ranges	17
3.1	Miscellanea	4	6.3	The main <code>\setmathfont</code> macro	20
<b>4</b>	<b>Package options</b>	<b>4</b>	6.4	(Big) operators	27
4.1	Math ‘style’	4	6.5	Radicals	30
4.2	Bold switching	5	6.6	Delimiters	31
4.3	Symbols requiring spe- cial attention	6	6.7	Maths accents	34
<b>I</b>	<b>The unicode-math pack- age</b>	<b>9</b>	<b>7</b>	<b>Font features</b>	<b>35</b>
<b>5</b>	<b>Things we need</b>	<b>9</b>	7.1	OpenType maths font features	35
5.1	Package options	13	7.2	Script and scriptscript font options	35
			7.3	Range processing	35

<b>8 Maths alphabets mapping definitions</b>	<b>43</b>	A.1 Overview	59
8.1 Bold alphabets' character mappings	48	<b>III X<sub>Y</sub>TeX math font dimensions</b>	<b>61</b>
8.2 Definitions of the math symbols	54	<b>IV Some manner of unit testing</b>	<b>66</b>
8.3 Epilogue	54	<b>B The regular weight alphabets</b>	<b>66</b>
<b>II stix table data extraction</b>	<b>58</b>	<b>C The bold alphabets</b>	<b>67</b>
<b>A Documenting maths support in the NFSS</b>	<b>59</b>		

## 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<sub>Y</sub>TeX, although it is conjectured that some effort could be spent to create a cross-format package that would also work with LuaTeX.

## 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  $x$ , `\xi` to  $\xi$ , `\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<sup>A</sup>TeX will remain the same, the specification for choosing the version fonts will probably be an optional argument:

`\setmathfont[Version=Bold,<font features>]{<font name>}`

This has not been implemented yet.

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, font features]{font name}`

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

## 3.1 Miscellanea

### 3.1.1 Primes

Primes ( $x'$ ) may be input in several ways. You may use any combination of ascii straight quote ( ' ), unicode prime (  $\prime$  ), and `\prime`; when multiple primes occur next to each other, they chain together to form double, triple, or quadruple primes if the font contains pre-drawn glyphs. These may also be accessed with `\prime-double`, `\primetriples`, and `\primequadruple`.

If the font does not contain the pre-drawn glyphs or more than four primes are used, the single prime glyph is used multiple times with a negative kern to get the spacing right. There is no user interface to adjust this negative kern yet (because I haven't decided what it should look like); if you need to, write something like this:

```
\ExplSyntaxOn
\muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }
\ExplSyntaxOff
```

## 4 Package options

### 4.1 Math 'style'

Classically,  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$  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  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ '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

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

Package option	Example	
	Latin	Greek
<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>	$(a, z, B, X)$	$(\alpha, \beta, \Gamma, \Xi)$

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. Figure 1 on page 8 shows every character under the effect of this package option.

## 4.2 Bold switching

Similar as in the previous section, ISO standards differ somewhat to  $\text{\TeX}$ ’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  $\xi = (\xi_r, \xi_\varphi, \xi_\theta)$ . Confusingly, the syntax in  $\text{\LaTeX}$  has been different for these two examples: `\mathbf{f}` in the former (‘ $\mathbf{M}$ ’), and `\bm` (or `\boldsymbol`, deprecated) in the latter (‘ $\xi$ ’).

In `unicode-math`, the `\mathbf{f}` 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.

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

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

Table 3: The various forms of nabla.

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

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. Figure 2 on page 8 shows every character under the effect of this package option.

### 4.3 Symbols requiring special attention

**Nabla** The symbol  $\nabla$  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).  $\text{\TeX}$  classically uses an upright nabla, but iso standards differ (I think). The package options `nabla=upright` and `nabla=italic` switch between the two choices. This is then inherited through `\mathbf{f}`, `\mathit{f}` and `\mathup{f}` 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** The same applies to the symbols `U+2202: PARTIAL DIFFERENTIAL` and `U+1D715: MATH ITALIC PARTIAL DIFFERENTIAL`.

At time of writing, both the Cambria Math and STIX fonts display these two glyphs in the same italic style, but this is hopefully a bug that will be corrected in the future — the 'plain' partial differential should really have an upright shape.

Table 4: The various forms of the partial differential. Note that in the fonts used to display these glyphs, the first upright partial is incorrectly shown in an italic style.

Description		Glyph
Regular	Upright	$\partial$
	Italic	$\partial$
Bold	Upright	$\partial$
	Italic	$\partial$
Sans bold	Upright	$\partial$
	Italic	$\partial$

Use the `partial=upright` or `partial=italic` package options to specify which one you would like. The default is (always, unless someone requests and argues otherwise) `partial=italic`.<sup>1</sup>

See table 4 for the variations on the partial differential symbol.

**Epsilon and phi:  $\varepsilon$  vs.  $\epsilon$  and  $\varphi$  vs.  $\phi$**   $\text{\TeX}$  defines `\epsilonpsilon` to look like  $\epsilon$  and `\varepsilonpsilon` to look like  $\varepsilon$ . The Unicode glyph directly after delta and before zeta is ‘epsilon’ and looks like  $\varepsilon$ ; there is a subsequent variant of epsilon that looks like  $\epsilon$ . This creates a problem. People who use unicode input won’t want their glyphs transforming;  $\text{\TeX}$  users will be confused that what they think as ‘normal epsilon’ is actual the ‘variant epsilon’. And the same problem exists for ‘phi’.

We have a package option to control this behaviour. With `vargreek-shape=TeX`, `\phi` and `\epsilonpsilon` produce  $\phi$  and  $\varepsilon$  and `\varphi` and `\varepsilonpsilon` produce  $\phi$  and  $\epsilon$ . With `vargreek-shape=unicode`, these symbols are swapped. Note, however, that unicode characters are not affected by this option. That is, no remapping occurs of the characters/glyphs, only the control sequences.

Unless `math-style=literal` is in effect, the default is to use `vargreek-shape=TeX`.

U+3B5: GREEK SMALL LETTER EPSILON

U+3F5: GREEK LUNATE EPSILON SYMBOL

U+3C6: GREEK SMALL LETTER PHI

U+3D5: GREEK SMALL LETTER SCRIPT PHI

**Normalising some input characters** I believe all variant forms should be used as legal input that is normalised to a consistent output glyph, because we want to be fault-tolerant in the input. Here are the duplicates:

<sup>1</sup>A good argument would revolve around some international standards body recommending upright over italic. I just don’t have the time right now to look it up.

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\rho\sigma\tau\upsilon\phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\vartheta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\Box\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\rho\sigma\tau\upsilon\phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\vartheta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\Box\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\rho\sigma\tau\upsilon\phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\Box\zeta\eta\theta\vartheta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega$

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

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\text{R}\text{O}\Sigma\text{T}\text{Y}\Phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega\epsilon\vartheta\kappa\phi\rho\omega$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\text{R}\text{O}\Sigma\text{T}\text{Y}\Phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega\epsilon\vartheta\kappa\phi\rho\omega$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $AB\Gamma\Delta EZH\Theta\text{I}\text{K}\Lambda\text{M}\text{N}\Xi\text{O}\Pi\text{R}\text{O}\Sigma\text{T}\text{Y}\Phi\chi\psi\Omega$   
 $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\pi\rho\varsigma\sigma\tau\upsilon\phi\chi\psi\omega\epsilon\vartheta\kappa\phi\rho\omega$

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

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



U+251: LATIN SMALL LETTER ALPHA U+25B: LATIN SMALL LETTER EPSILON U+263:  
 LATIN SMALL LETTER GAMMA U+269: LATIN SMALL LETTER IOTA U+278: LATIN SMALL LET-  
 TER PHI U+28A: LATIN SMALL LETTER UPSILON U+190: LATIN CAPITAL LETTER EPSILON  
 U+194: LATIN CAPITAL LETTER GAMMA U+196: LATIN CAPITAL LETTER IOTA U+1B1: LATIN  
 CAPITAL LETTER UPSILON

## File I

# The unicode-math package

This is the package.

```
1 \ProvidesPackage{unicode-math}
2 [2009/09/11 v0.4 Unicode maths in XeLaTeX]
```

## 5 Things we need

### Packages

```
3 \RequirePackage{expl3}[2009/08/12]
4 \RequirePackage{xparse}[2009/08/31]
5 \RequirePackage{fontspec}
```

Start using L<sup>A</sup>T<sub>E</sub>X3 — finally!

```
6 \ExplSyntaxOn
```

### Counters and conditionals

```
7 \newcounter{um@fam}
8 \newif\if@um@fontspec@feature
9 \newif\if@um@ot@math@
```

For math-style:

```
10 \newif\if@um@literal
11 \newif\if@um@upGreek
12 \newif\if@um@upgreek
13 \newif\if@um@upLatin
14 \newif\if@um@uplatin
```

For bold-style:

```
15 \newif\if@um@bfliteral
16 \newif\if@um@bfupGreek
17 \newif\if@um@bfupgreek
18 \newif\if@um@bfupLatin
19 \newif\if@um@bfuplatin
```

For nabla:

```
20 \newif\if@um@pNabla
21 \newif\if@um@ppartial
22 \bool_new:N \g_um_texgreek_bool
```

## Programming niceties

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

## Shortcuts

```
25 \newcommand\um@PackageError[2]{\PackageError{unicode-math}{#1}{#2}}
26 \newcommand\um@PackageWarning[1]{\PackageWarning{unicode-math}{#1}}
27 \newcommand\um@PackageInfo[1]{\PackageInfo{unicode-math}{#1}}
```

### 5.0.1 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.<sup>2</sup>

```
28 \def\um@usv@num{\0}
29 \def\um@usv@upLatin{\A}
30 \def\um@usv@uplatin{\a}
31 \def\um@usv@itLatin{"1D434}
32 \def\um@usv@itlatin{"1D44E}
33 \def\um@usv@upGreek{"391}
34 \def\um@usv@upgreek{"3B1}
35 \def\um@usv@itGreek{"1D6E2}
36 \def\um@usv@itgreek{"1D6FC}
37 \def\um@usv@bbnum{"1D7D8}
38 \def\um@usv@bbLatin{"1D538}
39 \def\um@usv@bblatin{"1D552}
40 \def\um@usv@scrLatin{"1D49C}
41 \def\um@usv@scrlatin{"1D4B6}
42 \def\um@usv@frakLatin{"1D504}
43 \def\um@usv@fraklatin{"1D51E}
44 \def\um@usv@s fnum{"1D7E2}
45 \def\um@usv@s fLatin{"1D5A0}
46 \def\um@usv@s flatin{"1D5BA}
47 \def\um@usv@s fitLatin{"1D608}
48 \def\um@usv@s fitlatin{"1D622}
49 \def\um@usv@ttnum{"1D7F6}
50 \def\um@usv@ttLatin{"1D670}
51 \def\um@usv@ttlLatin{"1D68A}
```

---

<sup>2</sup>'u.s.v.' stands for 'unicode scalar value'.

Bold:

```
52 \def\um@usv@b fnum{" 1D7CE}
53 \def\um@usv@b fLatin{" 1D400}
54 \def\um@usv@b flatin{" 1D41A}
55 \let\um@usv@b fuplatin\um@usv@b flatin
56 \def\um@usv@b fGreek{" 1D6A8}
57 \def\um@usv@b fgreek{" 1D6C2}
58 \def\um@usv@b fitLatin{" 1D468}
59 \def\um@usv@b fitlatin{" 1D482}
60 \def\um@usv@b fitGreek{" 1D71C}
61 \def\um@usv@b fitgreek{" 1D736}
62 \def\um@usv@b ffrakLatin{" 1D56C}
63 \def\um@usv@b ffraklatin{" 1D586}
64 \def\um@usv@b fscrLatin{" 1D4D0}
65 \def\um@usv@b fscrlatin{" 1D4EA}
66 \def\um@usv@b fs fnum{" 1D7EC}
67 \def\um@usv@b fs fLatin{" 1D5D4}
68 \def\um@usv@b fs flatin{" 1D5EE}
69 \let\um@usv@b fs fuplatin\um@usv@b fs flatin
70 \def\um@usv@b fs fGreek{" 1D756}
71 \def\um@usv@b fs fgreek{" 1D770}
72 \def\um@usv@b fs fitLatin{" 1D63C}
73 \def\um@usv@b fs fitlatin{" 1D656}
74 \def\um@usv@b fs fitGreek{" 1D790}
75 \def\um@usv@b fs fitgreek{" 1D7AA}
```

Greek variants:

```
76 \def\um@usv@varTheta{" 3F4}
77 \def\um@usv@Digamma{" 3DC}
78 \def\um@usv@varepsilon{" 3F5}
79 \def\um@usv@vartheta{" 3D1}
80 \def\um@usv@varkappa{" 3F0}
81 \def\um@usv@varphi{" 3D5}
82 \def\um@usv@varrho{" 3F1}
83 \def\um@usv@varpi{" 3D6}
84 \def\um@usv@digamma{" 3DD}
85 \tl_new:Nn \g_um_up_epsilon_letter_usv {" 25B}
86 \tl_new:Nn \g_um_up_epsilon_symbol_usv {" 3F5}
87 \tl_new:Nn \g_um_up_phi_letter_usv {" 3C6}
88 \tl_new:Nn \g_um_up_phi_symbol_usv {" 3D5}
```

Bold:

```
89 \def\um@usv@b fvarTheta{" 1D6B9}
90 \def\um@usv@b fDigamma{" 1D7CA}
91 \def\um@usv@b fvarepsilon{" 1D6DC}
92 \def\um@usv@b fvartheta{" 1D6DD}
93 \def\um@usv@b fvarkappa{" 1D6DE}
```

```

94 \def\um@usv@b fvarphi{"1D6DF}
95 \def\um@usv@b fvarrho{"1D6E0}
96 \def\um@usv@b fvarpi{"1D6E1}
97 \def\um@usv@b fdigamma{"1D7CB}
98 \tl_new:Nn \g_um_bfup_epsilon_letter_usv {"1D6C6}
99 \tl_new:Nn \g_um_bfup_epsilon_symbol_usv {"1D6DC}
100 \tl_new:Nn \g_um_bfup_phi_letter_usv {"1D6D7}
101 \tl_new:Nn \g_um_bfup_phi_symbol_usv {"1D6DF}

```

Italic Greek variants:

```

102 \def\um@usv@ith{"210E}
103 \def\um@usv@itvarTheta{"1D6F3}
104 \def\um@usv@itvarepsilon{"1D716}
105 \def\um@usv@itvartheta{"1D717}
106 \def\um@usv@itvarkappa{"1D718}
107 \def\um@usv@itvarphi{"1D719}
108 \def\um@usv@itvarrho{"1D71A}
109 \def\um@usv@itvarpi{"1D71B}
110 \tl_new:Nn \g_um_it_epsilon_symbol_usv {"1D716}
111 \tl_new:Nn \g_um_it_epsilon_letter_usv {"1D700}
112 \tl_new:Nn \g_um_it_phi_symbol_usv {"1D719}
113 \tl_new:Nn \g_um_it_phi_letter_usv {"1D711}

```

Bold:

```

114 \def\um@usv@b fuph{"1D421}
115 \def\um@usv@b fi th{"1D489}
116 \def\um@usv@b fitvarTheta{"1D72D}
117 \def\um@usv@b fitvarepsilon{"1D750}
118 \def\um@usv@b fitvartheta{"1D751}
119 \def\um@usv@b fitvarkappa{"1D752}
120 \def\um@usv@b fitvarphi{"1D753}
121 \def\um@usv@b fitvarrho{"1D754}
122 \def\um@usv@b fitvarpi{"1D755}
123 \tl_new:Nn \g_um_bfit_epsilon_letter_usv {"1D73A}
124 \tl_new:Nn \g_um_bfit_epsilon_symbol_usv {"1D750}
125 \tl_new:Nn \g_um_bfit_phi_letter_usv {"1D74B}
126 \tl_new:Nn \g_um_bfit_phi_symbol_usv {"1D753}

```

Nabla:

```

127 \def\um@usv@Nabla{"2207}
128 \def\um@usv@itNabla{"1D6FB}
129 \def\um@usv@b fNabla{"1D6C1}
130 \def\um@usv@b fitNabla{"1D735}
131 \def\um@usv@b fs fNabla{"1D76F}
132 \def\um@usv@b fs fitNabla{"1D7A9}

```

Partial:

```

133 \def\um@usv@partial{"2202}

```

```

134 \def\um@usv@i tpartial{"1D715}
135 \def\um@usv@b fpartial{"1D6DB}
136 \def\um@usv@b fi tpartial{"1D74F}
137 \def\um@usv@b fs fpartial{"1D789}
138 \def\um@usv@b fs fi tpartial{"1D7C3}

```

## 5.1 Package options

xkeyval's package support is used here.

### math-style

```

139 \define@choicekey*{unicode-math.sty}
140   {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{
141   \ifcase\@tempb\relax
142     \@um@upGreekfalse
143     \@um@upgreekfalse
144     \@um@upLatinfalse
145     \@um@uplatinfalse
146     \@um@b fupGreekfalse
147     \@um@b fupgreekfalse
148     \@um@uppartialfalse
149     \@um@b fupLatinfalse
150     \@um@b fuplatinfalse
151     \@um@upNablafalse
152     \bool_set_false:N \g_um_texgreek_bool
153   \or
154     \@um@upGreektrue
155     \@um@upgreekfalse
156     \@um@upLatinfalse
157     \@um@uplatinfalse
158     \@um@b fupGreektrue
159     \@um@b fupgreekfalse
160     \@um@uppartialfalse
161     \@um@b fupLatintrue
162     \@um@b fuplatintrue
163     \@um@upNabltrue
164     \bool_set_true:N \g_um_texgreek_bool
165   \or
166     \@um@upGreektrue
167     \@um@upgreektrue
168     \@um@upLatintrue
169     \@um@uplatinfalse
170     \@um@b fupGreektrue
171     \@um@b fupgreektrue
172     \@um@uppartialtrue
173     \@um@b fupLatintrue

```

```

174 \um@b fuplatintrue
175 \um@upNablatrue
176 \bool_set_false:N \g_um_texgreek_bool
177 \or
178 \um@literaltrue
179 \um@b fliteraltrue
180 \bool_set_false:N \g_um_texgreek_bool
181 \fi
182 }

```

### bold-style

```

183 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso, tex, french, literal}{
184 \ifcase\@tempb\relax
185 \um@b fupGreekfalse
186 \um@b fupgreekfalse
187 \um@uppartialfalse
188 \um@b fupLatinfalse
189 \um@b fuplatinfalse
190 \or
191 \um@b fupGreektrue
192 \um@b fupgreekfalse
193 \um@uppartialfalse
194 \um@b fupLatintrue
195 \um@b fuplatintrue
196 \or
197 \um@b fupGreektrue
198 \um@b fupgreektrue
199 \um@uppartialtrue
200 \um@b fupLatintrue
201 \um@b fuplatintrue
202 \or
203 \um@b fliteraltrue
204 \fi
205 }

```

### Symbol obliqueness

```

206 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright, italic}{
207 \ifcase\@tempb\relax
208 \um@upNablatrue
209 \or
210 \um@upNablafalse
211 \fi
212 }
213 \cs_set:Nn \um_setup_nabla: {
214 \if@um@upNabla

```

```

215 \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@Nabla }
216 \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfNabla }
217 \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfNabla }
218 \else
219 \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@itNabla }
220 \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfitNabla }
221 \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfitNabla }
222 \fi
223 }
224 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{
225 \ifcase\@tempb\relax
226 \um@uppartialtrue
227 \or
228 \um@uppartialfalse
229 \fi
230 }
231 \cs_set:Nn \um_setup_partial: {
232 \if@um@uppartial
233 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@partial }
234 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfpartial }
235 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfpartial }
236 \else
237 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@itpartial }
238 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfitpartial }
239 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfitpartial }
240 \fi
241 }

```

### Epsilon and phi shapes

```

242 \define@choicekey*{unicode-math.sty}{vargreek-shape}[\@tempa\@tempb]{unicode,TeX}{
243 \ifcase\@tempb\relax
244 \bool_set_false:N \g_um_texgreek_bool
245 \or
246 \bool_set_true:N \g_um_texgreek_bool
247 \fi
248 }
249 \cs_set:Nn \um_setup_vargreek: {
250 \bool_if:NTF \g_um_texgreek_bool {
251 \num_set_eq:NN \g_um_up_phi_usv \g_um_up_phi_symbol_usv
252 \num_set_eq:NN \g_um_up_varphi_usv \g_um_up_phi_letter_usv
253 \num_set_eq:NN \g_um_it_phi_usv \g_um_it_phi_symbol_usv
254 \num_set_eq:NN \g_um_it_varphi_usv \g_um_it_phi_letter_usv
255 \num_set_eq:NN \g_um_bfup_phi_usv \g_um_bfup_phi_symbol_usv
256 \num_set_eq:NN \g_um_bfup_varphi_usv \g_um_bfup_phi_letter_usv
257 \num_set_eq:NN \g_um_bfit_phi_usv \g_um_bfit_phi_symbol_usv
258 \num_set_eq:NN \g_um_bfit_varphi_usv \g_um_bfit_phi_letter_usv

```

```

259 \num_set_eq: NN \g_um_up_epsilon_usv \g_um_up_epsilon_symbol_usv
260 \num_set_eq: NN \g_um_up_varepsilon_usv \g_um_up_epsilon_letter_usv
261 \num_set_eq: NN \g_um_it_epsilon_usv \g_um_it_epsilon_symbol_usv
262 \num_set_eq: NN \g_um_it_varepsilon_usv \g_um_it_epsilon_letter_usv
263 \num_set_eq: NN \g_um_bfup_epsilon_usv \g_um_bfup_epsilon_symbol_usv
264 \num_set_eq: NN \g_um_bfup_varepsilon_usv \g_um_bfup_epsilon_letter_usv
265 \num_set_eq: NN \g_um_bfit_epsilon_usv \g_um_bfit_epsilon_symbol_usv
266 \num_set_eq: NN \g_um_bfit_varepsilon_usv \g_um_bfit_epsilon_letter_usv
267 }{
268 \num_set_eq: NN \g_um_up_varphi_usv \g_um_up_phi_symbol_usv
269 \num_set_eq: NN \g_um_up_phi_usv \g_um_up_phi_letter_usv
270 \num_set_eq: NN \g_um_it_varphi_usv \g_um_it_phi_symbol_usv
271 \num_set_eq: NN \g_um_it_phi_usv \g_um_it_phi_letter_usv
272 \num_set_eq: NN \g_um_bfup_varphi_usv \g_um_bfup_phi_symbol_usv
273 \num_set_eq: NN \g_um_bfup_phi_usv \g_um_bfup_phi_letter_usv
274 \num_set_eq: NN \g_um_bfit_varphi_usv \g_um_bfit_phi_symbol_usv
275 \num_set_eq: NN \g_um_bfit_phi_usv \g_um_bfit_phi_letter_usv
276 \num_set_eq: NN \g_um_up_varepsilon_usv \g_um_up_epsilon_symbol_usv
277 \num_set_eq: NN \g_um_up_epsilon_usv \g_um_up_epsilon_letter_usv
278 \num_set_eq: NN \g_um_it_varepsilon_usv \g_um_it_epsilon_symbol_usv
279 \num_set_eq: NN \g_um_it_epsilon_usv \g_um_it_epsilon_letter_usv
280 \num_set_eq: NN \g_um_bfup_varepsilon_usv \g_um_bfup_epsilon_symbol_usv
281 \num_set_eq: NN \g_um_bfup_epsilon_usv \g_um_bfup_epsilon_letter_usv
282 \num_set_eq: NN \g_um_bfit_varepsilon_usv \g_um_bfit_epsilon_symbol_usv
283 \num_set_eq: NN \g_um_bfit_epsilon_usv \g_um_bfit_epsilon_letter_usv
284 }
285 }
286 \ExecuteOptionsX{math-style=TeX}
287 \ProcessOptionsX

```

## 5.2 Overcoming \@onlypreamble

This will be refined later! Sort out which macros actually have to be removed from the \@preamblecmds token list. There is a macro to remove items from the \@preamblecmds list in gmutils.sty.

```

288 \def\@preamblecmds{}

```

## 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>

---

```

289 \def\um@fontdimen@percent#1#2{
290   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax
291 }

```

`\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.

```

292 \def\um@scaled@apply#1#2#3{
293   \ifx#1\scriptstyle
294     #2\um@fontdimen@percent{10}\um@font#3
295   \else
296     \ifx#1\scriptscriptstyle
297       #2\um@fontdimen@percent{11}\um@font#3
298     \else
299       #2#3%
300     \fi
301   \fi
302 }

```

## 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 `lftssbas.dtx`) we want to redefine

```

303 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cclvi}
304 \let\newfam\new@mathgroup

```

This is sufficient for L<sup>A</sup>T<sub>E</sub>X'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

```
305 \def \um@mathsymbol#1#2#3#4{
306   \expandafter\um@set@mathsymbol\csname sym#3\endcsname#1#2{#4}}
```

The final macros that actually define the maths symbol with X<sub>Y</sub>TeX 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.

```
307 \def\um@set@mathsymbol#1#2#3#4{
```

**Operators** In the examples following, say we're defining for the symbol  $\sum(\Sigma)$ .

```
308 \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 `\gobble` 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`.

```
309 \begingroup
310   \char_make_active:n {#4}
311   \global\mathcode#4="8000\relax
312   \um@scanactivedef #4 \@nil { \csname\string#2@op\endcsname }
313 \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`.

```
314 \expandafter\global\expandafter\XeTeXmathchardef
315   \csname\string#2@sym\endcsname
316   =" \mathchar@type#3 #1 #4\relax
```

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

```
317 \cs_gset:cpn { \string#2 @op } {
318   \csname\string#2@sym\endcsname
319   \expandafter\in@\expandafter#2\expandafter{\um@nolimits}
320   \ifin@
321     \expandafter\nolimits
322   \fi
323 }
```

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

```
324 \else
```

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

```
325 \expandafter\in@\expandafter#2\expandafter{\um@radicals,}
326 \ifin@
327 \cs_gset:cpn {\cs_to_str:N #2 sign} { \XeTeXradical #1 #4 \relax }
328 \else
```

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

```
329 \ifx\mathopen#3\relax
330 \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
331 \global\XeTeXdelcode#4=#1 #4\relax
332 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
333 \else
334 \ifx\mathclose#3\relax
335 \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
336 \global\XeTeXdelcode#4=#1 #4\relax
337 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
338 \else
```

**Accents**

```
339 \ifx\mathaccent#3\relax
340 \cs_gset:Npx #2 {\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}
341 \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.

```
342 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
343 \fi
344 \fi
345 \fi
346 \fi
347 \fi
348 }
```

`\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.

```
349 \newcommand\SetMathCode[4]{
350 \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax
351 }
```

---

A

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

---

### 6.3 The main `\setmathfont` macro

Here's the simplest usage:

---

$$Ax \triangleq \nabla \times \mathcal{L}$$

```
\setmathfont{Asana Math}
$Ax \eqdef \nabla \times \mscrZ$
```

---

An interesting (perhaps useless) example of the `Range` feature:

---

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

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

---

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

```
352 \DeclareDocumentCommand \setmathfont { O{} m } {
```

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

```
353 \let\glb@currsizel\relax
```

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

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

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

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

```
356 \@um@fontspec@featuretrue
```

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

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

- Set the name of the math version being defined. (obviously more needs to be done here!)

```
358 \def\um@mversion{normal}
359 \DeclareMathVersion{\um@mversion}
```

Define default font features for the script and scriptscript font. (This needs to be generalised so users can override it.)

```
360 \tl_set:Nn \l_um_script_features_tl {ScriptStyle}
361 \tl_set:Nn \l_um_sscript_features_tl {ScriptScriptStyle}
362 \tl_set:Nn \l_um_script_font_tl {#2}
363 \tl_set:Nn \l_um_sscript_font_tl {#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.

```
364 \setkeys*[um]{options}{#1}
365 \edef\@tempa{\noexpand\zf@fontspec{
366   Script = Math,
367   SizeFeatures = {
368     {Size = \tf@size-} ,
369     {Size = \sf@size-\tf@size ,
370      Font = \l_um_script_font_tl ,
371      \l_um_script_features_tl
372     } ,
373     {Size = -\sf@size ,
374      Font = \l_um_sscript_font_tl ,
375      \l_um_sscript_features_tl
376     }
377   },
378   \XKV@r m
379 }{#2}
380 }
381 \@tempa
```

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`:

```
382 \font\um@font="#2"\relax
383 \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt\relax
384   \um@ot@math@true
385 \else
```

```

386 \um@PackageWarning{
387   The~ font~ '#2' ~is~ not~ a~ valid~ OpenType~ maths~ font.~
388   Some~ maths~ features~ will~ not~ be~ available~ or~ behave~
389   in~ a~ substandard~ manner.
390 }
391 \fi

```

If we're defining the full unicode math repertoire, then we skip all the parsing processing needed if we're only defining a subset.

- Math symbols are defined with `\UnicodeMathSymbol`; see section §6.3.1 for the individual definitions

```

392 \ifx\um@char@range\@empty
393   \def\um@symfont{\um@allsym}
394   \um@PackageInfo{Defining~ the~ default~ maths~ font~ as~ '#2'}
395   \let \UnicodeMathSymbol \um@mathsymbol@noparse
396   \let \um_mathmap: Nnn \um_mathmap_noparse: Nnn
397   \cs_set_eq: NN \um_remap_symbol: nnn \um_remap_symbol_noparse: nnn
398   \cs_set_eq: NN \um_maybe_init_alphabet: n \um_init_alphabet: n
399 \else
400   \stepcounter{um@fam}
401   \edef\um@symfont{\um@fam\theum@fam}
402   \let \UnicodeMathSymbol \um@mathsymbol@parse
403   \let \um_mathmap: Nnn \um_mathmap_parse: Nnn
404   \cs_set_eq: NN \um_remap_symbol: nnn \um_remap_symbol_parse: nnn
405   \cs_set_eq: NN \um_maybe_init_alphabet: n \use_none: n
406 \fi

```

Now defined `\um@symfont` as the L<sup>A</sup>T<sub>E</sub>X math font to access everything:

```

407 \DeclareSymbolFont{\um@symfont}
408   {\encodingdefault}{\zf@family}{\mddefault}{\updefault}

```

And now we input every single maths char. See File II for the source to `unicode-math.tex` which is used to create `unicode-math-table.tex`.

```

409 \input{unicode-math-table.tex}

```

Finally,

- Set up shapes for italic/upright or ordinary/var symbols as per package options.
- Remap symbols that don't take their natural mathcode
- Activate any symbols that need to be math-active
- Setup all symbols not covered by the table (mostly alphanumerics)
- Setup the maths alphabets (`\mathbf` etc.)

```

410 \um_setup_shapes:
411 \um_remap_symbols:
412 \um_setup_mathactives:
413 \um_setup_alphanum:
414 \um_setup_alphabets:

End of the \setmathfont macro.

415 }

416 \cs_new:Nn \um_setup_shapes: {
417   \um_setup_nabla:
418   \um_setup_partial:
419   \um_setup_vargreek:
420 }

```

### 6.3.1 Functions for setting up symbols with mathcodes

`\um@mathsymbol@noparse`

```

421 \newcommand\um@mathsymbol@noparse[4]{
422   \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
423 }

```

`\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.

```

424 \newcommand\um@mathsymbol@parse[4]{
425   \um@parse@term{#1}{#2}{#3}{
426     %\um@PackageInfo{Defining \string#2 as mathchar #1}
427     \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
428   }
429 }

```

`\um_remap_symbols:`

This function is used to define the mathcodes for those chars which should be mapped to a different glyph than themselves.

```

430 \cs_new:Nn \um_remap_symbols: {
431   \um_remap_symbol:nnn{"2D}{\mathbin}{"02212}% hyphen to minus
432   \if@um@literal
433     \um_remap_symbol:nnn {\um@usv@Nabla}{\mathord}{\um@usv@Nabla}
434     \um_remap_symbol:nnn {\um@usv@itNabla}{\mathord}{\um@usv@itNabla}
435     \um_remap_symbol:nnn {\um@usv@partial}{\mathord}{\um@usv@partial}
436     \um_remap_symbol:nnn {\um@usv@itpartial}{\mathord}{\um@usv@itpartial}
437   \else
438     \um_remap_symbol:nnn {\um@usv@Nabla,\um@usv@itNabla}{\mathord}{\um_Nabla_up_or_it_usv}
439     \um_remap_symbol:nnn {\um@usv@partial,\um@usv@itpartial}{\mathord}{\um_partial_up_or_it_usv}
440   \fi

```

Some of these in the `bfliteral` block may be redundant, but that's okay:

```

441   \if@um@bfliteral

```

```

442 \um_remap_symbol: nnn {\um@usv@b fNabla }{\mathord}{\um@usv@b fNabla}
443 \um_remap_symbol: nnn {\um@usv@b fitNabla }{\mathord}{\um@usv@b fitNabla}
444 \um_remap_symbol: nnn {\um@usv@b fs fNabla }{\mathord}{\um@usv@b fs fNabla}
445 \um_remap_symbol: nnn {\um@usv@b fs fitNabla }{\mathord}{\um@usv@b fs fitNabla}
446 \um_remap_symbol: nnn {\um@usv@b fpartial }{\mathord}{\um@usv@b fpartial}
447 \um_remap_symbol: nnn {\um@usv@b fitpartial }{\mathord}{\um@usv@b fitpartial}
448 \um_remap_symbol: nnn {\um@usv@b fs fpartial }{\mathord}{\um@usv@b fs fpartial}
449 \um_remap_symbol: nnn {\um@usv@b fs fitpartial }{\mathord}{\um@usv@b fs fitpartial}
450 \else
451 \um_remap_symbol: nnn {\um@usv@b fNabla, \um@usv@b fitNabla}{\mathord}{\um_b fNabla_up_or_it_u
452 \um_remap_symbol: nnn {\um@usv@b fs fNabla, \um@usv@b fs fitNabla}{\mathord}{\um_b fs fNabla_up_o
453 \um_remap_symbol: nnn {\um@usv@b fpartial, \um@usv@b fitpartial}{\mathord}{\um_b fpartial_up_o
454 \um_remap_symbol: nnn {\um@usv@b fs fpartial, \um@usv@b fs fitpartial}{\mathord}{\um_b fs fpartia
455 \fi
456 }

```

Where `\um_remap_symbol: nnn` is defined to be one of these two, depending on the range setup:

```

457 \cs_new:Nn \um_remap_symbol_parse: nnn {
458   \um@parse@term {#3} {\@nil} {#2} {
459     \um_remap_symbol_noparse: nnn {#1} {#2} {#3}
460   }
461 }
462 \cs_new:Nn \um_remap_symbol_noparse: nnn {
463   \clist_map_inline:nn {#1} {
464     \SetMathCode {##1} {#2} {\um@symfont} {#3}
465   }
466 }

```

### 6.3.2 Active math characters

`\um_setup_mathactives:`

```

467 \cs_new:Nn \um_setup_mathactives: {
468   \um_make_mathactive: nNN {"2032} \primesingle \mathord
469 }

```

`\um_make_mathactive: nNN` Makes #1 a mathactive char, and gives cs #2 the meaning of mathchar #1 with class #3. You are responsible for giving active #1 a particular meaning!

```

470 \cs_new:Nn \um_make_mathactive: nNN {
471   \XeTeXmathchardef #2 = "\mathchar@type #3
472                               \csname sym\um@symfont\endcsname
473                               #1 \scan_stop:
474   \XeTeXmathcodenum #1 = "1FFFFF \scan_stop:
475 }

```



### 6.3.3 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.

`\um_setup_alphanum:` All symbols input that aren't defined directly in `unicode-math-table`.

```
476 \cs_set:Nn \um_setup_alphanum: {  
477   \ifx\um@char@range\empty  
478     \um@def@numbers
```

#### Normal weight

```
479   \if@um@literal  
480     \um_setup_literals:  
481   \else  
482     \if@um@upLatin\um@def@upLatin\else\um@def@itLatin\fi  
483     \if@um@uplatin\um@def@uplatin\else\um@def@itlatin\fi  
484     \if@um@upGreek\um@def@upGreek\else\um@def@itGreek\fi  
485     \if@um@upgreek\um@def@upgreek\else\um@def@itgreek\fi  
486   \fi
```

#### Bold

```
487   \if@um@bfliteral  
488     \um_setup_bf_literals:  
489   \else  
490     \if@um@bfupLatin  
491       \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfItLatin}{\um@usv@bfLatin}  
492     \else  
493       \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfItLatin}{\um@usv@bfItLatin}  
494     \fi  
495     \if@um@bfuplatin  
496       \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfItlatin}{\um@usv@bflatin}  
497     \else  
498       \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfItlatin}{\um@usv@bfItlatin}  
499     \fi  
500     \if@um@bfupGreek  
501       \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfItGreek}{\um@usv@bfGreek}  
502       \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfItvarTheta}{\um@usv@bfvarTheta}  
503     \else  
504       \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfItGreek}{\um@usv@bfItGreek}  
505       \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfItvarTheta}{\um@usv@bfItvarTheta}  
506     \fi  
507     \if@um@bfupgreek  
508       \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfItgreek}{\um@usv@bfGreek}  
509       \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfItvarepsilon}{\um@usv@bfvarepsilon}
```

```

510 \um@setmathcode{\um@usv@b fvartheta, \um@usv@b fitvartheta}{\um@usv@b fvartheta}
511 \um@setmathcode{\um@usv@b fvarkappa, \um@usv@b fitvarkappa}{\um@usv@b fvarkappa}
512 \um@setmathcode{\um@usv@b fvarphi, \um@usv@b fitvarphi}{\um@usv@b fvarphi}
513 \um@setmathcode{\um@usv@b fvarrho, \um@usv@b fitvarrho}{\um@usv@b fvarrho}
514 \um@setmathcode{\um@usv@b fvarpi, \um@usv@b fitvarpi}{\um@usv@b fvarpi}
515 \else
516 \um@setmathcode[25]{\um@usv@b fgreek, \um@usv@b fitgreek}{\um@usv@b fitgreek}
517 \um@setmathcode{\um@usv@b fvarepsilon, \um@usv@b fitvarepsilon}{\um@usv@b fitvarepsilon}
518 \um@setmathcode{\um@usv@b fvartheta, \um@usv@b fitvartheta}{\um@usv@b fitvartheta}
519 \um@setmathcode{\um@usv@b fvarkappa, \um@usv@b fitvarkappa}{\um@usv@b fitvarkappa}
520 \um@setmathcode{\um@usv@b fvarphi, \um@usv@b fitvarphi}{\um@usv@b fitvarphi}
521 \um@setmathcode{\um@usv@b fvarrho, \um@usv@b fitvarrho}{\um@usv@b fitvarrho}
522 \um@setmathcode{\um@usv@b fvarpi, \um@usv@b fitvarpi}{\um@usv@b fitvarpi}
523 \fi
524 \fi
525 \else
: TODO : what is supposed to happen here?
526 \fi
527 }

```

### 6.3.4 Functions for setting up the maths alphabets

`\um_mathmap_noparse: Nnn` #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.

```

528 \cs_set:Nn \um_mathmap_noparse:Nnn {
529   \clist_map_inline:nn {#2} {
530     \exp_args:No \um@addto@mathmap \um@symfont {##1}{#1}{#3}
531   }
532 }

```

`\um_mathmap_parse: Nnn` #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`).

```

533 \cs_set:Nn \um_mathmap_parse:Nnn {
534   \clist_map_inline:Nn \um@char@num@range {
535     \ifnum##1=#3\relax
536       \clist_map_inline:nn {#2} {

```

```

537         \exp_args:No \um@addto@mathmap \um@symfont {###1}{#1}{#3}
538     }
539     \fi
540 }
541 }

```

`\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.

```

542 \newcommand\um@addto@mathmap[4]{
543   \expandafter\g@addto@macro
544   \csname um_setup_\cs_to_str:N #3:\endcsname{
545     \SetMathCode{#2}{\mathalpha}{#1}{#4}
546   }
547 }

```

























## 6.4 (Big) operators

























Turns out that  $\text{\XeTeX}$  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  $\text{\TeX}$  *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	$\sum\limits_{0\atop 1}^1$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod\limits_{0\atop 1}^1$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod\limits_{0\atop 1}^1$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum\limits_{0\atop 1}^1$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int\limits_{0\atop 1}^1$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint\limits_{0\atop 1}^1$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint\limits_{0\atop 1}^1$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR

U+0222E		<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F		<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230		<code>\oiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231		<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232		<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233		<code>\ointctrackwise</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0		<code>\bigwedge</code>	LOGICAL OR OPERATOR
U+022C1		<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2		<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3		<code>\bigcup</code>	UNION OPERATOR
U+027D5		<code>\leftouterjoin</code>	LEFT OUTER JOIN
U+027D6		<code>\rightouterjoin</code>	RIGHT OUTER JOIN
U+027D7		<code>\fullouterjoin</code>	FULL OUTER JOIN
U+027D8		<code>\biguparrow</code>	LARGE UP TACK
U+027D9		<code>\bigdownarrow</code>	LARGE DOWN TACK
U+029F8		<code>\xsolidus</code>	BIG SOLIDUS
U+029F9		<code>\xbsolidus</code>	BIG REVERSE SOLIDUS
U+02A00		<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR
U+02A01		<code>\bigoplus</code>	N-ARY CIRCLED PLUS OPERATOR
U+02A02		<code>\bigotimes</code>	N-ARY CIRCLED TIMES OPERATOR
U+02A03		<code>\bigcupdot</code>	N-ARY UNION OPERATOR WITH DOT
U+02A04		<code>\biguplus</code>	N-ARY UNION OPERATOR WITH PLUS
U+02A05		<code>\bigsqcap</code>	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06		<code>\bigsqcup</code>	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		\iiiint	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 LINE INTEGRATION WITH RECTANGULAR
U+02A12		\rppolint	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A13		\scpolint	PATH AROUND POLE LINE INTEGRATION NOT INCLUDING THE POLE
U+02A14		\npolint	INTEGRAL AROUND A POINT OPERATOR
U+02A15		\pointint	QUATERNION INTEGRAL OPERATOR
U+02A16		\sqint	INTEGRAL WITH LEFTWARDS ARROW WITH HOOK
U+02A17		\intlarhk	INTEGRAL WITH TIMES SIGN
U+02A18		\intx	INTEGRAL WITH INTERSECTION
U+02A19		\intcap	INTEGRAL WITH UNION
U+02A1A		\intcup	INTEGRAL WITH OVERBAR
U+02A1B		\upint	INTEGRAL WITH UNDERBAR
U+02A1C		\lowint	JOIN
U+02A1D		\Join	LARGE LEFT TRIANGLE OPERATOR
U+02A1E		\bigtriangleleft	Z NOTATION SCHEMA COMPOSITION
U+02A1F		\zcmp	Z NOTATION SCHEMA PIPING
U+02A20		\zpipe	Z NOTATION SCHEMA PROJECTION
U+02A21		\zproject	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFC		\biginterleave	N-ARY WHITE VERTICAL BAR
U+02AFF		\bigtalloblong	N-ARY WHITE VERTICAL BAR

\um@nolimits This macro is a sequence containing those maths operators that require a \nolim-

its suffix. This list is used when processing `unicode-math-table.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 18). 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  $\iiint$ , but that might be a matter of preference.

```

548 \def\um@nolimits{
549   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oiint\@elt\oiint
550   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrclockwise\@elt\sumint
551   \@elt\intbar\@elt\intBar\@elt\oint\@elt\circnint\@elt\awint\@elt\rppoint
552   \@elt\scpoint\@elt\ntpoint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
553   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint
554 }

```

`\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`.

```

555 \newcommand\addnolimits[1]{
556   \expandafter\def\expandafter\um@nolimits\expandafter{\um@nolimits\@elt#1}
557 }

```

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

```

558 \def\removenolimits#1{
559   \begingroup
560     \def\@elt##1{
561       \ifx##1#1\else
562         \noexpand\@elt\noexpand##1
563       \fi}
564     \xdef\um@nolimits{\um@nolimits}
565   \endgroup
566 }

```

 $\iiint_V$ 
 $\iiint_V$ 
 $\iiint_V$ 

```

\def\dmath#1{\displaystyle #1$}
\setmathfont{Asana Math} \dmath{\iiint_V}
\removenolimits\iiint
\setmathfont{Asana Math} \dmath{\iiint_V}
\addnolimits\iiint
\setmathfont{Asana Math} \dmath{\iiint_V}

```

## 6.5 Radicals

The radical for square root is organised in `\um@set@mathsymbol` on page ?? . I think it’s the only radical ever. (Actually, there is also `\cuberoot` and `\fourthroot`, but they don’t seem to behave as proper radicals.)

Also, 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.

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

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

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

## 6.6 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>

```
568 \let\left@primitive\left
```

```
569 \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`

$$((((x^1)^2)^3)^4)^5$$

$$[[[[y^1]^2]^3]^4]^5$$

$${{{{z^1}^2}^3}^4}^5$$

```
\setmathfont{Asana Math}
\[ \left(\left(\left(\left(\left( x
\right)^1\right)^2\right)^3\right)^4\right)^5 \]
\[ \left[\left[\left[\left[\left[ y
\right]^1\right]^2\right]^3\right]^4\right]^5 \]
\[ \left\{\left\{\left\{\left\{\left\{ z
\right\}^1\right\}^2\right\}^3\right\}^4\right\}^5 \]
```

Here are all `\mathopen` characters:

USV	Ex.	Macro	Description
U+00028	(	<code>\lparen</code>	LEFT PARENTHESIS

U+0005B	[	\lbrack	LEFT SQUARE BRACKET
U+0007B	{	\lbrace	LEFT CURLY BRACKET
			DOUBLE ANGLE QUOTATION MARK
U+000AB	«	\guillemotleft	(GUILLEMET), LEFT
U+02018	‘	\lq	SINGLE QUOTATION MARK, LEFT
U+0201A	,	\quotsinglbase	RISING SINGLE QUOTE, LEFT (LOW)
U+0201E	„	\quotdblbase	RISING DOUBLE QUOTE, LEFT (LOW)
			SINGLE ANGLE QUOTATION MARK
U+02039	<	\guilsinglleft	(GUILLEMET), LEFT
U+0221A	√	\sqrt	RADICAL
U+0221B	∛	\cuberoot	CUBE ROOT
U+0221C	∜	\fourthroot	FOURTH ROOT
U+02308	⌈	\lceil	LEFT CEILING
U+0230A	⌋	\lfloor	LEFT FLOOR
U+0231C	⌵	\ulcorner	UPPER LEFT CORNER
U+0231E	⌷	\llcorner	LOWER LEFT CORNER
			LIGHT LEFT TORTOISE SHELL BRACKET
U+02772		\lbrbrak	ORNAMENT
U+027C5	⌵	\lbag	LEFT S-SHAPED BAG DELIMITER
U+027CC	)	\longdivision	LONG DIVISION
			MATHEMATICAL LEFT WHITE SQUARE
U+027E6	⌌	\lBrack	BRACKET
U+027E8	⌵	\langle	MATHEMATICAL LEFT ANGLE BRACKET
			MATHEMATICAL LEFT DOUBLE ANGLE
U+027EA	⌌	\lAngle	BRACKET
			MATHEMATICAL LEFT WHITE TORTOISE
U+027EC		\Lbrbrak	SHELL BRACKET
U+02983	⌌	\lBrace	LEFT WHITE CURLY BRACKET
U+02985	⌐	\lParen	LEFT WHITE PARENTHESIS
U+02987	⌐	\llparenthesis	Z NOTATION LEFT IMAGE BRACKET
U+02989	⌐	\llangle	Z NOTATION LEFT BINDING BRACKET
U+0298B	⌌	\lbrackubar	LEFT SQUARE BRACKET WITH UNDERBAR
			LEFT SQUARE BRACKET WITH TICK IN TOP
U+0298D	⌌	\lbrackultick	CORNER
			LEFT SQUARE BRACKET WITH TICK IN
U+0298F	⌌	\lbracklltick	BOTTOM CORNER
U+02991	⌵	\langedot	LEFT ANGLE BRACKET WITH DOT
U+02993	⌵	\lparenless	LEFT ARC LESS-THAN BRACKET
U+02997	⌌	\lblkbrbrak	LEFT BLACK TORTOISE SHELL BRACKET
U+029D8	⌌	\lvzigzag	LEFT WIGGLY FENCE
U+029DA	⌌	\Lvzigzag	LEFT DOUBLE WIGGLY FENCE
U+029FC	⌵	\lcurvyangle	LEFT POINTING CURVED ANGLE BRACKET
U+03014		\lbrbrak	LEFT BROKEN BRACKET
U+03018		\Lbrbrak	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
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)
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
U+02773	⌸	<code>\rbrbrak</code>	ORNAMENT
U+027C6	⌶	<code>\rbag</code>	RIGHT S-SHAPED BAG DELIMITER
U+027E7	⌷	<code>\rBrack</code>	BRACKET
U+027E9	⌹	<code>\rangle</code>	MATHEMATICAL RIGHT ANGLE BRACKET
U+027EB	⌻	<code>\rAngle</code>	BRACKET
U+027ED	⌽	<code>\Rbrbrak</code>	SHELL BRACKET
U+02984	⌢	<code>\rBrace</code>	RIGHT WHITE CURLY BRACKET
U+02986	⌣	<code>\rParen</code>	RIGHT WHITE PARENTHESIS
U+02988	⌤	<code>\rrparenthesis</code>	Z NOTATION RIGHT IMAGE BRACKET
U+0298A	⌥	<code>\rrangle</code>	Z NOTATION RIGHT BINDING BRACKET
U+0298C	⌦	<code>\rbrackubar</code>	RIGHT SQUARE BRACKET WITH UNDERBAR
U+0298E	⌧	<code>\rbracklrtick</code>	BOTTOM CORNER
U+02990	⌨	<code>\rbrackurtick</code>	RIGHT SQUARE BRACKET WITH TICK IN TOP
U+02992	〈	<code>\rangledot</code>	CORNER
U+02994	⌫	<code>\rpangtr</code>	RIGHT ANGLE BRACKET WITH DOT
U+02998	⌭	<code>\rblkrbrak</code>	RIGHT ARC GREATER-THAN BRACKET
U+029D9	⌹	<code>\rvzigzag</code>	RIGHT BLACK TORTOISE SHELL BRACKET
U+029DB	⌻	<code>\Rvzigzag</code>	RIGHT WIGGLY FENCE
U+029FD	⌽	<code>\rcurvyangle</code>	RIGHT DOUBLE WIGGLY FENCE
U+03015	⌵	<code>\rbrbrak</code>	RIGHT POINTING CURVED ANGLE BRACKET
U+03019	⌸	<code>\Rbrbrak</code>	RIGHT BROKEN BRACKET
			RIGHT WHITE TORTOISE SHELL BRACKET

## 6.7 Maths accents

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

USV	Ex.	Macro	Description
U+00300	☞	<code>\grave</code>	GRAVE ACCENT
U+00301	☞	<code>\acute</code>	ACUTE ACCENT
U+00302	☞	<code>\hat</code>	CIRCUMFLEX ACCENT
U+00303	☞	<code>\tilde</code>	TILDE
U+00304	☞	<code>\bar</code>	MACRON
U+00305	☞	<code>\overbar</code>	OVERBAR EMBELLISHMENT
U+00306	☞	<code>\breve</code>	BREVE
U+00307	☞	<code>\dot</code>	DOT ABOVE
U+00308	☞	<code>\ddot</code>	DIERESIS
U+00309	☞	<code>\ovhook</code>	COMBINING HOOK ABOVE
U+0030A	☞	<code>\ocirc</code>	RING
U+0030C	☞	<code>\check</code>	CARON
U+00310	☞	<code>\candra</code>	CANDRABINDU (NON-SPACING)
U+00312	☞	<code>\oturnedcomma</code>	COMBINING TURNED COMMA ABOVE GREEK PSILI (SMOOTH BREATHING)
U+00313	☞	<code>\osmooth</code>	(NON-SPACING) GREEK DASIA (ROUGH BREATHING)
U+00314	☞	<code>\orough</code>	(NON-SPACING)
U+00315	☞	<code>\ocommatoprigh</code>	COMBINING COMMA ABOVE RIGHT
U+0031A	☞	<code>\droang</code>	LEFT ANGLE ABOVE (NON-SPACING)
U+020D0	☞	<code>\leftharpoonaccent</code>	COMBINING LEFT HARPOON ABOVE
U+020D1	☞	<code>\rightharpoonaccent</code>	COMBINING RIGHT HARPOON ABOVE
U+020D2	☞	<code>\vertoverlay</code>	COMBINING LONG VERTICAL LINE OVERLAY
U+020D6	☞	<code>\overleftarrow</code>	COMBINING LEFT ARROW ABOVE
U+020D7	☞	<code>\vec</code>	COMBINING RIGHT ARROW ABOVE
U+020DB	☞	<code>\dddot</code>	COMBINING THREE DOTS ABOVE
U+020DC	☞	<code>\ddddot</code>	COMBINING FOUR DOTS ABOVE
U+020E1	☞	<code>\overleftrighthar</code>	COMBINING LEFT RIGHT ARROW ABOVE
U+020E7	☞	<code>\annuity</code>	COMBINING ANNUITY SYMBOL
U+020E8	☞	<code>\threeunderdot</code>	COMBINING TRIPLE UNDERDOT
U+020E9	☞	<code>\widebridgeabove</code>	COMBINING WIDE BRIDGE ABOVE COMBINING RIGHTWARDS HARPOON WITH
U+020EC	☞	<code>\underrightharpoondown</code>	BARB DOWNWARDS COMBINING LEFTWARDS HARPOON WITH
U+020ED	☞	<code>\underleftharpoondown</code>	BARB DOWNWARDS
U+020EE	☞	<code>\underleftarrow</code>	COMBINING LEFT ARROW BELOW
U+020EF	☞	<code>\underrightarrow</code>	COMBINING RIGHT ARROW BELOW
U+020F0	☞	<code>\asteraccent</code>	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.

```
570 \newcommand\um@zf@feature[2]{
571   \define@key[zf]{options}{#1}[] {
572     \ifum@fontspec@feature
573       #2
574     \else
575       \PackageError{fontspec/unicode-math}
576         {The ‘#1’ font feature can only be used for maths fonts}
577         {The feature you tried to use can only be in commands
578          like \protect\setmathfont}
579     \fi
580   }
581 }
```

### 7.1 OpenType maths font features

```
582 \um@zf@feature{ScriptStyle}{
583   \zf@update@ff{+ssty=0}
584 }
585 \um@zf@feature{ScriptScriptStyle}{
586   \zf@update@ff{+ssty=1}
587 }
```

### 7.2 Script and scriptscript font options

```
588 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
589 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
590 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
591 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

### 7.3 Range processing

The ‘ALL’ branch here is deprecated and happens automatically.

```
592 \define@choicekey+um{options}{Range}[\@tempa\@tempb]{ALL}{
593   \ifcase\@tempb\relax
594     \global\let\um@char@range\empty
595   \fi
596 }{
597   \xdef\um@char@range{#1}
598 }
```

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:

```
599 \newcommand\um@parse@term[4]{
600   \clist_map_variable:NNn \um@char@range \@ii {
601     \unless\ifx\@ii\empty
602       \@tempswafalse
```

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

```
603   \expandafter\um@firstchar\expandafter{\@ii}
604   \ifx\@tempa\um@backslash
605     \expandafter\ifx\@ii#2\relax
606       \@tempswatrue
607   \else
608     \expandafter\ifx\@ii#3\relax
609       \@tempswatrue
610   \fi
611 \fi
```

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

```
612   \else
613     \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
614   \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.

```
615   \if@tempswa
616     \ifx\um@char@num@range\empty
```

```

617         \g@addto@macro\um@char@num@range{#1}
618     \else
619         \g@addto@macro\um@char@num@range{, #1}
620     \fi
621     #4%
622 \fi
623 \fi
624 }
625 }
626 \def\um@firstof#1#2\@nil{#1}
627 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
628 \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`.

```

629 \def\um@parse@range#1-#2-#3\@nil#4\@nil{
630     \def\@tempa{#1}
631     \def\@tempb{#2}

```

---

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

---

```

632     \expandafter\ifx\expandafter\@marker\@tempb\relax
633         \ifnum#4=#1\relax
634             \@tempwatrue
635         \fi
636     \else

```

---

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

---

```

637     \ifx\@empty\@tempb
638         \ifnum#4>\numexpr#1-1\relax
639             \@tempwatrue
640         \fi
641     \else

```

---

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

---

```

642     \ifx\@empty\@tempa
643     \ifnum#4<\numexpr#2+1\relax
644     \@tempswattrue
645     \fi

```

---

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

---

```

646     \else
647     \ifnum#4>\numexpr#1-1\relax
648     \ifnum#4<\numexpr#2+1\relax
649     \@tempswattrue
650     \fi
651     \fi
652     \fi
653     \fi
654     \fi
655 }

```

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

```

656 \newcommand\um@setmathcode[3][1]{
657   \clist_map_variable:nNn {#2} \l_um_input_num {
658     \prg_stepwise_variable:nnnNn{1}{1}{#1} \l_um_incr_num {
659       \SetMathCode
660       {\numexpr \l_um_incr_num+ \l_um_input_num - 1\relax}
661       {\mathalpha}{\um@symfont}
662       {\numexpr \l_um_incr_num + #3 - 1\relax}
663     }
664   }
665 }

```

$\backslash um\_set\_mathalphabet\_char:Nnnn$  #1 : Maths alphabet  
#2 : Input char(s)  
#3 : Output char  
Loops through character ranges setting  $\backslash mathcode$ .

```

666 \cs_set:Npn \exp_args:Nnff {\::n\::f\::f\:::}
667 \cs_new:Nn \um_set_mathalphabet_char:Nnn {
668   \clist_map_variable:nNn {#2} \l_um_input_num {

```

```

669 \exp_args:Nnff \um_mathmap:Nnn {#1}
670 {\number\numexpr\l_um_input_num\relax} {\number\numexpr#3\relax}
671 }
672 }

```

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

```

673 \cs_new:Nn \um_set_mathalph_range:nNnn {
674   \clist_map_variable:nNn {#3} \l_um_input_num {
675     \prg_stepwise_variable:nnnNn {0}{1}{#1} \l_um_inc_num {
676       \exp_args:Nnff \um_mathmap:Nnn {#2}
677       {\number\numexpr \l_um_inc_num + \l_um_input_num \relax}
678       {\number\numexpr \l_um_inc_num + #4 \relax}
679     }
680   }
681 }
682 \cs_new:Nn \um_set_mathalphabet_numbers:Nnn {
683   \um_set_mathalph_range:nNnn {9}{#1}{#2}{#3}
684 }
685 \cs_new:Nn \um_set_mathalphabet_latin:Nnn {
686   \um_set_mathalph_range:nNnn {25}{#1}{#2}{#3}
687 }
688 \cs_new:Nn \um_set_mathalphabet_greek:Nnn {
689   \um_set_mathalph_range:nNnn {24}{#1}{#2}{#3}
690 }

```

---

**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.

```

691 \AtBeginDocument{\um@resolve@greek}
692 \newcommand\um@resolve@greek{
693   \def\Alpha{\mitAlpha}
694   \def\Beta{\mitBeta}
695   \def\Gamma{\mitGamma}
696   \def\Delta{\mitDelta}
697   \def\Epsilon{\mitEpsilon}
698   \def\Zeta{\mitZeta}
699   \def\Eta{\mitEta}

```

```

700 \def\Theta{\mit\Theta}
701 \def\Iota{\mit\Iota}
702 \def\Kappa{\mit\Kappa}
703 \def\Lambda{\mit\Lambda}
704 \def\Mu{\mit\mu}
705 \def\Nu{\mit\Nu}
706 \def\Xi{\mit\Xi}
707 \def\Omicron{\mit\Omicron}
708 \def\Pi{\mit\Pi}
709 \def\Rho{\mit\Rho}
710 \def\varTheta{\mit\varTheta}
711 \def\Sigma{\mit\Sigma}
712 \def\Tau{\mit\Tau}
713 \def\Upsilon{\mit\Upsilon}
714 \def\Phi{\mit\Phi}
715 \def\Chi{\mit\Chi}
716 \def\Psi{\mit\Psi}
717 \def\Omega{\mit\Omega}

```

Lowercase:

```

718 \def\alpha{\mit\alpha}
719 \def\beta{\mit\beta}
720 \def\gamma{\mit\gamma}
721 \def\delta{\mit\delta}
722 \def\epsilon{\mit\epsilon}
723 \def\zeta{\mit\zeta}
724 \def\eta{\mit\eta}
725 \def\theta{\mit\theta}
726 \def\iota{\mit\iota}
727 \def\kappa{\mit\kappa}
728 \def\lambda{\mit\lambda}
729 \def\mu{\mit\mu}
730 \def\nu{\mit\nu}
731 \def\xi{\mit\xi}
732 \def\omicron{\mit\omicron}
733 \def\pi{\mit\pi}
734 \def\rho{\mit\rho}
735 \def\varsigma{\mit\varsigma}
736 \def\sigma{\mit\sigma}
737 \def\tau{\mit\tau}
738 \def\upsilon{\mit\upsilon}
739 \def\phi{\mit\phi}
740 \def\chi{\mit\chi}
741 \def\psi{\mit\psi}
742 \def\omega{\mit\omega}
743 \def\varepsilon{\mit\varepsilon}
744 \def\vartheta{\mit\vartheta}

```



```

745 \def\varkappa{\mitvarkappa}
746 \def\varphi{\mitvarphi}
747 \def\varrho{\mitvarrho}
748 \def\varpi{\mitvarpi}
749 }

```

\um@def@numbers

```

750 \newcommand\um@def@numbers{
751   \um@setmathcode[10]{\um@usv@num}{\um@usv@num}
752 }

```

\um\_setup\_literals: : TODO : other literal symbols

```

753 \cs_set:Nn \um_setup_literals: {
754   \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}
755   \um@setmathcode[26]{\um@usv@itLatin}{\um@usv@itLatin}
756   \um@setmathcode[26]{\um@usv@itlatin}{\um@usv@itlatin}
757   \um@setmathcode{\um@usv@ith}{\um@usv@ith}
758   \um@setmathcode[26]{\um@usv@uplatin}{\um@usv@uplatin}
759   \um@setmathcode[25]{\um@usv@upGreek}{\um@usv@upGreek}
760   \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}
761   \um@setmathcode[25]{\um@usv@itGreek}{\um@usv@itGreek}
762   \um@setmathcode[25]{\um@usv@upgreek}{\um@usv@upgreek}
763 }

```

\um\_setup\_bf\_literals: TODO: other literal symbols

```

764 \cs_set:Nn \um_setup_bf_literals: {
765   \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}
766   \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}
767   \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}
768   \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}
769   \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}
770   \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}
771   \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}
772   \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}
773 }

```

\um@def@upLatin

```

774 \newcommand\um@def@upLatin{
775   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
776 }

```

\um@def@itLatin

```

777 \newcommand\um@def@itLatin{
778   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
779 }

```

`\um@def@itlatin` Don't overlook 'h', which maps to U+210E: PLANCK CONSTANT instead of the expected U+1D455: MATHEMATICAL ITALIC SMALL H.

```

780 \newcommand\um@def@itlatin{
781   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
782   \um@setmathcode{\`h,\um@usv@ith}{\um@usv@ith}
783 }

```

`\um@def@uplatin`

```

784 \newcommand\um@def@uplatin{
785   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
786   \um@setmathcode{\um@usv@ith}{\`h}
787 }

```

`\um@def@upGreek`

```

788 \newcommand\um@def@upGreek{
789   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
790   \um@setmathcode{\um@usv@varTheta,"1D6F3}{\um@usv@varTheta}
791 }

```

`\um@def@itGreek`

```

792 \newcommand\um@def@itGreek{
793   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
794   \um@setmathcode{\um@usv@varTheta}{\um@usv@itvarTheta}
795 }

```

`\um@def@upgreek`

```

796 \newcommand\um@def@upgreek{
797   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
798   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
799   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
800   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
801   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
802   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
803   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
804 }

```

`\um@def@itgreek`

```

805 \newcommand\um@def@itgreek{
806   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
807   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
808   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
809   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
810   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
811   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
812   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
813 }

```

## 8 Maths alphabets mapping definitions

Algorithm for setting alphabet fonts:

- By default, try and set all of them.
- Check for the first glyph of each to detect if the font supports each alphabet. (This doesn't work to distinguish Latin/Greek but we hope all maths fonts will have at least them!)
- For alphabets that are not supported, *do nothing*. (This includes leaving the old alphabet definition in place.)
- For alphabets that do exist, overwrite whatever's already there.

```

814 \cs_new:Nn \um_setup_math_alphabet:n {
815   \um_glyph_if_exist:nTF {\csname um@usv@#1latin \endcsname}{
816     \um_maybe_init_alphabet:n {#1}
817     \um_prepare_alph:n {#1}
818     \use:c {um_config_math#1:}
819   }{
820     \um@PackageWarning{Math~ alphabet~ "#1"~ not~ found~ with~ this~ font}
821   }
822 }

823 \cs_set:Nn \um_init_alphabet:n {
824   \cs_set_eq:cN {um_setup_math#1:} \prg_do_nothing:
825 }
```

`\um_glyph_if_exist:nTF` : TODO: Generalise for arbitrary fonts! `\um@font` is not always the one used for a specific glyph!!

```

826 \prg_new_conditional:Nnn \um_glyph_if_exist:n {p,TF,T,F} {
827   \etex_iffontchar:D \um@font #1 \scan_stop: \prg_return_true: \else: \prg_return_false: \fi:
828 }
```

`\um_prepare_alph:n` If `\mathXY` hasn't been (re-)declared yet, then define it in terms of unicode-math definitions.

```

829 \cs_new:Nn \um_prepare_alph:n {
830   \cs_if_exist:cF {um_math#1:n} {
831     \cs_set:cpn {um_math#1:n} ##1 {
832       \begingroup \use:c {um_setup_math#1:} ##1 \endgroup
833     }
834     \cs_set_protected:cpn {math#1} {
835       \mode_if_math:F {
836         \expandafter\non@alpherr\expandafter{\csname math#1\endcsname\space}
837       }
838       \use:c {um_math#1:n}
839     }
840   }
```

```

840 }
841 }

842 \cs_new:Nn \um_setup_alphabets: {
843   \um_setup_math_alphabet: n {up }
844   \um_setup_math_alphabet: n {it }
845   \um_setup_math_alphabet: n {bb }
846   \um_setup_math_alphabet: n {scr }
847   \um_setup_math_alphabet: n {frak }
848   \um_setup_math_alphabet: n {sf }
849   \um_setup_math_alphabet: n {sfit }
850   \um_setup_math_alphabet: n {tt }
851   \um_setup_math_alphabet: n {bf }
852   \um_setup_math_alphabet: n {bfup }
853   \um_setup_math_alphabet: n {bfit }
854   \um_setup_math_alphabet: n {bfscr }
855   \um_setup_math_alphabet: n {bffrak }
856   \um_setup_math_alphabet: n {bfsf }
857   \um_setup_math_alphabet: n {bfsfup }
858   \um_setup_math_alphabet: n {bfsfit }
859 }

```

: TODO : nested alphabets?

### 8.0.1 Upright: `\mathup`

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

□

αβγδεζηθικλμνξοπρστυφχψω

□ϑκρρω

```

 $\mathup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$  \\
 $\mathup{abcdefghijklmnopqrstuvwxyz}$  \\
 $\mathup{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$  \\quad  $\mathup{□}$  \\
 $\mathup{αβγδεζηθικλμνξοπρστυφχψω}$  \\quad  $\mathup{□ϑκρρω}$  \\

```

Takes both upright and italic characters to be typeset as upright symbols.

```

860 \cs_new:Npn \um_config_mathup: {
861   \um_set_mathalphabet_latin: Nnn{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
862   \um_set_mathalphabet_latin: Nnn{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
863   \um_set_mathalphabet_greek: Nnn{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
864   \um_set_mathalphabet_greek: Nnn{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
865   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}
866   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}
867   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}
868   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
869   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
870   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}

```

```

871 \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@varphi}
872 \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@varrho}
873 \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@varpi}
874 }

```

## 8.0.2 Italic: `\mathit`

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ

Θ

αβγδεζηθικλμνξοπρστυφχψω

εθζφρω

```

$\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathit{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathit{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}$ \\quad $\mathit{Θ}$ \\
$\mathit{αβγδεζηθικλμνξοπρστυφχψω}$ \\quad $\mathit{εθζφρω}$ \\

```

Roman:

```

875 \cs_new:Npn \um_config_mathit: {
876 \um_set_mathalphabet_latin: Nnn{\mathit}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@itLatin}
877 \um_set_mathalphabet_latin: Nnn{\mathit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@itlatin}
878 \um_set_mathalphabet_char: Nnn{\mathit}{`h, \um@usv@ith}{\um@usv@ith}

```

Greek:

```

879 \um_set_mathalphabet_greek: Nnn{\mathit}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@itGreek}
880 \um_set_mathalphabet_greek: Nnn{\mathit}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@itgreek}
881 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@itNabla}
882 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@partial, \um@usv@itpartial}{\um@usv@itpartial}
883 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@itvarTheta}
884 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@itvarepsilon}
885 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@itvartheta}
886 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@itvarkappa}
887 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@itvarphi}
888 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@itvarrho}
889 \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@itvarpi}
890 }

```

## 8.0.3 Blackboard or double-struck: `\mathbb`

0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

```

$\mathbb{0123456789}$ \\
$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbb{abcdefghijklmnopqrstuvwxyz}$ \\

```

Numbers:

```

891 \cs_new:Npn \um_config_mathbb: {
892 \um_set_mathalphabet_numbers: Nnn{\mathbb}{\um@usv@num}{\um@usv@bbnum}

```

Roman uppercase:

```

893 \um_set_mathalphabet_latin: Nnn{\mathbb}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bbLatin}
894 \um_set_mathalphabet_char: Nnn{\mathbb}{`\C, "1D60A}{ "2102}
895 \um_set_mathalphabet_char: Nnn{\mathbb}{`\H, "1D60F}{ "210D}
896 \um_set_mathalphabet_char: Nnn{\mathbb}{`\N, "1D60F}{ "2115}
897 \um_set_mathalphabet_char: Nnn{\mathbb}{`\P, "1D617}{ "2119}
898 \um_set_mathalphabet_char: Nnn{\mathbb}{`\Q, "1D618}{ "211A}
899 \um_set_mathalphabet_char: Nnn{\mathbb}{`\R, "1D619}{ "211D}
900 \um_set_mathalphabet_char: Nnn{\mathbb}{`\Z, "1D621}{ "2124}

```

Roman lowercase:

```

901 \um_set_mathalphabet_latin: Nnn{\mathbb}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bblatin}
902 }

```

#### 8.0.4 Script or caligraphic: `\mathscr` and `\mathcal`

---

<i>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</i>	<code>\$\mathscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
<i>a b c d e f g h i j k l m n o p q r s t u v w x y z</i>	<code>\$\mathscr{abcdefghijklmnopqrstuvwxyz}\$ \</code>

---

```

903 \cs_new:Npn \um_config_mathscr: {
904 \um_set_mathalphabet_latin: Nnn{\mathscr}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@scrLatin}
905 \um_set_mathalphabet_char: Nnn{\mathscr}{`\B, "1D435}{ "212C}
906 \um_set_mathalphabet_char: Nnn{\mathscr}{`\E, "1D438}{ "2130}
907 \um_set_mathalphabet_char: Nnn{\mathscr}{`\F, "1D439}{ "2131}
908 \um_set_mathalphabet_char: Nnn{\mathscr}{`\H, "1D43B}{ "210B}
909 \um_set_mathalphabet_char: Nnn{\mathscr}{`\I, "1D43C}{ "2110}
910 \um_set_mathalphabet_char: Nnn{\mathscr}{`\L, "1D43F}{ "2112}
911 \um_set_mathalphabet_char: Nnn{\mathscr}{`\M, "1D440}{ "2133}
912 \um_set_mathalphabet_char: Nnn{\mathscr}{`\R, "1D445}{ "211B}
913 \um_set_mathalphabet_latin: Nnn{\mathscr}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@scrlatin}
914 \um_set_mathalphabet_char: Nnn{\mathscr}{`\e, "1D452}{ "212F}
915 \um_set_mathalphabet_char: Nnn{\mathscr}{`\g, "1D454}{ "210A}
916 \um_set_mathalphabet_char: Nnn{\mathscr}{`\o, "1D45C}{ "2134}
917 }

```

#### 8.0.5 Fraktur or fraktur or blackletter: `\mathfrak`

---

<i>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</i>	<code>\$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
<i>a b c d e f g h i j k l m n o p q r s t u v w x y z</i>	<code>\$\mathfrak{abcdefghijklmnopqrstuvwxyz}\$ \</code>

---

Letters, with exceptions `{\C, \S, \J, \R, \O}`:

```

918 \cs_new:Npn \um_config_mathfrak: {

```

```

919 \um_set_mathalphabet_latin: Nnn{\mathfrak}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@frakLat
920 \um_set_mathalphabet_char: Nnn{\mathfrak}{`\C, "1D436}{ "212D}
921 \um_set_mathalphabet_char: Nnn{\mathfrak}{`\H, "1D43B}{ "210C}
922 \um_set_mathalphabet_char: Nnn{\mathfrak}{`\I, "1D43C}{ "2111}
923 \um_set_mathalphabet_char: Nnn{\mathfrak}{`\R, "1D445}{ "211C}
924 \um_set_mathalphabet_char: Nnn{\mathfrak}{`\Z, "1D44D}{ "2128}
925 \um_set_mathalphabet_latin: Nnn{\mathfrak}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@fraklat
926 }

```

## 8.0.6 Sans serif: \mathsf

---

0123456789	$\mathsf{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathsf{abcdefghijklmnopqrstuvwxyz}$

---

```

927 \cs_new:Npn \um_config_mathsf: {
928   \um_set_mathalphabet_numbers: Nnn{\mathsf}{\um@usv@num}{\um@usv@s fnum}
929   \um_set_mathalphabet_latin: Nnn{\mathsf}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@s flatin}
930   \um_set_mathalphabet_latin: Nnn{\mathsf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@s flatin}
931 }

```

## 8.0.7 Sans serif italic: \mathsfit

---

0123456789	$\mathsfit{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathsfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathsfit{abcdefghijklmnopqrstuvwxyz}$

---

```

932 \cs_new:Npn \um_config_mathsfitt: {
933   \um_set_mathalphabet_numbers: Nnn{\mathsfit}{\um@usv@num}{\um@usv@s fnum}
934   \um_set_mathalphabet_latin: Nnn{\mathsfit}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@s fitLatin}
935   \um_set_mathalphabet_latin: Nnn{\mathsfit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@s fitlatin}
936 }

```

## 8.0.8 Typewriter or monospaced: \mathtt

---

0123456789	$\mathtt{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathtt{abcdefghijklmnopqrstuvwxyz}$

---

```

937 \cs_new:Npn \um_config_mathtt: {
938   \um_set_mathalphabet_numbers: Nnn{\mathtt}{\um@usv@num}{\um@usv@ttnum}

```

```

939 \um_set_mathalphabet_latin:Nnn{\mathhtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}
940 \um_set_mathalphabet_latin:Nnn{\mathhtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}
941 }

```

## 8.1 Bold alphabets' character mappings

### 8.1.1 Bold: `\mathbf`

0123456789	
ABCDEFGHIJKLMNOPQRSTUVWXYZ	
abcdefghijklmnopqrstuvwxyz	$\mathbb{Z}$
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ	$\mathbb{A}$
Θ	$\mathbb{C}$
αβγδεζηθικλμνξοπρστυφχψω	$\mathbb{A}$
εθκφρω	$\mathbb{A}$

```

942 \cs_new:Npn \um_config_mathbf: {
943   \um_set_mathalphabet_numbers: Nnn{\mathbf f}{\um@usv@num}{\um@usv@bnum}
944   \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@Digamma}" 1D7CA}
945   \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@digamma}" 1D7CB}
946   \if@um@bfliteral
947     \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@upLatin}{\um@usv@bflatin}
948     \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@iTLatin}{\um@usv@bfiTLatin}
949     \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@uplatin}{\um@usv@bflatin}
950     \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@itlatin}{\um@usv@bfitlatin}
951     \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upGreek}{\um@usv@bfGreek}
952     \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@iTGreek}{\um@usv@bfITGreek}
953     \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upgreek}{\um@usv@bfgreek}
954     \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@itgreek}{\um@usv@bfitgreek}
955     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@ith}{\um@usv@bfith}
956     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varTheta}{\um@usv@bvarTheta}
957     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@Nabla}{\um@usv@bNabla}
958     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@Digamma}{\um@usv@bDigamma}
959     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@partial}{\um@usv@bfpartial}
960     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}
961     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@vartheta}{\um@usv@bvartheta}
962     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varkappa}{\um@usv@bfvarkappa}
963     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varphi}{\um@usv@bfvarphi}
964     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varrho}{\um@usv@bfvarrho}
965     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varpi}{\um@usv@bfvarpi}
966     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@digamma}{\um@usv@bDigamma}
967     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
968     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@iT Nabla}{\um@usv@bfiTNabla}
969     \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itpartial}{\um@usv@bfitpartial}

```



```

970 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
971 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvartheta}{\um@usv@bfitvartheta}
972 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
973 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarphi}{\um@usv@bfitvarphi}
974 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarrho}{\um@usv@bfitvarrho}
975 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@itvarpi}{\um@usv@bfitvarpi}
976 \else
977 \if@um@b fupLatin
978 \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfitLatin}
979 \else
980 \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfitLatin}
981 \fi
982 \if@um@b fuplatin
983 \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfitlatin}
984 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@ith}{\um@usv@bfuph}
985 \else
986 \um_set_mathalphabet_latin: Nnn{\mathbf f}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfitlatin}
987 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@ith}{\um@usv@bfith}
988 \fi
989 \if@um@b fupGreek
990 \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfGreek}
991 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@bfvarTheta}
992 \else
993 \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfitGreek}
994 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@bfitvarTheta}
995 \fi
996 \if@um@b fupgreek
997 \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfGreek}
998 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
999 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@bfvartheta}
1000 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@bfvarkappa}
1001 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@bfvarphi}
1002 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@bfvarrho}
1003 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@bfvarpi}
1004 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfpartial}
1005 \else
1006 \um_set_mathalphabet_greek: Nnn{\mathbf f}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfitgreek}
1007 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
1008 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@bfvartheta}
1009 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@bfvarkappa}
1010 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@bfvarphi}
1011 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@bfvarrho}
1012 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@bfvarpi}
1013 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfpartial}
1014 \fi
1015 \um_set_mathalphabet_char: Nnn{\mathbf f}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@bfNabla}

```

```

1016 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um_bfpartial_u
1017 \fi
1018 }

```

### 8.1.2 Bold Italic: \mathbfit

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

```

1019 \cs_new:Npn \um_config_mathbfit: {
1020   \um_set_mathalphabet_numbers: Nnn{\mathbf}{\um@usv@num}{\um@usv@bnum}
1021   \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
1022   \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
1023   \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}
1024   \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}
1025   \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@bflatin}{\um@usv@bfitlatin}
1026   \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@bflatin}{\um@usv@bfitlatin}
1027   \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@bfgreek}{\um@usv@bfitgreek}
1028   \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@bfgreek}{\um@usv@bfitgreek}
1029   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
1030   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfitNabla}
1031   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpartial}
1032   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
1033   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitvartheta}
1034   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
1035   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi}
1036   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}
1037   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
1038 }

```

### 8.1.3 Bold Italic: \mathbfup

<p style="text-align: center;">0123456789</p> <p style="text-align: center;">ABCDEFGHIJKLMNOPQRSTUVWXYZ</p> <p style="text-align: center;">abcdefghijklmnopqrstuvwxyz</p> <p style="text-align: center;">ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ</p> <p style="text-align: center;">αβγδεζηθικλμνξοπρστυφχψω εθκφρω</p>	<pre> \$ \mathbfup{0123456789}\$ \ \$ \mathbfup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \ \$ \mathbfup{abcdefghijklmnopqrstuvwxyz}\$ \ \$ \mathbfup{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$ \quad \$ \mathbfup{Θ}\$ \ \$ \mathbfup{αβγδεζηθικλμνξοπρστυφχψω}\$ \quad \$ \mathbfup{εθκφρω}\$ \ </pre>
--	--

```

1039 \cs_new:Npn \um_config_mathbfup: {

```

```

1040 \um_set_mathalphabet_numbers: Nnn{\mathbfup}{\um@usv@num}{\um@usv@bfnun}
1041 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bflatin}
1042 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bflatin}
1043 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfgreek}
1044 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfgreek}
1045 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@bflatin}{\um@usv@bflatin}
1046 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@bflatin}{\um@usv@bflatin}
1047 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@bfgreek}{\um@usv@bfgreek}
1048 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@bfgreek}{\um@usv@bfgreek}
1049 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@bfva}
1050 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@bfNabla}
1051 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfpart}
1052 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@}
1053 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@bfva}
1054 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@bfva}
1055 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@bfvarphi}
1056 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@bfvarrho}
1057 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@bfvarpi}
1058 }

```

#### 8.1.4 Bold fractur or fraktur or blackletter: `\mathbffrak`

***ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz***

`$\mathbffrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\  
$\mathbffrak{abcdefghijklmnopqrstuvwxyz}$ \\`

```

1059 \cs_new:Npn \um_config_mathbffrak: {
1060   \um_set_mathalphabet_numbers: Nnn{\mathbffrak}{\um@usv@num}{\um@usv@bfnun}
1061   \um_set_mathalphabet_latin: Nnn{\mathbffrak}{\um@usv@upLatin, \um@usv@itLatin, \um@usv@frakLa}
1062   \um_set_mathalphabet_latin: Nnn{\mathbffrak}{\um@usv@uplatin, \um@usv@itlatin, \um@usv@frakla}
1063 }

```

#### 8.1.5 Bold script or calligraphic: `\mathbfscr`

***ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz***

`$\mathbfscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\  
$\mathbfscr{abcdefghijklmnopqrstuvwxyz}$ \\`

```

1064 \cs_new:Npn \um_config_mathbfscr: {
1065   \um_set_mathalphabet_numbers: Nnn{\mathbfscr}{\um@usv@num}{\um@usv@bfnun}
1066   \um_set_mathalphabet_latin: Nnn{\mathbfscr}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfscrL}
1067   \um_set_mathalphabet_latin: Nnn{\mathbfscr}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfscr1}
1068 }

```

### 8.1.6 Bold sans serif: `\mathbf{f}`

0123456789  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ  
 αβγδεζηθικλμνξοπρστυφχψω εδκφρτ

```
\setmathfont{STIXGeneral-Bold}
$\mathbf{f}{0123456789}$ \\\
$\mathbf{f}{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\\
$\mathbf{f}{abcdefghijklmnopqrstuvwxyz}$ \\\
$\mathbf{f}{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad
$\mathbf{f}{Θ}$ \\\
$\mathbf{f}{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbf{f}{εδκφρτ}$ \\\
```

: TODO : These should be contextual!  
 Numbers (always upright) and letters:

```
1069 \cs_new:Npn \um_config_mathbf f: {
1070   \um_set_mathalphabet_numbers:Nnn{\mathbf f}{\um@sv@num}{\um@sv@b fnum}
1071   \um_set_mathalphabet_latin:Nnn{\mathbf f}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@b f flat
1072   \um_set_mathalphabet_latin:Nnn{\mathbf f}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@b f flat
1073   \um_set_mathalphabet_greek:Nnn{\mathbf f}{\um@sv@upGreek,\um@sv@i tGreek}{\um@sv@b f gree
1074   \um_set_mathalphabet_greek:Nnn{\mathbf f}{\um@sv@upGreek,\um@sv@i tGreek}{\um@sv@b f gree
```

Others:

```
1075   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varTheta,\um@sv@i tvarTheta}{1D767}
1076   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@Nabla,\um@sv@i tNabla}{1D76F}
1077   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@partial,\um@sv@i tpartial}{1D789}
1078   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varepsilon,\um@sv@i tvarepsilon}{1D78A}
1079   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@vartheta,\um@sv@i tvartheta}{1D78B}
1080   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varkappa,\um@sv@i tvarkappa}{1D78C}
1081   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varphi,\um@sv@i tvarphi}{1D78D}
1082   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varrho,\um@sv@i tvarrho}{1D78E}
1083   \um_set_mathalphabet_char:Nnn{\mathbf f}{\um@sv@varpi,\um@sv@i tvarpi}{1D78F}
1084 }
```

### 8.1.7 Bold upright sans serif: `\mathbf{fup}`

0123456789  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ  
 αβγδεζηθικλμνξοπρστυφχψω εδκφρτ

```
\setmathfont{STIXGeneral-Bold}
$\mathbf{fup}{0123456789}$ \\\
$\mathbf{fup}{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\\
$\mathbf{fup}{abcdefghijklmnopqrstuvwxyz}$ \\\
$\mathbf{fup}{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad
$\mathbf{fup}{Θ}$ \\\
$\mathbf{fup}{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbf{fup}{εδκφρτ}$ \\\
```

Numbers (always upright) and letters:

```
1085 \cs_new:Npn \um_config_mathbf fup: {
1086   \um_set_mathalphabet_numbers:Nnn{\mathbf fup}{\um@sv@num}{\um@sv@b fnum}
1087   \um_set_mathalphabet_latin:Nnn{\mathbf fup}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@b f flat
1088   \um_set_mathalphabet_latin:Nnn{\mathbf fup}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@b f flat
```

```

1089 \um_set_mathalphabet_greek: Nnn{\mathbfs fup}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfs fg
1090 \um_set_mathalphabet_greek: Nnn{\mathbfs fup}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfs fg

```

Others:

```

1091 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varTheta, \um@usv@itvarTheta}{1D767}
1092 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@Nabla, \um@usv@itNabla}{1D76F}
1093 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@partial, \um@usv@itpartial}{1D789}
1094 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varepsilon, \um@usv@itvarepsilon}{1D78A}
1095 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@vartheta, \um@usv@itvartheta}{1D78B}
1096 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varkappa, \um@usv@itvarkappa}{1D78C}
1097 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varphi, \um@usv@itvarphi}{1D78D}
1098 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varrho, \um@usv@itvarrho}{1D78E}
1099 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varpi, \um@usv@itvarpi}{1D78F}
1100 }

```

### 8.1.8 Bold italic sans serif: `\mathbfs fit`

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ**  
**αβγδεζηθικλμνξοπρστυφχψω εδκφρϖ**

```

\setmathfont{STIXGeneral-BoldItalic}
$\mathbfs fit{0123456789}$ \
$\mathbfs fit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \
$\mathbfs fit{abcdefghijklmnopqrstuvwxyz}$ \
$\mathbfs fit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad
$\mathbfs fit{ε}$ \
$\mathbfs fit{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbfs fit{εδκφρϖ}$ \

```

```

1101 \cs_new:Npn \um_config_mathbfs fit: {
1102 \um_set_mathalphabet_numbers: Nnn{\mathbfs fit}{\um@usv@num}{\um@usv@bfnun}
1103 \um_set_mathalphabet_latin: Nnn{\mathbfs fit}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfs fi
1104 \um_set_mathalphabet_latin: Nnn{\mathbfs fit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfs fi
1105 \um_set_mathalphabet_greek: Nnn{\mathbfs fit}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfs fi
1106 \um_set_mathalphabet_greek: Nnn{\mathbfs fit}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfs fi

```

Other symbols:

```

1107 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varTheta}{1D7A1}
1108 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@bfs fitNa
1109 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfs fi
1110 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varepsilon, \um@usv@itvarepsilon}{1D7C4}
1111 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@vartheta, \um@usv@itvartheta}{1D7C5}
1112 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varkappa, \um@usv@itvarkappa}{1D7C6}
1113 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varphi, \um@usv@itvarphi}{1D7C7}
1114 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varrho, \um@usv@itvarrho}{1D7C8}
1115 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varpi, \um@usv@itvarpi}{1D7C9}
1116 }

```

## 8.2 Definitions of the math symbols

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

`\um@scancharlet` We need to do some trickery to transform the `\UnicodeMathSymbol` argument  
`\um@scanactivedef` "ABCDEF into the  $\text{\TeX}$  ‘caret input’ form `^^^^^abcdef`. It is *very important* that the argument has five characters. Otherwise we need to change the number of ^ chars.

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.

```
1117 \begingroup
1118   \char_make_other:N \^
1119   \gdef\um@scancharlet#1="#2\@nil{
1120     \lowercase{
1121       \scantokens{\global\let#1=^^^^^#2}
1122     }
1123   }
1124   \gdef\um@scanactivedef"#1\@nil#2{
1125     \lowercase{
1126       \scantokens{\global\def^^^^^#1{#2}}
1127     }
1128   }
1129 \endgroup
1130 \let\unicodemathgobble\@gobble
```

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

```
1131 \begingroup
1132   \def\UnicodeMathSymbol#1#2#3#4{
1133     \um@scancharlet#2=#1\@nil
1134   }
1135   \@input{unicode-math-table.tex}
1136 \endgroup
```

## 8.3 Epilogue

Lots of little things to tidy up.

### 8.3.1 Unicode radicals

Undo the damage made to `\sqrt`:

```
1137 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}
```

### 8.3.2 Primes

$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$	$[\$x\prime] \ [\$x\prime\prime] \ [\$x\prime\prime\prime]$
$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$	$[\$x\prime\prime\prime] \ [\$x\prime\prime\prime\prime] \ [\$x\prime\prime\prime\prime\prime]$ $\backslash\sim$
$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$	$[\$x\prime\prime\prime\prime] \ [\$x\prime\prime\prime\prime\prime] \ [\$x\prime\prime\prime\prime\prime\prime]$

We need a new ‘prime’ algorithm. Unicode math has four pre-drawn prime glyphs.

U+2032: PRIME (\primesingle):  $x$   
 U+2033: DOUBLE PRIME (\primedouble):  $x$   
 U+2034: TRIPLE PRIME (\primetripel):  $x$   
 U+2057: QUADRUPLE PRIME (\primequadruple):  $x$

As you can see, they're all drawn at the correct height without being superscripted. However, in a correctly behaviour OpenType font with the MATH table, we also see different behaviour after the `ss ty` feature is applied:

U+2032: PRIME in the 'scriptstyle' font: *x'*

The shrinking and offsetting is done as it is turned into a superscript. This means, luckily, that by default things work nicely for single primes. We can write `x\prime` or `x^\prime` and get:  $x$  and  $x$ . To support single primes, then, things are easier than in  $\text{\LaTeX}$ ; we can just map `'` to `\prime` and not worry about it.

However, it would be nice to use the pre-composed primes above if they exist in the font; consider  $\mathfrak{x}$  vs.  $\mathbf{x}$ . Our algorithm is

- Prime encountered; pcount=1.
- Scan ahead; if prime: pcount:=pcount+1; repeat.
- If not prime, stop scanning.
- If pcount=1, \prime, end.
- If pcount=2, check \primedouble; if it exists, use it, end; if not, goto last step.
- Ditto pcount=3 & \primetriple.
- Ditto pcount=4 & \primequadruple.
- If pcount>4 or the glyph doesn't exist, insert pcount \primes with \primek-ern between each.

```

1138 \muskip_new: N \g_um_primekern_muskip
1139 \muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }% arbitrary
1140 \num_new: N \l_um_primecount_num

```

```

1141 \cs_new:Nn \um_nprimes:n {
1142   \primesingle
1143   \prg_replicate:nn {#1-1} { \mskip \g_um_primekern_muskip \primesingle }
1144 }
1145 \cs_new:Nn \um_nprimes_select:n {
1146   \prg_case_int:nnn {#1}{
1147     {1} { \primesingle }
1148     {2} {
1149       \um_glyph_if_exist:NTF {"2033} {\primedouble} {\um_nprimes:n {#1}}
1150     }
1151     {3} {
1152       \um_glyph_if_exist:NTF {"2034} {\primetriple} {\um_nprimes:n {#1}}
1153     }
1154     {4} {
1155       \um_glyph_if_exist:NTF {"2057} {\primequadruple} {\um_nprimes:n {#1}}
1156     }
1157   }{
1158     \um_nprimes:n {#1}
1159   }
1160 }

```

Scanning is more annoying than you'd think because we want to support all three of `\prime`, `'`, and the unicode prime. And `\ifx` doesn't work with mathactive chars.

Insert a `\bgroup...\egroup` wrapper so that superscript primes work, but does this break spacing for the rest of the time?

```

1161 \cs_new:Nn \um_scanprime: {
1162   \bgroup
1163   \num_zero:N \l_um_primecount_num
1164   \um_scanprime_collect:
1165 }
1166 \cs_new:Nn \um_scanprime_collect: {
1167   \num_incr:N \l_um_primecount_num
1168   \peek_charcode_remove:NTF ' ' {
1169     \um_scanprime_collect:
1170   }{
1171     \peek_meaning_remove:NTF \um_scanprime: {
1172       \um_scanprime_collect:
1173     }{
1174       \peek_charcode_remove:NTF ^^^^2032 {
1175         \um_scanprime_collect:
1176       }{
1177         \um_nprimes_select:n {\l_um_primecount_num}
1178         \egroup
1179       }
1180     }
1181   }

```



```

1182 }
1183 \cs_set_eq:NN \prime \um_scanprime:
1184 \group_begin:
1185   \char_make_active:N \'
1186   \char_make_active:n {"2032}
1187   \cs_gset_eq:NN ' \um_scanprime:
1188   \cs_gset_eq:NN ^^^^2032 \um_scanprime:
1189 \group_end:

```

### 8.3.3 Radicals

`\r@@t` #1 : A mathstyle (for `\mathpalette`)  
 #2 : Leading superscript for the sqrt sign  
 A re-implementation of L<sup>A</sup>T<sub>E</sub>X's hard-coded n-root sign using the appropriate `\fontdimens`.

```

1190 \def\r@@t#1#2{
1191   \setbox\z@\hbox{$\m@th #1\sqrtsign{#2}$}
1192   \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}
1193   \raise \dimexpr(
1194     \um@fontdimen@percent{65}{\um@font}\ht\z@-
1195     \um@fontdimen@percent{65}{\um@font}\dp\z@
1196   )\relax
1197   \copy \rootbox
1198   \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}
1199   \box \z@
1200 }

```

### 8.3.4 Synonyms and all the rest

We need to change L<sup>A</sup>T<sub>E</sub>X's idea of the font used to typeset things like `\sin` and `\cos`:

```

1201 \def\operator@font{\um_setup_mathup:}
1202 \def\to{\rightarrow}
1203 \def\le{\leq}
1204 \def\ge{\geq}

```

`\mathcal`

```

1205 \def\mathcal{\mathscr}

```

`\mathrm`

```

1206 \def\mathrm{\mathup}

```

Overriding amsmath definitions:

```

1207 \AtBeginDocument{
1208   \def\@cdots{\mathinner{\cdots}}
1209 }

```

Interaction with beamer:

```
1210 \AtBeginDocument{
1211   \@ifpackageloaded{beamer}{
1212     \ifbeamer@suppressreplacements\else
1213       \PackageWarningNoLine{unicode-math}{
1214         Disabling~ beamer's~ math~ setup.^^}
1215       Please~ load~ beamer~ with~ the~ [professionalfonts]~ class~ option
1216     }
1217     \beamer@suppressreplacementstrue
1218   \fi
1219 }{}
1220 }
    The end.
1221 \ExplSyntaxOff
```

## File II

# STIX table data extraction

The source for the  $\TeX$  names for the very large number of mathematical glyphs are provided via Barbara Beeton's table file for the STIX project ([ams.org/STIX](http://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.

This table is converted into a form suitable for reading by  $\XeTeX$ , and then hand-edited by the author; the result is `unicode-math-table.tex`.

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  $\text{\XeTeX}$  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, *NFSS decl.* stands for something like `{T1}{lmr}{m}{n}`.

**Maths symbol fonts** Fonts for symbols: , ,

`\DeclareSymbolFont{ name } NFSS decl.`

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

**Maths alphabet fonts** Fonts for  $ABC-xyz$ ,  $\mathfrak{ABC}-\mathcal{XYZ}$ , etc.

`\DeclareMathAlphabet{cmd}` *NFSS decl.*

For commands such as `\mathbf{f}`, 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  $\XTeX$ . The syntax used in  $\LaTeX$ ’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  $\XTeX$ .

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 III

# X<sub>Y</sub>TeX math font dimensions

These are the extended `\fontdimens` available for suitable fonts in X<sub>Y</sub>TeX. Note that LuaTeX takes an alternative route, and this package will eventually provide a wrapper interface to the two (I hope).

<code>\fontdimen</code>	Dimension name	Description
10	<code>SCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 1. Suggested value: 80%.
11	<code>SCRIPTSCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 2 (ScriptScript). Suggested value: 60%.
12	<code>DELIMITEDSUBFORMULAMINHEIGHT</code>	Minimum height required for a delimited expression to be treated as a subformula. Suggested value: normal line height $\times$ 1.5.
13	<code>DISPLAYOPERATORMINHEIGHT</code>	Minimum height of n-ary operators (such as integral and summation) for formulas in display mode.
14	<code>MATHLEADING</code>	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 ( <code>os2.sTypoAscender</code> + <code>os2.sTypoLineGap</code> – <code>MathLeading</code> ) or with ink going below <code>os2.sTypoDescender</code> will result in increasing line height.
15	<code>AXISHEIGHT</code>	Axis height of the font.
16	<code>ACCENTBASEHEIGHT</code>	Maximum (ink) height of accent base that does not require raising the accents. Suggested: x-height of the font ( <code>os2.sxHeight</code> ) plus any possible overshots.

\fontdimen	Dimension name	Description
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	SUBSCRIPTBASELINEDROPMIN	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	SUPERSCRIPSHIFTUPCRAMPED	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.
26	SUPERSCRIPBOTTOMMAX-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: /5 x-height.

\fontdimen	Dimension name	Description
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	UPPERLIMITBASELINERISEMIN	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	LOWERLIMITBASELINEDROP-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	STACKTOPDISPLAYSTYLESHIFT-UP	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	STACKBOTTOMDISPLAYSTYLE-SHIFTDOWN	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×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×default rule thickness.
38	STRETCHSTACKTOPSHIFTUP	Standard shift up applied to the top element of the stretch stack.
39	STRETCHSTACKBOTTOMSHIFT-DOWN	Standard shift down applied to the bottom element of the stretch stack. Positive for moving in the downward direction.

\fontdimen	Dimension name	Description
40	STRETCHSTACKGAPABOVEMIN	Minimum gap between the ink of the stretched element, and the (ink) bottom of the element above. Suggested: UpperLimitGapMin
41	STRETCHSTACKGAPBELOWMIN	Minimum gap between the ink of the stretched element, and the (ink) top of the element below. Suggested: LowerLimitGapMin.
42	FRACTIONNUMERATORSHIFTUP	Standard shift up applied to the numerator.
43	FRACTIONNUMERATOR- DISPLAYSTYLESHIFTUP	Standard shift up applied to the numerator in display style. Suggested: StackTopDisplayStyleShiftUp.
44	FRACTIONDENOMINATORSHIFT- DOWN	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	FRACTIONNUMDISPLAYSTYLE- GAPMIN	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	FRACTIONDENOMINATORGAP- MIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar. Suggested: default rule thickness
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.



\fontdimen	Dimension name	Description
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	UNDERBAREXTRADESCENDER	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	RADICALKERNBEFOREDEGREE	Extra horizontal kern before the degree of a radical, if such is present. Suggested: 5/18 of em.
64	RADICALKERNAFTERDEGREE	Negative kern after the degree of a radical, if such is present. Suggested: -10/18 of em.

<code>\fontdimen</code>	Dimension name	Description
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 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 *
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{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ΣΤΥΦΧΨΩ}
10 \def\upgreek{αβγδεζηθικλμνξοπρστυφχψω}
11 \def\itLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
12 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
13 \def\itGreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ΣΤΥΦΧΨΩ}
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]{Asana 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 /

```

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 *
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{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ϚΣΤΥΦΧΨΩ}
45 \def\upgreek{αβγδεζηθικλμνξοπρςστυφχψω}
46 \def\itLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
47 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
48 \def\itGreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ϚΣΤΥΦΧΨΩ}
49 \def\itgreek{αβγδεζηθικλμνξοπρςστυφχψω}
50 \def\bfulatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
51 \def\bfulatin{abcdefghijklmnopqrstuvwxyz}
52 \def\bfulgreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ϚΣΤΥΦΧΨΩ}
53 \def\bfulgreek{αβγδεζηθικλμνξοπρςστυφχψω}
54 \def\bfitLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
55 \def\bfitlatin{abcdefghijklmnopqrstuvwxyz}
56 \def\bfitGreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ϚΣΤΥΦΧΨΩ}
57 \def\bfitgreek{αβγδεζηθικλμνξοπρςστυφχψω}
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]{Asana 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 /

```

# 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
\'	1185
\:::	666
\::f	666
\::n	666
\@cclvi	303
\@cdots	1208
\@elt	549–553, 556, 560, 562
\@empty	354, 355, 392, 477, 594, 601, 616, 637, 642
\@gobble	1130
\@ifnextchar	1137
\@ifpackageloaded	1211
\@ii	600, 601, 603, 605, 608, 613
\@input	409, 1135
\@marker	613, 632
\@nil	312, 458, 613, 626–629, 1119, 1124, 1133
\@preamblecmds	288
\@sqrt	1137
\@tempa	140, 183, 206, 224, 242, 365, 381, 592, 604, 628, 630, 642
\@tempb	140, 141, 183, 184, 206, 207, 224, 225, 242, 243, 592, 593, 631, 632, 637
\@tempswafalse	602
\@tempwattrue	606, 609, 634, 639, 644, 649
\@um@bfliteraltrue	179, 203
\@um@bfupGreekfalse	146, 185
\@um@bfupGreektrue	158, 170, 191, 197
\@um@bfupLatinfalse	149, 188
\@um@bfupLatintrue	161, 173, 194, 200
\@um@bfupgreekfalse	147, 159, 186, 192
\@um@bfupgreektrue	171, 198
\@um@bfuplatinfalse	150, 189
\@um@bfuplatintrue	162, 174, 195, 201
\@um@fontspec@featuretrue	356
\@um@literaltrue	178
\@um@ot@math@true	384
\@um@upGreekfalse	142
\@um@upGreektrue	154, 166
\@um@upLatinfalse	144, 156
\@um@upLatintrue	168
\@um@upNablafalse	151, 210
\@um@upNablatrue	163, 175, 208
\@um@upgreekfalse	143, 155
\@um@upgreektrue	167
\@um@uplatinfalse	145, 157, 169
\@um@uppartialfalse	148, 160, 187, 193, 228
\@um@uppartialtrue	172, 199, 226
\@	10–13, 17, 23–33, 72–74
\^	33, 1118
Numbers	
\0	28
\_	17–20, 23–32
A	
\A	29
\a	30
\addnolimits	555
\advance	30, 77
\alloc@	303
\Alpha	693
\alpha	718
\AtBeginDocument	691, 1207, 1210
\awint	551
B	
\B	905
\beamer@suppressreplacements	1217
\begin	19, 66
\begingroup	309, 559, 832, 1117, 1131
\Beta	694
\beta	719
\bfitGreek	56
\bfitgreek	57
\bfitLatin	54

<code>\bfitlatin</code> .....	55	<code>\csname</code> .....	17, 18, 61–64, 306, 312, 315, 318, 350, 357, 472, 544, 815, 836
<code>\bfupgreek</code> .....	52	<b>D</b>	
<code>\bfupgreek</code> .....	53	<code>\DeclareDocumentCommand</code> .....	352
<code>\bfuplatin</code> .....	50	<code>\DeclareMathVersion</code> .....	359
<code>\bfuplatin</code> .....	51	<code>\DeclareRobustCommand</code> .....	1137
<code>\bgroup</code> .....	1162	<code>\DeclareSymbolFont</code> .....	407
<code>\bool_if:NTF</code> .....	250	<code>\def</code> .....	7–15, 23, 24, 28–68, 70–84, 89–97, 102–109, 114–122, 127–138, 288, 289, 292, 303, 305, 307, 358, 393, 548, 556, 558, 560, 567, 569, 626, 628–631, 693–748, 1126, 1132, 1190, 1201–1206, 1208
<code>\bool_new:N</code> .....	22	<code>\define@choicekey</code> .....	139, 183, 206, 224, 242, 592
<code>\bool_set_false:N</code> ...	152, 176, 180, 244	<code>\define@cmdkey</code> .....	588–591
<code>\bool_set_true:N</code> .....	164, 246	<code>\define@key</code> .....	571
<code>\box</code> .....	32, 33, 79, 80, 1199	<code>\Delta</code> .....	696
<b>C</b>		<code>\delta</code> .....	721
<code>\C</code> .....	894, 920	<code>\dimen</code> .....	29–31, 76–78
<code>\cdots</code> .....	1208	<code>\dimexpr</code> .....	290, 383, 1193
<code>\char_make_active:N</code> .....	1185	<code>\documentclass</code> .....	2, 37
<code>\char_make_active:n</code> .....	310, 1186	<code>\dp</code> .....	30, 77, 1195
<code>\char_make_other:N</code> .....	1118	<b>E</b>	
<code>\chardef</code> .....	303	<code>\E</code> .....	906
<code>\Chi</code> .....	715	<code>\e</code> .....	914
<code>\chi</code> .....	740	<code>\edef</code> .....	31, 78, 365, 401, 627, 628
<code>\cirfnint</code> .....	551	<code>\egroup</code> .....	1178
<code>\clist_map_inline:Nn</code> .....	534	<code>\else</code> .....	218, 236, 295, 298, 324, 328, 333, 338, 341, 385, 399, 437, 450, 481–485, 489, 492, 497, 503, 515, 525, 561, 574, 607, 612, 618, 636, 641, 646, 976, 979, 985, 992, 1005, 1212
<code>\clist_map_inline:nn</code> ...	463, 529, 536	<code>\else:</code> .....	827
<code>\clist_map_variable:NNn</code> .....	600	<code>\encodingdefault</code> .....	408
<code>\clist_map_variable:nNn</code> .	657, 668, 674	<code>\end</code> .....	34, 81
<code>\copy</code> .....	1197	<code>\endcsname</code> ..	17, 18, 61–64, 306, 312, 315, 318, 350, 357, 472, 544, 815, 836
<code>\cs_gset:cpn</code> .....	317, 327	<code>\endgroup</code> .....	313, 565, 832, 1129, 1136
<code>\cs_gset:Npn</code> .....	330, 335	<code>\Epsilon</code> .....	697
<code>\cs_gset:Npx</code> .....	340	<code>\epsilon</code> .....	722
<code>\cs_gset_eq:NN</code> .....	1187, 1188	<code>\Eta</code> .....	699
<code>\cs_if_exist:cF</code> .....	830	<code>\eta</code> .....	724
<code>\cs_new:Nn</code> .....	416, 430, 457, 462, 467, 470, 667, 673, 682, 685, 688, 814, 829, 842, 1141, 1145, 1161, 1166	<code>\etex_iffontchar:D</code> .....	827
<code>\cs_new:Npn</code> .....	860, 875, 891, 903, 918, 927, 932, 937, 942, 1019, 1039, 1059, 1064, 1069, 1085, 1101		
<code>\cs_set:cpn</code> .....	831		
<code>\cs_set:Nn</code> .....	213, 231, 249, 476, 528, 533, 753, 764, 823		
<code>\cs_set:Npn</code> .....	666		
<code>\cs_set_eq:cN</code> .....	824		
<code>\cs_set_eq:NN</code> ..	397, 398, 404, 405, 1183		
<code>\cs_set_protected:cpn</code> .....	834		
<code>\cs_to_str:N</code> .....	327, 544		



<code>\if@um@bfupLatin</code> . . . . .	18, 490, 977
<code>\if@um@bfuplatin</code> . . . . .	19, 495, 982
<code>\if@um@fontspec@feature</code> . . . . .	8, 572
<code>\if@um@literal</code> . . . . .	10, 432, 479
<code>\if@um@ot@math@</code> . . . . .	9
<code>\if@um@upGreek</code> . . . . .	11, 484
<code>\if@um@upgreek</code> . . . . .	12, 485
<code>\if@um@upLatin</code> . . . . .	13, 482
<code>\if@um@uplatin</code> . . . . .	14, 483
<code>\if@um@upNabla</code> . . . . .	20, 214
<code>\if@um@uppartial</code> . . . . .	21, 232
<code>\ifbeamer@suppressreplacements</code> .	1212
<code>\ifcase</code> . . . . .	141, 184, 207, 225, 243, 593
<code>\ifdim</code> . . . . .	383
<code>\ifin@</code> . . . . .	320, 326
<code>\ifnum</code> . . . . .	535, 633, 638, 643, 647, 648
<code>\ifx</code> . . . . .	293, 296, 308, 329, 334, 339, 392, 477, 561, 601, 604, 605, 608, 616, 632, 637, 642
<code>\iiiint</code> . . . . .	549
<code>\iiint</code> . . . . .	549
<code>\iint</code> . . . . .	549
<code>\in@</code> . . . . .	319, 325
<code>\int</code> . . . . .	549
<code>\intBar</code> . . . . .	551
<code>\intbar</code> . . . . .	551
<code>\intcap</code> . . . . .	553
<code>\intclockwise</code> . . . . .	550
<code>\intcup</code> . . . . .	553
<code>\intlarhk</code> . . . . .	552
<code>\intx</code> . . . . .	552
<code>\Iota</code> . . . . .	701
<code>\iota</code> . . . . .	726
<code>\itGreek</code> . . . . .	13, 48
<code>\itgreek</code> . . . . .	14, 49
<code>\itLatin</code> . . . . .	11, 46
<code>\itlatin</code> . . . . .	12, 47
<b>K</b>	
<code>\Kappa</code> . . . . .	702
<code>\kappa</code> . . . . .	727
<code>\kern</code> . . . . .	1192, 1198
<b>L</b>	
<code>\L</code> . . . . .	910
<code>\l_um_inc_num</code> . . . . .	675, 677, 678
<code>\l_um_incr_num</code> . . . . .	658, 660, 662
<code>\l_um_input_num</code> . . . . .	657, 660, 668, 670, 674, 677
<code>\l_um_primecount_num</code> . . . . .	1140, 1163, 1167, 1177
<code>\l_um_script_features_tl</code> . . .	360, 371
<code>\l_um_script_font_tl</code> . . . . .	362, 370
<code>\l_um_sscript_features_tl</code> . .	361, 375
<code>\l_um_sscript_font_tl</code> . . . . .	363, 374
<code>\Lambda</code> . . . . .	703
<code>\lambda</code> . . . . .	728
<code>\le</code> . . . . .	1203
<code>\left</code> . . . . .	568
<code>\left@primitive</code> . . . . .	568, 569
<code>\leq</code> . . . . .	1203
<code>\let</code> . . . . .	55, 69, 304, 353–355, 395, 396, 402, 403, 568, 594, 1121, 1130
<code>\linewidth</code> . . . . .	16, 60
<code>\lowercase</code> . . . . .	1120, 1125
<code>\lowint</code> . . . . .	553
<b>M</b>	
<code>\M</code> . . . . .	911
<code>\m@th</code> . . . . .	1191
<code>\makebox</code> . . . . .	16–18, 60–64
<code>\mathaccent</code> . . . . .	339
<code>\mathalpha</code> . . . . .	545, 661
<code>\mathalphabet</code> . . . . .	58, 61, 62
<code>\mathbb</code> . . . . .	892–901
<code>\mathbf</code> 58, 943–945, 947–975, 978, 980, 983, 984, 986, 987, 990, 991, 993, 994, 997–1004, 1006–1013, 1015, 1016	
<code>\mathbffrak</code> . . . . .	1060–1062
<code>\mathbfit</code> . . . . .	1020–1037
<code>\mathbfscr</code> . . . . .	1065–1067
<code>\mathbfsf</code> . . . . .	1070–1083
<code>\mathbfsfit</code> . . . . .	1102–1115
<code>\mathbfsfup</code> . . . . .	1086–1099
<code>\mathbfup</code> . . . . .	1040–1057
<code>\mathbin</code> . . . . .	431
<code>\mathcal</code> . . . . .	1205
<code>\mathchar@type</code> . . . . .	316, 330, 332, 335, 337, 340, 342, 350, 471
<code>\mathclose</code> . . . . .	334
<code>\mathcode</code> . . . . .	311
<code>\mathfrak</code> . . . . .	919–925
<code>\mathgroup</code> . . . . .	303
<code>\mathinner</code> . . . . .	1208





\num_zero: N	1163	\protect	578
\number	670, 677, 678	\providecommand	58
\numexpr	638,	\ProvidesPackage	1
643, 647, 648, 660, 662, 670, 677, 678		\Psi	716
		\psi	741
<b>O</b>			
\o	916	<b>Q</b>	
\oiint	549	\Q	898
\oint	549	<b>R</b>	
\ointclockwise	550	\R	899, 912, 923
\Omega	717	\r@@t	1190
\omega	742	\raise	1193
\Omicron	707	\relax	141,
\omicron	732	184, 207, 225, 243, 290, 308, 311,	
\operator@font	1201	316, 327, 329–332, 334–337, 339,	
\or	153,	340, 342, 350, 353, 382, 383, 535,	
165, 177, 190, 196, 202, 209, 227, 245		593, 605, 608, 632, 633, 638, 643,	
		647, 648, 660, 662, 670, 677, 678, 1196	
<b>P</b>			
\P	897	\removenolimits	558
\PackageError	25, 575	\RequirePackage	3–5
\PackageInfo	27	\Rho	709
\PackageWarning	26	\rho	734
\PackageWarningNoLine	1213	\rightarrow	1202
\papersize	31, 32, 78, 79	\rootbox	1197
\parindent	21, 68	\rppolint	551
\peek_charcode_remove: NTF	1168, 1174	<b>S</b>	
\peek_meaning_remove: NTF	1171	\scan_stop:	473, 474, 827
\Phi	714	\scantokens	1121, 1126
\phi	739	\scpolint	552
\Pi	708	\scriptscriptstyle	296
\pi	733	\scriptstyle	293
\pointint	552	\setbox	24, 32, 71, 79, 1191
\prg_case_int: nnn	1146	\setkeys	364
\prg_do_nothing:	824	\setmainfont	5, 40
\prg_new_conditional: Nnn	826	\SetMathCode	349, 464, 545, 659
\prg_replicate: nn	1143	\setmathfont	20, 67, 352, 578
\prg_return_false:	827	\sf@size	369, 373
\prg_return_true:	827	\shipout	33, 80
\prg_stepwise_variable: nnnNn	658, 675	\Sigma	711
\prime	1183	\sigma	736
\primedouble	1149	\space	836
\primequadruple	1155	\special	32, 79
\primesingle	468, 1142, 1143, 1147	\sqint	552
\primetriple	1152	\sqrt	567, 1137
\ProcessOptionsX	287	\sqrtsign	1137, 1191

<code>\stepcounter</code> .....	400	<code>\um@nolimits</code> .....	319, <u>548</u> , 556, 564
<code>\string</code> .. 312, 315, 317, 318, 426, 627, 628		<code>\um@PackageError</code> .....	25
<code>\strip@pt</code> .....	290	<code>\um@PackageInfo</code> .....	27, 394, 426
<code>\sumint</code> .....	550	<code>\um@PackageWarning</code> .....	26, 386, 820
<b>T</b>		<code>\um@parse@range</code> .....	613, <u>629</u>
<code>\Tau</code> .....	712	<code>\um@parse@term</code> .....	425, 458, <u>599</u>
<code>\tau</code> .....	737	<code>\um@Pool</code> .....	23, 24
<code>\testmath</code> .....	15, 25–28, 59, 72–75	<code>\um@radicals</code> .....	325, <u>567</u>
<code>\tf@size</code> .....	368, 369	<code>\um@resolve@greek</code> .....	<u>691</u>
<code>\the</code> .....	31, 78	<code>\um@scaled@apply</code> .....	<u>292</u> , 1192, 1198
<code>\Theta</code> .....	700	<code>\um@scanactivedef</code> .....	312, <u>1117</u>
<code>\theta</code> .....	725	<code>\um@scancharlet</code> .....	<u>1117</u> , 1133
<code>\theum@fam</code> .....	401	<code>\um@set@mathsymbol</code> .....	306, <u>307</u>
<code>\thinmuskip</code> .....	1139	<code>\um@setmathcode</code> .....	
<code>\tl_new: Nn</code> .....		..... 491, 493, 496, 498, 501,	
.. 85–88, 98–101, 110–113, 123–126		502, 504, 505, 508–514, 516–522,	
<code>\tl_set: Nn</code> .....	215–217,	<u>656</u> , 751, 754–762, 765–772, 775,	
219–221, 233–235, 237–239, 360–363		778, 781, 782, 785, 786, 789,	
<code>\to</code> .....	1202	790, 793, 794, 797–803, 806–812	
<b>U</b>		<code>\um@symfont</code> .....	393, 401,
<code>\um@addto@mathmap</code> .....	530, 537, <u>542</u>	407, 422, 427, 464, 472, 530, 537, 661	
<code>\um@backslash</code> .....	604, 627	<code>\um@usv@bbLatin</code> .....	38, 893
<code>\um@Break</code> .....	<u>23</u>	<code>\um@usv@bblatin</code> .....	39, 901
<code>\um@char@num@range</code> 355, 534, 616, 617, 619		<code>\um@usv@bbnum</code> .....	37, 892
<code>\um@char@range</code> 354, 392, 477, 594, 597, 600		<code>\um@usv@bfDigamma</code> .....	90, 958
<code>\um@def@itGreek</code> .....	484, <u>792</u>	<code>\um@usv@bfdigamma</code> .....	97, 966
<code>\um@def@itgreek</code> .....	485, <u>805</u>	<code>\um@usv@bfrakLatin</code> .....	62, 1061
<code>\um@def@itLatin</code> .....	482, <u>777</u>	<code>\um@usv@bfraklatin</code> .....	63, 1062
<code>\um@def@itlatin</code> .....	483, <u>780</u>	<code>\um@usv@bfGreek</code> .....	56, 501,
<code>\um@def@numbers</code> .....	478, <u>750</u>	504, 769, 951, 990, 1027, 1043, 1047	
<code>\um@def@upGreek</code> .....	484, <u>788</u>	<code>\um@usv@bfgreek</code> .....	57, 508,
<code>\um@def@upgreek</code> .....	485, <u>796</u>	516, 770, 953, 997, 1028, 1044, 1048	
<code>\um@def@upLatin</code> .....	482, <u>774</u>	<code>\um@usv@bfitGreek</code> .....	
<code>\um@def@uplatin</code> .....	483, <u>784</u>	60, 501, 504, 771, 952, 993, 1023, 1027	
<code>\um@firstchar</code> .....	603, 628	<code>\um@usv@bfitgreek</code> .....	61,
<code>\um@firsttof</code> .....	626–628	508, 516, 772, 954, 1006, 1024, 1028	
<code>\um@font</code> .....	294, 297,	<code>\um@usv@bfith</code> .....	115, 955, 987
382, 383, 827, 1192, 1194, 1195, 1198		<code>\um@usv@bfitLatin</code> .....	
<code>\um@fontdimen@percent</code> .....		58, 491, 493, 767, 948, 980, 1021, 1025	
..... <u>289</u> , 294, 297, 1194, 1195		<code>\um@usv@bfitlatin</code> .....	
<code>\um@Loop</code> .....	<u>23</u>	59, 496, 498, 768, 950, 986, 1022, 1026	
<code>\um@mathsymbol</code> .....	<u>305</u> , 422, <u>427</u>	<code>\um@usv@bfitNabla</code> .....	
<code>\um@mathsymbol@noparse</code> .....	395, <u>421</u>	..... 130, 220, 443, 451, 968, 1030	
<code>\um@mathsymbol@parse</code> .....	402, <u>424</u>	<code>\um@usv@bfitpartial</code> .....	
<code>\um@mversion</code> .....	358, 359	. 136, 238, 447, 453, 969, 1013, 1031	

$\backslash\mathrm{um@usv@bfitvarepsilon}$ . . . . .	$\backslash\mathrm{um@usv@bfvarkappa}$ . . . . .
. . . . . 117, 509, 517, 970, 1007, 1032	. . . . . 93, 511, 519, 962, 1000, 1054
$\backslash\mathrm{um@usv@bfitvarkappa}$ . . . . .	$\backslash\mathrm{um@usv@bfvarphi}$ . . . . .
. . . . . 119, 511, 519, 972, 1009, 1034	. . . . . 94, 512, 520, 963, 1001, 1055
$\backslash\mathrm{um@usv@bfitvarphi}$ . . . . .	$\backslash\mathrm{um@usv@bfvarpi}$ . . . . .
. . . . . 120, 512, 520, 973, 1010, 1035	. . . . . 96, 514, 522, 965, 1003, 1057
$\backslash\mathrm{um@usv@bfitvarpi}$ . . . . .	$\backslash\mathrm{um@usv@bfvarrho}$ . . . . .
. . . . . 122, 514, 522, 975, 1012, 1037	. . . . . 95, 513, 521, 964, 1002, 1056
$\backslash\mathrm{um@usv@bfitvarrho}$ . . . . .	$\backslash\mathrm{um@usv@bfvarTheta}$ . . . . .
. . . . . 121, 513, 521, 974, 1011, 1036	. . . . . 89, 502, 505, 956, 991, 1049
$\backslash\mathrm{um@usv@bfitvarTheta}$ . . . . .	$\backslash\mathrm{um@usv@bfvartheta}$ . . . . .
. . . . . 116, 502, 505, 967, 994, 1029	. . . . . 92, 510, 518, 961, 999, 1053
$\backslash\mathrm{um@usv@bfitvartheta}$ . . . . .	$\backslash\mathrm{um@usv@Digamma}$ . . . . . 77, 944, 958
. . . . . 118, 510, 518, 971, 1008, 1033	$\backslash\mathrm{um@usv@digamma}$ . . . . . 84, 945, 966
$\backslash\mathrm{um@usv@bflatin}$ . . . . . 53, 491,	$\backslash\mathrm{um@usv@frakLatin}$ . . . . . 42, 919, 1061
493, 765, 947, 978, 1025, 1041, 1045	$\backslash\mathrm{um@usv@fraklatin}$ . . . . . 43, 925, 1062
$\backslash\mathrm{um@usv@bflatin}$ . . . . . 54, 55, 496,	$\backslash\mathrm{um@usv@itGreek}$ . . . . .
498, 766, 949, 983, 1026, 1042, 1046	. 35, 761, 789, 793, 863, 879, 952,
$\backslash\mathrm{um@usv@bfNabla}$ . . . . .	990, 993, 1023, 1043, 1073, 1089, 1105
. . . . . 129, 216, 442, 451, 957, 1050	$\backslash\mathrm{um@usv@itgreek}$ . . . . .
$\backslash\mathrm{um@usv@bfnun}$ . . . . . 52, 943, 1020,	. 36, 797, 806, 864, 880, 954, 997,
1040, 1060, 1065, 1070, 1086, 1102	1006, 1024, 1044, 1074, 1090, 1106
$\backslash\mathrm{um@usv@bfpartial}$ . . . . .	$\backslash\mathrm{um@usv@ith}$ . . . . .
. 135, 234, 446, 453, 959, 1004, 1051	102, 757, 782, 786, 878, 955, 984, 987
$\backslash\mathrm{um@usv@bfsrLatin}$ . . . . . 64, 1066	$\backslash\mathrm{um@usv@itLatin}$ . . . . . 31, 755,
$\backslash\mathrm{um@usv@bfsrlatin}$ . . . . . 65, 1067	775, 778, 861, 876, 893, 904, 919,
$\backslash\mathrm{um@usv@bfsfGreek}$ . . . . . 70, 1073, 1089	929, 934, 939, 948, 978, 980, 1021,
$\backslash\mathrm{um@usv@bfsfgreek}$ . . . . . 71, 1074, 1090	1041, 1061, 1066, 1071, 1087, 1103
$\backslash\mathrm{um@usv@bfsfitGreek}$ . . . . . 74, 1105	$\backslash\mathrm{um@usv@itlatin}$ . . . . . 32, 756,
$\backslash\mathrm{um@usv@bfsfitgreek}$ . . . . . 75, 1106	781, 785, 862, 877, 901, 913, 925,
$\backslash\mathrm{um@usv@bfsfitLatin}$ . . . . . 72, 1103	930, 935, 940, 950, 983, 986, 1022,
$\backslash\mathrm{um@usv@bfsfitlatin}$ . . . . . 73, 1104	1042, 1062, 1067, 1072, 1088, 1104
$\backslash\mathrm{um@usv@bfsfitNabla}$ . . . . .	$\backslash\mathrm{um@usv@itNabla}$ . . . . .
. . . . . 132, 221, 445, 452, 1108	128, 219, 434, 438, 865, 881, 968,
$\backslash\mathrm{um@usv@bfsfitpartial}$ . . . . .	1015, 1030, 1050, 1076, 1092, 1108
. . . . . 138, 239, 449, 454, 1109	$\backslash\mathrm{um@usv@itpartial}$ . . . 134, 237, 436,
$\backslash\mathrm{um@usv@bfsfLatin}$ . . . . . 67, 1071, 1087	439, 866, 882, 969, 1004, 1013,
$\backslash\mathrm{um@usv@bfsflatin}$ . . . 68, 69, 1072, 1088	1016, 1031, 1051, 1077, 1093, 1109
$\backslash\mathrm{um@usv@bfsfNabla}$ . . . 131, 217, 444, 452	$\backslash\mathrm{um@usv@itvarepsilon}$ . . . . .
$\backslash\mathrm{um@usv@bbsfnun}$ . . . . . 66	104, 798, 807, 868, 884, 970, 998,
$\backslash\mathrm{um@usv@bbsfpartial}$ . 137, 235, 448, 454	1007, 1032, 1052, 1078, 1094, 1100
$\backslash\mathrm{um@usv@bbsfuplatin}$ . . . . . 69	$\backslash\mathrm{um@usv@itvarkappa}$ . . . . .
$\backslash\mathrm{um@usv@bfuph}$ . . . . . 114, 984	106, 800, 809, 870, 886, 972, 1000,
$\backslash\mathrm{um@usv@bfuplatin}$ . . . . . 55	1009, 1034, 1054, 1080, 1096, 1112
$\backslash\mathrm{um@usv@bfvarepsilon}$ . . . . .	$\backslash\mathrm{um@usv@itvarphi}$ . . . . .
. . . . . 91, 509, 517, 960, 998, 1052	107, 801, 810, 871, 887, 973, 1001,

1010, 1035, 1055, 1081, 1097, 1113	$\backslash\mathrm{um@usv@varepsilon}$ . . . . .
$\backslash\mathrm{um@usv@itvarpi}$ . . . . .	. 78, 798, 807, 868, 884, 960, 998,
109, 803, 812, 873, 889, 975, 1003,	1007, 1032, 1052, 1078, 1094, 1110
1012, 1037, 1057, 1083, 1099, 1115	$\backslash\mathrm{um@usv@varkappa}$ . . . . .
$\backslash\mathrm{um@usv@itvarrho}$ . . . . .	80, 800, 809, 870, 886, 962, 1000,
108, 802, 811, 872, 888, 974, 1002,	1009, 1034, 1054, 1080, 1096, 1112
1011, 1036, 1056, 1082, 1098, 1114	$\backslash\mathrm{um@usv@varphi}$ . . . . .
$\backslash\mathrm{um@usv@itvartheta}$ 103, 794, 867, 883,	81, 801, 810, 871, 887, 963, 1001,
967, 991, 994, 1029, 1049, 1075, 1091	1010, 1035, 1055, 1081, 1097, 1113
$\backslash\mathrm{um@usv@itvartheta}$ . . . . .	$\backslash\mathrm{um@usv@varpi}$ . . . . .
105, 799, 808, 869, 885, 971, 999,	83, 803, 812, 873, 889, 965, 1003,
1008, 1033, 1053, 1079, 1095, 1111	1012, 1037, 1057, 1083, 1099, 1115
$\backslash\mathrm{um@usv@Nabla}$ . . . . .	$\backslash\mathrm{um@usv@varrho}$ . . . . .
127, 215, 433, 438, 865, 881, 957,	82, 802, 811, 872, 888, 964, 1002,
1015, 1030, 1050, 1076, 1092, 1108	1011, 1036, 1056, 1082, 1098, 1114
$\backslash\mathrm{um@usv@num}$ . . . . . 28,	$\backslash\mathrm{um@usv@vartheta}$ . . . . .
751, 892, 928, 933, 938, 943, 1020,	. 76, 760, 790, 794, 867, 883, 956,
1040, 1060, 1065, 1070, 1086, 1102	991, 994, 1029, 1049, 1075, 1091, 1107
$\backslash\mathrm{um@usv@partial}$ . . . . . 133, 233, 435,	$\backslash\mathrm{um@usv@vartheta}$ . . . . .
439, 866, 882, 959, 1004, 1013,	. 79, 799, 808, 869, 885, 961, 999,
1016, 1031, 1051, 1077, 1093, 1109	1008, 1033, 1053, 1079, 1095, 1111
$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 40, 904	$\backslash\mathrm{um@zf@feature}$ . . . . . 570, 582, 585
$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 41, 913	$\backslash\mathrm{um\_bfNabla\_up\_or\_it\_usv}$ . . . . .
$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 47, 934	. . . . . 216, 220, 451, 1015
$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 48, 935	$\backslash\mathrm{um\_bfpartial\_up\_or\_it\_usv}$ . . . . .
$\backslash\mathrm{um@usv@sflatin}$ . . . . . 45, 929	. . . . . 234, 238, 453, 1016
$\backslash\mathrm{um@usv@sflatin}$ . . . . . 46, 930	$\backslash\mathrm{um\_bfsfNabla\_up\_or\_it\_usv}$ . . . . .
$\backslash\mathrm{um@usv@sfnun}$ . . . . . 44, 928, 933	. . . . . 217, 221, 452
$\backslash\mathrm{um@usv@ttLatin}$ . . . . . 50, 939	$\backslash\mathrm{um\_bfsfpartial\_up\_or\_it\_usv}$ . . . . .
$\backslash\mathrm{um@usv@ttLatin}$ . . . . . 51, 940	. . . . . 235, 239, 454
$\backslash\mathrm{um@usv@ttnum}$ . . . . . 49, 938	$\backslash\mathrm{um\_config\_mathbb}$ : . . . . . 891
$\backslash\mathrm{um@usv@upGreek}$ . . . . .	$\backslash\mathrm{um\_config\_mathbf}$ : . . . . . 942
. 33, 759, 789, 793, 863, 879, 951,	$\backslash\mathrm{um\_config\_mathbffrac}$ : . . . . . 1059
990, 993, 1023, 1043, 1073, 1089, 1105	$\backslash\mathrm{um\_config\_mathbf fit}$ : . . . . . 1019
$\backslash\mathrm{um@usv@upgreek}$ . . . . . 34,	$\backslash\mathrm{um\_config\_mathbf fscr}$ : . . . . . 1064
762, 797, 806, 864, 880, 953, 997,	$\backslash\mathrm{um\_config\_mathbf fsf}$ : . . . . . 1069
1006, 1024, 1044, 1074, 1090, 1106	$\backslash\mathrm{um\_config\_mathbf fs fit}$ : . . . . . 1101
$\backslash\mathrm{um@usv@upLatin}$ . . . . . 29, 754,	$\backslash\mathrm{um\_config\_mathbf fsfup}$ : . . . . . 1085
775, 778, 861, 876, 893, 904, 919,	$\backslash\mathrm{um\_config\_mathbf fup}$ : . . . . . 1039
929, 934, 939, 947, 978, 980, 1021,	$\backslash\mathrm{um\_config\_mathfrak}$ : . . . . . 918
1041, 1061, 1066, 1071, 1087, 1103	$\backslash\mathrm{um\_config\_mathit}$ : . . . . . 875
$\backslash\mathrm{um@usv@uplatin}$ . . . . . 30, 758,	$\backslash\mathrm{um\_config\_mathscr}$ : . . . . . 903
781, 785, 862, 877, 901, 913, 925,	$\backslash\mathrm{um\_config\_mathsf}$ : . . . . . 927
930, 935, 940, 949, 983, 986, 1022,	$\backslash\mathrm{um\_config\_mathsf fit}$ : . . . . . 932
1042, 1062, 1067, 1072, 1088, 1104	$\backslash\mathrm{um\_config\_mathtt}$ : . . . . . 937
	$\backslash\mathrm{um\_config\_mathup}$ : . . . . . 860

<code>\um_glyph_if_exist:n</code> . . . . .	826		
<code>\um_glyph_if_exist:nTF</code> . . . . .			
. . . . .	815, <u>826</u> , 1149, 1152, 1155		
<code>\um_init_alphabet:n</code> . . . . .	398, 823		
<code>\um_make_mathactive:nNN</code> . . . .	468, <u>470</u>		
<code>\um_mathmap:Nnn</code> . . . . .	396, 403, 669, 676		
<code>\um_mathmap_noparse:Nnn</code> . . . .	396, <u>528</u>		
<code>\um_mathmap_parse:Nnn</code> . . . . .	403, <u>533</u>		
<code>\um_maybe_init_alphabet:n</code>	398, 405, 816		
<code>\um_Nabla_up_or_it_usv</code> . .	215, 219, 438		
<code>\um_nprimes:n</code>	1141, 1149, 1152, 1155, 1158		
<code>\um_nprimes_select:n</code> . . . . .	1145, 1177		
<code>\um_partial_up_or_it_usv</code>	233, 237, 439		
<code>\um_prepare_alph:n</code> . . . . .	817, <u>829</u>		
<code>\um_remap_symbol:nnn</code> .	397, 404, 431,		
433–436, 438, 439, 442–449, 451–454			
<code>\um_remap_symbol_noparse:nnn</code> . . . .			
. . . . .	397, 459, 462		
<code>\um_remap_symbol_parse:nnn</code> .	404, 457		
<code>\um_remap_symbols:</code> . . . . .	411, <u>430</u>		
<code>\um_scanprime:</code> . . . . .			
. . . . .	1161, 1171, 1183, 1187, 1188		
<code>\um_scanprime_collect:</code> . . . . .			
. . . . .	1164, 1166, 1169, 1172, 1175		
<code>\um_set_mathalph_range:Nnn</code> . . . .	<u>673</u>		
<code>\um_set_mathalph_range:nNnn</code> . . . .			
. . . . .	673, 683, 686, 689		
<code>\um_set_mathalphabet_char:Nnn</code>	667,		
865–873, 878, 881–889, 894–900,			
905–912, 914–916, 920–924, 944,			
945, 955–975, 984, 987, 991,			
994, 998–1004, 1007–1013, 1015,			
1016, 1029–1037, 1049–1057,			
1075–1083, 1091–1099, 1107–1115			
<code>\um_set_mathalphabet_char:Nnnn</code> . .	<u>666</u>		
<code>\um_set_mathalphabet_greek:Nnn</code> . .			
. . . . .	688,		
863, 864, 879, 880, 951–954, 990,			
993, 997, 1006, 1023, 1024, 1027,			
1028, 1043, 1044, 1047, 1048,			
1073, 1074, 1089, 1090, 1105, 1106			
<code>\um_set_mathalphabet_latin:Nnn</code> . .			
. . . . .	685, 861, 862, 876, 877,		
893, 901, 904, 913, 919, 925, 929,			
930, 934, 935, 939, 940, 947–950,			
978, 980, 983, 986, 1021, 1022,			
		1025, 1026, 1041, 1042, 1045,	
		1046, 1061, 1062, 1066, 1067,	
		1071, 1072, 1087, 1088, 1103, 1104	
<code>\um_set_mathalphabet_numbers:Nnn</code>			
682, 892, 928, 933, 938, 943, 1020,			
1040, 1060, 1065, 1070, 1086, 1102			
<code>\um_setup_alphabets:</code> . . . . .	414, 842		
<code>\um_setup_alphanum:</code> . . . . .	413, <u>476</u>		
<code>\um_setup_bf_literals:</code> . . . . .	488, <u>764</u>		
<code>\um_setup_literals:</code> . . . . .	480, <u>753</u>		
<code>\um_setup_math_alphabet:n</code>	814, 843–858		
<code>\um_setup_mathactives:</code> . . . . .	412, 467		
<code>\um_setup_mathup:</code> . . . . .	1201		
<code>\um_setup_nabla:</code> . . . . .	213, 417		
<code>\um_setup_partial:</code> . . . . .	231, 418		
<code>\um_setup_shapes:</code> . . . . .	410, 416		
<code>\um_setup_vargreek:</code> . . . . .	249, 419		
<code>\unicodemathgobble</code> . . . . .	1130		
<code>\UnicodeMathSymbol</code> . . . .	395, 402, 1132		
<code>\unless</code> . . . . .	601		
<code>\updefault</code> . . . . .	408		
<code>\upGreek</code> . . . . .	9, 44		
<code>\upgreek</code> . . . . .	10, 45		
<code>\upint</code> . . . . .	553		
<code>\upLatin</code> . . . . .	7, 42		
<code>\uplatin</code> . . . . .	8, 43		
<code>\Upsilon</code> . . . . .	713		
<code>\upsilon</code> . . . . .	738		
<code>\use:c</code> . . . . .	818, 832, 838		
<code>\use_none:n</code> . . . . .	405		
<code>\usepackage</code> . . . . .	3, 4, 6, 38, 39, 41		
<b>V</b>			
<code>\varepsilon</code> . . . . .	743		
<code>\varkappa</code> . . . . .	745		
<code>\varointclockwise</code> . . . . .	550		
<code>\varphi</code> . . . . .	746		
<code>\varpi</code> . . . . .	748		
<code>\varrho</code> . . . . .	747		
<code>\varsigma</code> . . . . .	735		
<code>\varTheta</code> . . . . .	710		
<code>\vartheta</code> . . . . .	744		
<code>\vbox</code> . . . . .	24, 32, 71, 79		
<code>\voffset</code> . . . . .	22, 69		
<b>W</b>			
<code>\wd</code> . . . . .	31, 78		

<b>X</b>			
<code>\xdef</code>	564, 597	<code>\xi</code>	731
<code>\XeTeXdelcode</code>	331, 336	<code>\XKV@rm</code>	378
<code>\XeTeXdelimiter</code>	330, 335	<b>Z</b>	
<code>\XeTeXmathaccent</code>	340	<code>\Z</code>	900, 924
<code>\XeTeXmathchardef</code>	314, 471	<code>\z@</code>	1191, 1194, 1195, 1199
<code>\XeTeXmathcode</code>	332, 337, 342, 350	<code>\Zeta</code>	698
<code>\XeTeXmathcodenum</code>	474	<code>\zeta</code>	723
<code>\XeTeXradical</code>	327	<code>\zf@family</code>	408
<code>\Xi</code>	706	<code>\zf@fontspec</code>	365
		<code>\zf@update@ff</code>	583, 586