

# 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>2</b>	<b>Specification</b>	<b>2</b>			
2.1	Using multiple fonts	3		5.2	Overcoming <code>\@on-</code> <code>lypreamble</code> 15
2.2	Script and scriptscript fonts/features	3		5.3	Other things 15
<b>3</b>	<b>Maths input</b>	<b>3</b>		<b>6</b>	<b>Fundamentals 16</b>
3.1	Miscellanea	4		6.1	Enlarging the number of maths families 16
<b>4</b>	<b>Package options</b>	<b>4</b>		6.2	<code>\DeclareMathSymbol</code> for unicode ranges 16
4.1	Math ‘style’	4		6.3	The main <code>\setmathfont</code> macro 18
4.2	Bold switching	5		6.4	(Big) operators 26
4.3	Symbols requiring spe- cial attention 6			6.5	Radicals 29
				6.6	Delimiters 30
				6.7	Maths accents 32
<b>I</b>	<b>The unicode-math pack- age</b>	<b>9</b>		<b>7</b>	<b>Font features 33</b>
<b>5</b>	<b>Things we need</b>	<b>9</b>		7.1	OpenType maths font features 34
5.1	Package options	12		7.2	Script and scriptscript font options 34
				7.3	Range processing 34

<b>8 Maths alphabets mapping definitions</b>	<b>41</b>	A.1 Overview	58
8.1 Bold alphabets' character mappings	47	<b>III X<sub>Y</sub>TeX math font dimensions</b>	<b>60</b>
8.2 Definitions of the math symbols	53	<b>IV Some manner of unit testing</b>	<b>65</b>
8.3 Epilogue	53	<b>B The regular weight alphabets</b>	<b>65</b>
<b>II stix table data extraction</b>	<b>57</b>	<b>C The bold alphabets</b>	<b>66</b>
<b>A Documenting maths support in the NFSS</b>	<b>58</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  $\text{\LaTeX}$  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  $A_{BC}$ ).

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

$\text{\XeTeX}$ ’s unicode support allows maths input through two methods. Like classical  $\text{\TeX}$ , 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 (`'`), 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 `\primedouble`, `\primetriple`, 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  $\text{\LaTeX}$ ’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>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=TeX</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=French</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$

Table 3: The various forms of nabla.

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

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	<b><math>\partial</math></b>
	Italic	<b><math>\partial</math></b>
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:  $\epsilon$  vs.  $\varepsilon$  and  $\phi$  vs.  $\varphi$**   $\text{\TeX}$  defines `\epsilon` to look like  $\varepsilon$  and `\varepsilon` to look like  $\epsilon$ . The Unicode glyph directly after delta and before zeta is ‘epsilon’ and looks like  $\epsilon$ ; there is a subsequent variant of epsilon that looks like  $\varepsilon$ . 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 `\epsilon` produce  $\phi$  and  $\epsilon$  and `\varphi` and `\varepsilon` produce  $\varphi$  and  $\varepsilon$ . 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@upNabla
21 \newif\if@um@uppartial
22 \bool_new:N \g_um_texgreek_bool
```

### 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>

```
23 \def\um@usv@num{\0}
24 \def\um@usv@upLatin{\A}
25 \def\um@usv@uplatin{\a}
26 \def\um@usv@itLatin{"1D434}
27 \def\um@usv@itlatin{"1D44E}
28 \def\um@usv@upGreek{"391}
29 \def\um@usv@upgreek{"3B1}
30 \def\um@usv@itGreek{"1D6E2}
31 \def\um@usv@itgreek{"1D6FC}
32 \def\um@usv@bnum{"1D7D8}
33 \def\um@usv@bblatin{"1D538}
34 \def\um@usv@bblatin{"1D552}
35 \def\um@usv@scrLatin{"1D49C}
36 \def\um@usv@scrlatin{"1D4B6}
37 \def\um@usv@frakLatin{"1D504}
38 \def\um@usv@fraklatin{"1D51E}
39 \def\um@usv@snum{"1D7E2}
40 \def\um@usv@sflatin{"1D5A0}
41 \def\um@usv@sflatin{"1D5BA}
42 \def\um@usv@sfitLatin{"1D608}
43 \def\um@usv@sfitlatin{"1D622}
44 \def\um@usv@ttnum{"1D7F6}
45 \def\um@usv@ttlLatin{"1D670}
46 \def\um@usv@ttlLatin{"1D68A}
```

Bold:

```
47 \def\um@usv@bnum{"1D7CE}
48 \def\um@usv@bflatin{"1D400}
49 \def\um@usv@bflatin{"1D41A}
50 \let\um@usv@bfuplatin\um@usv@bflatin
51 \def\um@usv@bfgreek{"1D6A8}
52 \def\um@usv@bfgreek{"1D6C2}
53 \def\um@usv@bfitLatin{"1D468}
54 \def\um@usv@bfitlatin{"1D482}
55 \def\um@usv@bfitGreek{"1D71C}
56 \def\um@usv@bfitgreek{"1D736}
```

---

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

```

57 \def\um@usv@bfracLatin{"1D56C}
58 \def\um@usv@bfracLatin{"1D586}
59 \def\um@usv@bfsrLatin{"1D4D0}
60 \def\um@usv@bfsrLatin{"1D4EA}
61 \def\um@usv@bfsfnum{"1D7EC}
62 \def\um@usv@bfsfLatin{"1D5D4}
63 \def\um@usv@bfsfLatin{"1D5EE}
64 \let\um@usv@bfsfuplatin\um@usv@bfsfLatin
65 \def\um@usv@bfsfGreek{"1D756}
66 \def\um@usv@bfsfGreek{"1D770}
67 \def\um@usv@bfsfitLatin{"1D63C}
68 \def\um@usv@bfsfitLatin{"1D656}
69 \def\um@usv@bfsfitGreek{"1D790}
70 \def\um@usv@bfsfitGreek{"1D7AA}

```

Greek variants:

```

71 \def\um@usv@varTheta{"3F4}
72 \def\um@usv@Digamma{"3DC}
73 \def\um@usv@varepsilon{"3F5}
74 \def\um@usv@vartheta{"3D1}
75 \def\um@usv@varkappa{"3F0}
76 \def\um@usv@varphi{"3D5}
77 \def\um@usv@varrho{"3F1}
78 \def\um@usv@varpi{"3D6}
79 \def\um@usv@digamma{"3DD}

```

Bold:

```

80 \def\um@usv@bvarTheta{"1D6B9}
81 \def\um@usv@bDigamma{"1D7CA}
82 \def\um@usv@bvarepsilon{"1D6DC}
83 \def\um@usv@bvartheta{"1D6DD}
84 \def\um@usv@bvarkappa{"1D6DE}
85 \def\um@usv@bvarphi{"1D6DF}
86 \def\um@usv@bvarrho{"1D6E0}
87 \def\um@usv@bvarpi{"1D6E1}
88 \def\um@usv@bdigamma{"1D7CB}

```

Italic Greek variants:

```

89 \def\um@usv@ith{"210E}
90 \def\um@usv@itvarTheta{"1D6F3}
91 \def\um@usv@itvarepsilon{"1D716}
92 \def\um@usv@itvartheta{"1D717}
93 \def\um@usv@itvarkappa{"1D718}
94 \def\um@usv@itvarphi{"1D719}
95 \def\um@usv@itvarrho{"1D71A}
96 \def\um@usv@itvarpi{"1D71B}

```

Bold:

```

97 \def\um@usv@bfuph{"1D421}
98 \def\um@usv@bfith{"1D489}
99 \def\um@usv@bfivarTheta{"1D72D}
100 \def\um@usv@bfivarepsilon{"1D750}
101 \def\um@usv@bfivartheta{"1D751}
102 \def\um@usv@bfivarkappa{"1D752}
103 \def\um@usv@bfivarphi{"1D753}
104 \def\um@usv@bfivarrho{"1D754}
105 \def\um@usv@bfivarpi{"1D755}

```

Nabla:

```

106 \def\um@usv@Nabla{"2207}
107 \def\um@usv@itNabla{"1D6FB}
108 \def\um@usv@bfNabla{"1D6C1}
109 \def\um@usv@bfivarNabla{"1D735}
110 \def\um@usv@bfNabla{"1D76F}
111 \def\um@usv@bfivarNabla{"1D7A9}

```

Partial:

```

112 \def\um@usv@partial{"2202}
113 \def\um@usv@itpartial{"1D715}
114 \def\um@usv@bfpartial{"1D6DB}
115 \def\um@usv@bfivarpartial{"1D74F}
116 \def\um@usv@bfNabla{"1D789}
117 \def\um@usv@bfivarpartial{"1D7C3}

```

## 5.1 Package options

xkeyval's package support is used here.

### math-style

```

118 \define@choicekey*{unicode-math.sty}
119   {math-style}[ \@tempa\@tempb]{iso, tex, french, literal}{
120   \ifcase\@tempb\relax
121     \@um@upGreek false
122     \@um@upgreek false
123     \@um@upLatin false
124     \@um@uplatin false
125     \@um@bfupGreek false
126     \@um@bfupgreek false
127     \@um@uppartial false
128     \@um@bfupLatin false
129     \@um@bfuplatin false
130     \@um@upNabla false
131     \bool_set_false:N \g_um_texgreek_bool
132   \or
133     \@um@upGreek true

```

```

134 \um@upgreek false
135 \um@upLatin false
136 \um@uplatin false
137 \um@b fupGreek true
138 \um@b fupgreek false
139 \um@uppartial false
140 \um@b fupLatin true
141 \um@b fuplatin true
142 \um@upNabla true
143 \bool_set_true: N \g_um_texgreek_bool
144 \or
145 \um@upGreek true
146 \um@upgreek true
147 \um@upLatin true
148 \um@uplatin false
149 \um@b fupGreek true
150 \um@b fupgreek true
151 \um@uppartial true
152 \um@b fupLatin true
153 \um@b fuplatin true
154 \um@upNabla true
155 \bool_set_false: N \g_um_texgreek_bool
156 \or
157 \um@literal true
158 \um@b fliteral true
159 \bool_set_false: N \g_um_texgreek_bool
160 \fi
161 }

```

### bold-style

```

162 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso, tex, french, literal}{
163 \ifcase\@tempb\relax
164 \um@b fupGreek false
165 \um@b fupgreek false
166 \um@uppartial false
167 \um@b fupLatin false
168 \um@b fuplatin false
169 \or
170 \um@b fupGreek true
171 \um@b fupgreek false
172 \um@uppartial false
173 \um@b fupLatin true
174 \um@b fuplatin true
175 \or
176 \um@b fupGreek true
177 \um@b fupgreek true

```

```

178 \um@upartialtrue
179 \um@bfupLatintrue
180 \um@bfuplatintrue
181 \or
182 \um@bfliteraltrue
183 \fi
184 }

```

### Symbol obliqueness

```

185 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright,italic}{
186 \ifcase\@tempb\relax
187 \um@upNablatrue
188 \or
189 \um@upNablafalse
190 \fi
191 }
192 \cs_set:Nn \um_setup_nabla: {
193 \if@um@upNabla
194 \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@Nabla }
195 \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfNabla }
196 \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfNabla }
197 \else
198 \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@itNabla }
199 \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfitNabla }
200 \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfitNabla }
201 \fi
202 }
203 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{
204 \ifcase\@tempb\relax
205 \um@uppartialtrue
206 \or
207 \um@uppartialfalse
208 \fi
209 }
210 \cs_set:Nn \um_setup_partial: {
211 \if@um@uppartial
212 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@partial }
213 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfpartial }
214 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfpartial }
215 \else
216 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@itpartial }
217 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfitpartial }
218 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfitpartial }
219 \fi
220 }

```

## Epsilon and phi shapes

```

221 \define@choicekey*{unicode-math.sty}{vargreek-shape}[\@tempa\@tempb]{unicode,TeX}{
222   \ifcase\@tempb\relax
223     \bool_set_false:N \g_um_texgreek_bool
224   \or
225     \bool_set_true:N \g_um_texgreek_bool
226   \fi
227 }
228 \ExecuteOptionsX{math-style=TeX}
229 \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.

```

230 \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  
0.60  
0.65

```

\font\tmp font="Cambria Math"
\um@fontdimen@percent{10}{\tmp font}\
\um@fontdimen@percent{11}{\tmp font}\
\um@fontdimen@percent{65}{\tmp font}

```

```

231 \def\um@fontdimen@percent#1#2{
232   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax
233 }

```

`\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.

```

234 \def\um@scaled@apply#1#2#3{
235   \ifx#1\scriptstyle
236     #2\um@fontdimen@percent{10}\um@font#3
237   \else
238     \ifx#1\scriptscriptstyle

```

```

239     #2\um@fontdimen@percent{11}\um@font#3
240   \else
241     #2#3%
242   \fi
243 \fi
244 }

```

## 6 Fundamentals

### 6.1 Enlarging the number of maths families

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

```

245 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cclvi}
246 \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
247 \def \um@mathsymbol#1#2#3#4{
248   \expandafter\um@set@mathsymbol\csname sym#3\endcsname#1#2{#4}}

```

The final macros that actually define the maths symbol with X<sub>e</sub>T<sub>E</sub>X 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.
249 \def\um@set@mathsymbol#1#2#3#4{

```



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

```
250 \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`.

```
251 \begingroup
252 \char_make_active:n {#4}
253 \global\mathcode#4="8000\relax
254 \um@scanactivedef #4 \@nil { \csname\string#2@op\endcsname }
255 \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`.

```
256 \expandafter\global\expandafter\XeTeXmathchardef
257 \csname\string#2@sym\endcsname
258 =" \mathchar@type#3 #1 #4\relax
```

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

```
259 \cs_gset:cpn { \string#2 @op } {
260 \csname\string#2@sym\endcsname
261 \expandafter\in@\expandafter#2\expandafter{\um@nolimits}
262 \ifin@
263 \expandafter\nolimits
264 \fi
265 }
```

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

```
266 \else
```

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

```
267 \expandafter\in@\expandafter#2\expandafter{\um@radicals,}
268 \ifin@
269 \cs_gset:cpn { \cs_to_str:N #2 sign } { \XeTeXradical #1 #4 \relax }
270 \else
```

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

```
271 \ifx\mathopen#3\relax
```

```

272     \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
273     \global\XeTeXdelcode#4=#1 #4\relax
274     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
275   \else
276     \ifx\mathclose#3\relax
277       \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
278       \global\XeTeXdelcode#4=#1 #4\relax
279       \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
280     \else

```

## Accents

```

281     \ifx\mathaccent#3\relax
282     \cs_gset:Npx #2 {\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}
283   \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.

```

284       \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
285     \fi
286   \fi
287 \fi
288 \fi
289 \fi
290 }

```

`\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.

```

291 \newcommand\SetMathCode[4]{
292   \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax
293 }

```

---

A

```

\zf@fontspec{}\Asana Math}
\let\glb@currsize\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}
$Ax \eqdef \nabla \times \mathscr{L}$

```

---

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

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

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`

294 `\DeclareDocumentCommand \setmathfont { 0{} m } {`

- Erase any conception L<sup>A</sup>T<sub>E</sub>X has of previously defined math symbol fonts; this allows `\DeclareSymbolFont` at any point in the document.

295 `\let\glb@currsizel\relax`

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

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

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

- Tell fontspec that maths font features are actually allowed.

298 `\@um@fontspec@featuretrue`

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

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

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

300 `\def\um@mversion{normal}`

301 `\DeclareMathVersion{\um@mversion}`

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

302 `\tl_set:Nn \l_um_script_features_tl {ScriptStyle}`

303 `\tl_set:Nn \l_um_sscript_features_tl {ScriptScriptStyle}`

304 `\tl_set:Nn \l_um_script_font_tl {#2}`

305 `\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.

```

306 \setkeys*[um]{options}{#1}
307 \edef\@tempa{\noexpand\zf@fontspec{
308   Script = Math,
309   SizeFeatures = {
310     {Size = \tf@size-} ,
311     {Size = \sf@size-\tf@size ,
312       Font = \l_um_script_font_tl ,
313       \l_um_script_features_tl
314     } ,
315     {Size = -\sf@size ,
316       Font = \l_um_sscript_font_tl ,
317       \l_um_sscript_features_tl
318     }
319   },
320   \XKV@m
321 }{#2}
322 }
323 \@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`:

```

324 \font\um@font="#2"\relax
325 \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt\relax
326   \@um@ot@math@true
327 \else
328   \PackageWarning{unicode-math}{
329     The~ font~ '#2' ~is~ not~ a~ valid~ OpenType~ maths~ font.~
330     Some~ maths~ features~ will~ not~ be~ available~ or~ behave~
331     in~ a~ substandard~ manner.
332   }
333 \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

```

334 \ifx\um@char@range\@empty
335   \def\um@symfont{um@allsym}
336   \PackageInfo{unicode-math}{Defining~ the~ default~ maths~ font~ as~ '#2' }
337   \let \UnicodeMathSymbol \um@mathsymbol@noparse
338   \let \um_mathmap: Nnn \um_mathmap_noparse: Nnn
339   \cs_set_eq: NN \um_remap_symbol: nnn \um_remap_symbol_noparse: nnn

```

```

340 \cs_set_eq:NN \um_maybe_init_alphabet:n \um_init_alphabet:n
341 \else
342 \stepcounter{um@fam}
343 \edef\um@symfont{\um@fam\theum@fam}
344 \let \UnicodeMathSymbol \um@mathsymbol@parse
345 \let \um_mathmap:Nnn \um_mathmap_parse:Nnn
346 \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_parse:nnn
347 \cs_set_eq:NN \um_maybe_init_alphabet:n \use_none:n
348 \fi

```

Now defined `\um@symfont` as the  $\text{\LaTeX}$  math font to access everything:

```

349 \DeclareSymbolFont{\um@symfont}
350 {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`.

```

351 \@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.)

```

352 \um_setup_shapes:
353 \um_remap_symbols:
354 \um_setup_mathactives:
355 \um_setup_alphanum:
356 \um_setup_alphabets:

```

End of the `\setmathfont` macro.

```

357 }

358 \cs_new:Nn \um_setup_shapes: {
359 \um_setup_nabla:
360 \um_setup_partial:
361 }

```

### 6.3.1 Functions for setting up symbols with mathcodes

`\um@mathsymbol@noparse`

```

362 \newcommand\um@mathsymbol@noparse[4]{
363 \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
364 }

```

`\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.

```

365 \newcommand\um@mathsymbol@parse[4]{
366   \um@parse@term{#1}{#2}{#3}{
367     \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
368   }
369 }

```

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

```

370 \cs_new:Nn \um_remap_symbols: {
371   \um_remap_symbol:nnn{"2D}{\mathbin}{"02212}% hyphen to minus
372   \if@um@literal
373     \um_remap_symbol:nnn {\um@usv@Nabla}{\mathord}{\um@usv@Nabla}
374     \um_remap_symbol:nnn {\um@usv@itNabla}{\mathord}{\um@usv@itNabla}
375     \um_remap_symbol:nnn {\um@usv@partial}{\mathord}{\um@usv@partial}
376     \um_remap_symbol:nnn {\um@usv@itpartial}{\mathord}{\um@usv@itpartial}
377   \else
378     \um_remap_symbol:nnn {\um@usv@Nabla,\um@usv@itNabla}{\mathord}{\um_Nabla_up_or_it_usv}
379     \um_remap_symbol:nnn {\um@usv@partial,\um@usv@itpartial}{\mathord}{\um_partial_up_or_it_usv}
380   \fi

```

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

```

381   \if@um@bfliteral
382     \um_remap_symbol:nnn {\um@usv@b fNabla}{\mathord}{\um@usv@b fNabla}
383     \um_remap_symbol:nnn {\um@usv@b fitNabla}{\mathord}{\um@usv@b fitNabla}
384     \um_remap_symbol:nnn {\um@usv@b fs fNabla}{\mathord}{\um@usv@b fs fNabla}
385     \um_remap_symbol:nnn {\um@usv@b fs fitNabla}{\mathord}{\um@usv@b fs fitNabla}
386     \um_remap_symbol:nnn {\um@usv@b fpartial}{\mathord}{\um@usv@b fpartial}
387     \um_remap_symbol:nnn {\um@usv@b fitpartial}{\mathord}{\um@usv@b fitpartial}
388     \um_remap_symbol:nnn {\um@usv@b fs fpartial}{\mathord}{\um@usv@b fs fpartial}
389     \um_remap_symbol:nnn {\um@usv@b fs fitpartial}{\mathord}{\um@usv@b fs fitpartial}
390   \else
391     \um_remap_symbol:nnn {\um@usv@b fNabla,\um@usv@b fitNabla}{\mathord}{\um_b fNabla_up_or_it_usv}
392     \um_remap_symbol:nnn {\um@usv@b fs fNabla,\um@usv@b fs fitNabla}{\mathord}{\um_b fs fNabla_up_or_it_usv}
393     \um_remap_symbol:nnn {\um@usv@b fpartial,\um@usv@b fitpartial}{\mathord}{\um_b fpartial_up_or_it_usv}
394     \um_remap_symbol:nnn {\um@usv@b fs fpartial,\um@usv@b fs fitpartial}{\mathord}{\um_b fs fpartial_up_or_it_usv}
395   \fi
396 }

```

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

```

397 \cs_new:Nn \um_remap_symbol_parse:nnn {
398   \um@parse@term {#3} {\@nil} {#2} {
399     \um_remap_symbol_noparse:nnn {#1} {#2} {#3}
400   }

```

```

401 }
402 \cs_new:Nn \um_remap_symbol_noparse:nnn {
403   \clist_map_inline:nn {#1} {
404     \SetMathCode {##1} {#2} {\um@symfont} {#3}
405   }
406 }

```

### 6.3.2 Active math characters

`\um_setup_mathactives:`

```

407 \cs_new:Nn \um_setup_mathactives: {
408   \um_make_mathactive:nnn {"2032} \primesingle \mathord
409 }

```

`\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!

```

410 \cs_new:Nn \um_make_mathactive:nnn {
411   \XeTeXmathchardef #2 = "\mathchar@type #3
412                               \csname sym\um@symfont\endcsname
413                               #1 \scan_stop:
414   \XeTeXmathcodenum #1 = "1FFFFF \scan_stop:
415 }

```

### 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.

```

416 \cs_set:Nn \um_setup_alphanum: {
417   \ifx\um@char@range\@empty
418     \um@def@numbers

```

#### Normal weight

```

419   \ifum@literal
420     \um_setup_literals:
421   \else
422     \ifum@upLatin\um@def@upLatin\else\um@def@itLatin\fi
423     \ifum@uplatin\um@def@uplatin\else\um@def@itlatin\fi
424     \ifum@upGreek\um@def@upGreek\else\um@def@itGreek\fi
425     \ifum@upgreek\um@def@upgreek\else\um@def@itgreek\fi
426   \fi

```

## Bold

```
427 \if@um@bfliteral
428 \um_setup_bf_literals:
429 \else
430 \if@um@bfupLatin
431 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bflatin}
432 \else
433 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bfitlatin}
434 \fi
435 \if@um@bfuplatin
436 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bflatin}
437 \else
438 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bfitlatin}
439 \fi
440 \if@um@bfupGreek
441 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfgreek}
442 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}
443 \else
444 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}
445 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}
446 \fi
447 \if@um@bfupgreek
448 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfgreek}
449 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfvarepsilon}
450 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}
451 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}
452 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}
453 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}
454 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}
455 \else
456 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}
457 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}
458 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}
459 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}
460 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}
461 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}
462 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}
463 \fi
464 \fi
465 \else
: TODO : what is supposed to happen here?
466 \fi
467 }
```



### 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.

```

468 \cs_set:Nn \um_mathmap_noparse:Nnn {
469   \clist_map_inline:nn {#2} {
470     \exp_args:No \um@addto@mathmap \um@symfont {##1}{#1}{#3}
471   }
472 }
```

`\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`).

```

473 \cs_set:Nn \um_mathmap_parse:Nnn {
474   \clist_map_inline:Nn \um@char@num@range {
475     \ifnum##1=#3\relax
476       \clist_map_inline:nn {#2} {
477         \exp_args:No \um@addto@mathmap \um@symfont {####1}{#1}{#3}
478       }
479     \fi
480   }
481 }
```

`\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.

```

482 \newcommand\um@addto@mathmap[4]{
483   \expandafter\g@addto@macro
484     \csname um_setup_\cs_to_str:N #3:\endcsname{
485     \SetMathCode{#2}{\mathalpha}{#1}{#4}
486   }
487 }
```

## 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	$\sum\limits_{0\atop 1}$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod\limits_{0\atop 1}$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod\limits_{0\atop 1}$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum\limits_{0\atop 1}$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int\limits_{0\atop 1}$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint\limits_{0\atop 1}$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint\limits_{0\atop 1}$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	$\oint\limits_{0\atop 1}$	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oiint\limits_{0\atop 1}$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oiiint\limits_{0\atop 1}$	<code>\oiiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\int\limits_{0\atop 1}$	<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232	$\oint\limits_{0\atop 1}$	<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\oint\limits_{0\atop 1}$	<code>\ointctrackwise</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0	$\bigwedge\limits_{0\atop 1}$	<code>\bigwedge</code>	LOGICAL OR OPERATOR
U+022C1	$\bigvee\limits_{0\atop 1}$	<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2	$\bigcap\limits_{0\atop 1}$	<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3	$\bigcup\limits_{0\atop 1}$	<code>\bigcup</code>	UNION OPERATOR
U+027D5	$\leftthreetimes\limits_{0\atop 1}$	<code>\leftouterjoin</code>	LEFT OUTER JOIN
U+027D6	$\rightthreetimes\limits_{0\atop 1}$	<code>\rightouterjoin</code>	RIGHT OUTER JOIN

U+027D7	$\overset{1}{\times}$	<code>\fullouterjoin</code>	FULL OUTER JOIN
U+027D8	$\overset{1}{\perp}$	<code>\bigbot</code>	LARGE UP TACK
U+027D9	$\overset{1}{\top}$	<code>\bigtop</code>	LARGE DOWN TACK
U+029F8	$\overset{1}{/}$	<code>\xsol</code>	BIG SOLIDUS
U+029F9	$\overset{1}{\backslash}$	<code>\xbsol</code>	BIG REVERSE SOLIDUS
U+02A00	$\overset{1}{\odot}$	<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR
U+02A01	$\overset{1}{\oplus}$	<code>\bigoplus</code>	N-ARY CIRCLED PLUS OPERATOR
U+02A02	$\overset{1}{\otimes}$	<code>\bigotimes</code>	N-ARY CIRCLED TIMES OPERATOR
U+02A03	$\overset{1}{\cup\cdot}$	<code>\bigcupdot</code>	N-ARY UNION OPERATOR WITH DOT
U+02A04	$\overset{1}{\cup+}$	<code>\bigcupplus</code>	N-ARY UNION OPERATOR WITH PLUS
U+02A05	$\overset{1}{\sqcap}$	<code>\bigsqcap</code>	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06	$\overset{1}{\sqcup}$	<code>\bigsqcup</code>	N-ARY SQUARE UNION OPERATOR
U+02A07	$\overset{1}{\&}$	<code>\conjquant</code>	TWO LOGICAL AND OPERATOR
U+02A08	$\overset{1}{\vee}$	<code>\disjquant</code>	TWO LOGICAL OR OPERATOR
U+02A09	$\overset{1}{\times}$	<code>\bigtimes</code>	N-ARY TIMES OPERATOR
U+02A0B	$\overset{1}{\int}_0$	<code>\sumint</code>	SUMMATION WITH INTEGRAL
U+02A0C	$\overset{1}{\iiint}_0$	<code>\iiiint</code>	QUADRUPLE INTEGRAL OPERATOR
U+02A0D	$\overset{1}{f}_0$	<code>\intbar</code>	FINITE PART INTEGRAL
U+02A0E	$\overset{1}{\oint}_0$	<code>\intBar</code>	INTEGRAL WITH DOUBLE STROKE
U+02A0F	$\overset{1}{f}_0$	<code>\fint</code>	INTEGRAL AVERAGE WITH SLASH
U+02A10	$\overset{1}{f}_0$	<code>\cirfnint</code>	CIRCULATION FUNCTION
U+02A11	$\overset{1}{f}_0$	<code>\awint</code>	ANTICLOCKWISE INTEGRATION LINE INTEGRATION WITH RECTANGULAR
U+02A12	$\overset{1}{f}_0$	<code>\rppoint</code>	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A13	$\overset{1}{f}_0$	<code>\scpoint</code>	PATH AROUND POLE LINE INTEGRATION NOT INCLUDING THE
U+02A14	$\overset{1}{f}_0$	<code>\npoint</code>	POLE

U+02A15		<code>\pointint</code>	INTEGRAL AROUND A POINT OPERATOR
U+02A16		<code>\sqint</code>	QUATERNION INTEGRAL OPERATOR
U+02A17		<code>\intlarhk</code>	INTEGRAL WITH LEFTWARDS ARROW WITH HOOK
U+02A18		<code>\intx</code>	INTEGRAL WITH TIMES SIGN
U+02A19		<code>\intcap</code>	INTEGRAL WITH INTERSECTION
U+02A1A		<code>\intcup</code>	INTEGRAL WITH UNION
U+02A1B		<code>\upint</code>	INTEGRAL WITH OVERBAR
U+02A1C		<code>\lowint</code>	INTEGRAL WITH UNDERBAR
U+02A1D		<code>\Join</code>	JOIN
U+02A1E		<code>\bigtriangleleft</code>	LARGE LEFT TRIANGLE OPERATOR
U+02A1F		<code>\zcmp</code>	Z NOTATION SCHEMA COMPOSITION
U+02A20		<code>\zpipe</code>	Z NOTATION SCHEMA PIPING
U+02A21		<code>\zproject</code>	Z NOTATION SCHEMA PROJECTION
U+02AFC		<code>\biginterleave</code>	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFF		<code>\bigtalloblong</code>	N-ARY WHITE VERTICAL BAR

`\um@nolimits` This macro is a sequence containing those maths operators that require a `\nolimits` suffix. This list is used when processing `unicode-math-table.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 16). I’ve chosen essentially just the operators that look like integrals; hopefully a better mathematician can help me out here. I’ve a feeling that it’s more useful *not* to include the multiple integrals such as , but that might be a matter of preference.

```

488 \def\um@nolimits{
489   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oiint\@elt\oiint
490   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrclockwise\@elt\sumint
491   \@elt\intbar\@elt\intBar\@elt\oint\@elt\cirfnint\@elt\awint\@elt\rppoint
492   \@elt\scpolint\@elt\ntopolint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
493   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint
494 }

```

`\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`.

```

495 \newcommand\addnolimits[1]{
496   \expandafter\def\expandafter\um@nolimits\expandafter{\um@nolimits\@elt#1}
497 }

```

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

```

498 \def\removenolimits#1{
499   \begingroup
500     \def\@elt##1{
501       \ifx##1#1\else
502         \noexpand\@elt\noexpand##1
503       \fi}
504     \xdef\um@nolimits{\um@nolimits}
505   \endgroup
506 }

```

---


$$\iiint_V$$

$$\iiint_V$$

$$\iiint_V$$

```

\def\dmath#1{\displaystyle #1$}
\setmathfont{Asana Math} \dmath{\iint_V}
\removenolimits\iint
\setmathfont{Asana Math} \dmath{\iint_V}
\addnolimits\iint
\setmathfont{Asana Math} \dmath{\iint_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.

```

507 \def\um@radicals{\sqrt}

```

---


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


---

```

\[ \sqrt{1+\sqrt{1+
\sqrt{1+ \sqrt{1+
\sqrt{1+\sqrt{1+
\sqrt{1+x}}}}}} \]

```

---



---


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


---

```

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

```

---

## 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>





















```
508 \let\left@primitive\left
509 \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`

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

Here are all `\mathopen` characters:

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

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

And `\mathclose`:

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

U+027C6	⌋	<code>\rbag</code>	RIGHT S-SHAPED BAG DELIMITER MATHEMATICAL RIGHT WHITE SQUARE
U+027E7	⌋	<code>\rBrack</code>	BRACKET
U+027E9	⌋	<code>\rangle</code>	MATHEMATICAL RIGHT ANGLE BRACKET MATHEMATICAL RIGHT DOUBLE ANGLE
U+027EB	⌋	<code>\rAngle</code>	BRACKET MATHEMATICAL RIGHT WHITE TORTOISE
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 RIGHT SQUARE BRACKET WITH TICK IN
U+0298E	⌋	<code>\rbracklrtick</code>	BOTTOM CORNER RIGHT SQUARE BRACKET WITH TICK IN TOP
U+02990	⌋	<code>\rbrackurtick</code>	CORNER
U+02992	⌋	<code>\rangledot</code>	RIGHT ANGLE BRACKET WITH DOT
U+02994	⌋	<code>\rpangtr</code>	RIGHT ARC GREATER-THAN BRACKET
U+02998	⌋	<code>\rblkrbrak</code>	RIGHT BLACK TORTOISE SHELL BRACKET
U+029D9	⌋	<code>\rvzigzag</code>	RIGHT WIGGLY FENCE
U+029DB	⌋	<code>\Rvzigzag</code>	RIGHT DOUBLE WIGGLY FENCE
U+029FD	⌋	<code>\rcurvyangle</code>	RIGHT POINTING CURVED ANGLE BRACKET
U+03015		<code>\rbrbrak</code>	RIGHT BROKEN BRACKET
U+03019		<code>\Rbrbrak</code>	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>\ocommatopright</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>\overleftrightharpoon</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>\underrightharpoonowdown</code>	BARB DOWNWARDS COMBINING LEFTWARDS HARPOON WITH
U+020ED	⋈̸	<code>\underleftharpoonowdown</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.

```

510 \newcommand\um@zf@feature[2]{
511   \define@key[zf]{options}{#1}[] {
512     \if@um@fontspec@feature
513       #2
514     \else
515       \PackageError{fontspec/unicode-math}
516       {The ‘#1’ font feature can only be used for maths fonts}
517       {The feature you tried to use can only be in commands
518        like \protect\setmathfont}
519     \fi
520   }
521 }
```

## 7.1 OpenType maths font features

```

522 \um@zf@feature{ScriptStyle}{
523   \zf@update@ff{+ssty=0}
524 }
525 \um@zf@feature{ScriptScriptStyle}{
526   \zf@update@ff{+ssty=1}
527 }

```

## 7.2 Script and scriptscript font options

```

528 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
529 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
530 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
531 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

## 7.3 Range processing

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

```

532 \define@choicekey+[um]{options}{Range}[ \@tempa\@tempb]{ALL}{
533   \ifcase\@tempb\relax
534     \global\let\um@char@range\@empty
535   \fi
536 }{
537   \xdef\um@char@range{#1}
538 }

```

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.

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

```

539 \newcommand\um@parse@term[4]{
540   \clist_map_variable:NNn \um@char@range \@ii {
541     \unless\ifx\@ii\@empty
542       \@tempswafalse

```

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

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

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

```

543 \expandafter\um@firstchar\expandafter{\@ii}
544 \ifx\@tempa\um@backslash
545 \expandafter\ifx\@ii#2\relax
546 \@tempswatrue
547 \else
548 \expandafter\ifx\@ii#3\relax
549 \@tempswatrue
550 \fi
551 \fi

```

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

```

552 \else
553 \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
554 \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.

```

555 \if@tempswa
556 \ifx\um@char@num@range\@empty
557 \g@addto@macro\um@char@num@range{#1}
558 \else
559 \g@addto@macro\um@char@num@range{, #1}
560 \fi
561 #4%
562 \fi
563 \fi
564 }
565 }
566 \def\um@firstof#1#2\@nil{#1}
567 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
568 \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

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

569	<code>\def\um@parse@range#1-#2-#3\@nil#4\@nil{</code>
570	<code>\def\@tempa{#1}</code>
571	<code>\def\@tempb{#2}</code>
<hr/>	
Range	$r = x$
C-list input	<code>\@ii=X</code>
Macro input	<code>\um@parse@range X-\@marker-\@nil#1\@nil</code>
Arguments	$\#1-\#2-\#3 = X-\textcolor{blue}{\@marker}-\{\}$
<hr/>	
572	<code>\expandafter\ifx\expandafter\@marker\@tempb\relax</code>
573	<code>\ifnum#4=#1\relax</code>
574	<code>\@tempswatru</code>
575	<code>\fi</code>
576	<code>\else</code>
<hr/>	
Range	$r \geq x$
C-list input	<code>\@ii=X-</code>
Macro input	<code>\um@parse@range X--\@marker-\@nil#1\@nil</code>
Arguments	$\#1-\#2-\#3 = X-\{\}-\textcolor{green}{\@marker-}$
<hr/>	
577	<code>\ifx\@empty\@tempb</code>
578	<code>\ifnum#4&gt;\numexpr#1-1\relax</code>
579	<code>\@tempswatru</code>
580	<code>\fi</code>
581	<code>\else</code>
<hr/>	
Range	$r \leq y$
C-list input	<code>\@ii=-Y</code>
Macro input	<code>\um@parse@range -Y-\@marker-\@nil#1\@nil</code>
Arguments	$\#1-\#2-\#3 = \{\}-Y-\textcolor{green}{\@marker-}$
<hr/>	
582	<code>\ifx\@empty\@tempa</code>
583	<code>\ifnum#4&lt;\numexpr#2+1\relax</code>
584	<code>\@tempswatru</code>
585	<code>\fi</code>
<hr/>	
Range	$x \leq r \leq y$
C-list input	<code>\@ii=X-Y</code>
Macro input	<code>\um@parse@range X-Y-\@marker-\@nil#1\@nil</code>
Arguments	$\#1-\#2-\#3 = X-Y-\textcolor{green}{\@marker-}$

```

586         \else
587         \ifnum#4>\numexpr#1-1\relax
588         \ifnum#4<\numexpr#2+1\relax
589         \@tempwattrue
590         \fi
591       \fi
592     \fi
593   \fi
594 \fi
595 }

```

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

```

596 \newcommand\um@setmathcode[3][1]{
597   \clist_map_variable:nNn {#2} \l_um_input_num {
598     \prg_stepwise_variable:nnnNn{1}{1}{#1} \l_um_incr_num {
599       \SetMathCode
600       {\numexpr \l_um_incr_num+ \l_um_input_num - 1\relax}
601       {\mathalpha}{\um@symfont}
602       {\numexpr \l_um_incr_num + #3 - 1\relax}
603     }
604   }
605 }

```

`\um_set_mathalphabet_char:Nnnn` #1 : Maths alphabet  
 #2 : Input char(s)  
 #3 : Output char  
 Loops through character ranges setting `\mathcode`.

```

606 \cs_set:Npn \exp_args:Nnff {\::n\::f\::f\:::}
607 \cs_new:Nn \um_set_mathalphabet_char:Nnn {
608   \clist_map_variable:nNn {#2} \l_um_input_num {
609     \exp_args:Nnff \um_mathmap:Nnn {#1}
610     {\number\numexpr\l_um_input_num\relax} {\number\numexpr#3\relax}
611   }
612 }

```

`\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`.

```

613 \cs_new:Nn \um_set_mathalph_range:nNnn {
614   \clist_map_variable:nNn {#3} \l_um_input_num {
615     \prg_stepwise_variable:nnnNn {0}{1}{#1} \l_um_inc_num {
616       \exp_args:Nnff \um_mathmap:Nnn {#2}

```

```

617         {\number\numexpr \l_um_inc_num + \l_um_input_num \relax}
618         {\number\numexpr \l_um_inc_num + #4 \relax}
619     }
620 }
621 }
622 \cs_new:Nn \um_set_mathalphabet_numbers:Nnn {
623     \um_set_mathalph_range:nNnn {9}{#1}{#2}{#3}
624 }
625 \cs_new:Nn \um_set_mathalphabet_latin:Nnn {
626     \um_set_mathalph_range:nNnn {25}{#1}{#2}{#3}
627 }
628 \cs_new:Nn \um_set_mathalphabet_greek:Nnn {
629     \um_set_mathalph_range:nNnn {24}{#1}{#2}{#3}
630 }

```

---

**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.

```

631 \AtBeginDocument{\um@resolve@greek}
632 \newcommand\um@resolve@greek{
633     \def\Alpha{\mitAlpha}
634     \def\Beta{\mitBeta}
635     \def\Gamma{\mitGamma}
636     \def\Delta{\mitDelta}
637     \def\Epsilon{\mitEpsilon}
638     \def\Zeta{\mitZeta}
639     \def\Eta{\mitEta}
640     \def\Theta{\mitTheta}
641     \def\Iota{\mitIota}
642     \def\Kappa{\mitKappa}
643     \def\Lambda{\mitLambda}
644     \def\Mu{\mitMu}
645     \def\Nu{\mitNu}
646     \def\Xi{\mitXi}
647     \def\Omicron{\mitOmicron}
648     \def\Pi{\mitPi}
649     \def\Rho{\mitRho}
650     \def\varTheta{\mitvarTheta}
651     \def\Sigma{\mitSigma}
652     \def\Tau{\mitTau}

```

```

653 \def\Upsilon{\mitUpsilon}
654 \def\Phi{\mitPhi}
655 \def\Chi{\mitChi}
656 \def\Psi{\mitPsi}
657 \def\Omega{\mitOmega}

Lowercase:
658 \def\alpha{\mitalpha}
659 \def\beta{\mitbeta}
660 \def\gamma{\mitgamma}
661 \def\delta{\mitdelta}
662 \def\epsilon{
663   \bool_if:NTF \g_um_texgreek_bool {\mitvarepsilon}{\mitepsilon}
664 }
665 \def\zeta{\mitzeta}
666 \def\eta{\miteta}
667 \def\theta{\mittheta}
668 \def\iota{\mitiota}
669 \def\kappa{\mitkappa}
670 \def\lambda{\mitlambda}
671 \def\mu{\mitmu}
672 \def\nu{\mitnu}
673 \def\xi{\mitxi}
674 \def\omicron{\mitomicron}
675 \def\pi{\mitpi}
676 \def\rho{\mitrho}
677 \def\varsigma{\mitvarsigma}
678 \def\sigma{\mitsigma}
679 \def\tau{\mittau}
680 \def\upsilon{\mitupsilon}
681 \def\phi{
682   \bool_if:NTF \g_um_texgreek_bool {\mitvarphi}{\mitphi}
683 }
684 \def\chi{\mitchi}
685 \def\psi{\mitpsi}
686 \def\omega{\mitomega}
687 \def\varepsilon{
688   \bool_if:NTF \g_um_texgreek_bool {\mitepsilon}{\mitvarepsilon}
689 }
690 \def\vartheta{\mitvartheta}
691 \def\varkappa{\mitvarkappa}
692 \def\varphi{
693   \bool_if:NTF \g_um_texgreek_bool {\mitphi}{\mitvarphi}
694 }
695 \def\varrho{\mitvarrho}
696 \def\varpi{\mitvarpi}
697 }

```

`\um@def@numbers`

```
698 \newcommand\um@def@numbers{
699   \um@setmathcode[10]{\um@usv@num}{\um@usv@num}
700 }
```

`\um_setup_literals:` : TODO : other literal symbols

```
701 \cs_set:Nn \um_setup_literals: {
702   \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}
703   \um@setmathcode[26]{\um@usv@itLatin}{\um@usv@itLatin}
704   \um@setmathcode[26]{\um@usv@itlatin}{\um@usv@itlatin}
705   \um@setmathcode{\um@usv@ith}{\um@usv@ith}
706   \um@setmathcode[26]{\um@usv@uplatin}{\um@usv@uplatin}
707   \um@setmathcode[25]{\um@usv@upGreek}{\um@usv@upGreek}
708   \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}
709   \um@setmathcode[25]{\um@usv@itGreek}{\um@usv@itGreek}
710   \um@setmathcode[25]{\um@usv@upgreek}{\um@usv@upgreek}
711 }
```

`\um_setup_bf_literals:` TODO: other literal symbols

```
712 \cs_set:Nn \um_setup_bf_literals: {
713   \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}
714   \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}
715   \um@setmathcode[26]{\um@usv@bfitLatin}{\um@usv@bfitLatin}
716   \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}
717   \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}
718   \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}
719   \um@setmathcode[25]{\um@usv@bfitGreek}{\um@usv@bfitGreek}
720   \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}
721 }
```

`\um@def@upLatin`

```
722 \newcommand\um@def@upLatin{
723   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
724 }
```

`\um@def@itLatin`

```
725 \newcommand\um@def@itLatin{
726   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
727 }
```

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

```
728 \newcommand\um@def@itlatin{
729   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
730   \um@setmathcode{\`h,\um@usv@ith}{\um@usv@ith}
731 }
```



`\um@def@uplatin`

```
732 \newcommand\um@def@uplatin{
733   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
734   \um@setmathcode{\um@usv@ith}{`\h}
735 }
```

`\um@def@upGreek`

```
736 \newcommand\um@def@upGreek{
737   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
738   \um@setmathcode{\um@usv@varTheta,"1D6F3}{\um@usv@varTheta}
739 }
```

`\um@def@itGreek`

```
740 \newcommand\um@def@itGreek{
741   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
742   \um@setmathcode{\um@usv@varTheta}{\um@usv@itvarTheta}
743 }
```

`\um@def@upgreek`

```
744 \newcommand\um@def@upgreek{
745   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
746   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
747   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
748   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
749   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
750   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
751   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
752 }
```

`\um@def@itgreek`

```
753 \newcommand\um@def@itgreek{
754   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
755   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
756   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
757   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
758   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
759   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
760   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
761 }
```

## 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.

```

762 \cs_new:Nn \um_setup_math_alphabet:n {
763   \um_glyph_if_exist:nTF {\csname um@usv@#1latin \endcsname}{
764     \um_maybe_init_alphabet:n {#1}
765     \um_prepare_alph:n {#1}
766     \use:c {um_config_math#1:}
767   }{
768     \PackageWarning{unicode-math}{Math~ alphabet~ "#1"~ not~ found~ with~ this~ font}
769   }
770 }

771 \cs_set:Nn \um_init_alphabet:n {
772   \cs_set_eq:cN {um_setup_math#1:} \prg_do_nothing:
773 }

```

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

```

774 \prg_new_conditional:Nnn \um_glyph_if_exist:n {p,TF,T,F} {
775   \etex_iffontchar:D \um@font #1 \scan_stop: \prg_return_true: \else: \prg_return_false: \fi:
776 }

```

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

```

777 \cs_new:Nn \um_prepare_alph:n {
778   \cs_if_exist:cF {um_math#1:n} {
779     \cs_set:cpn {um_math#1:n} ##1 {
780       \begingroup \use:c {um_setup_math#1:} ##1 \endgroup
781     }
782     \cs_set_protected:cpn {math#1} {
783       \mode_if_math:F {
784         \expandafter\non@alpherr\expandafter{\csname math#1\endcsname\space}
785       }
786       \use:c {um_math#1:n}
787     }
788   }
789 }

790 \cs_new:Nn \um_setup_alphabets: {
791   \um_setup_math_alphabet:n {up}
792   \um_setup_math_alphabet:n {it}

```

```

793 \um_setup_math_alphabet: n {bb      }
794 \um_setup_math_alphabet: n {scr    }
795 \um_setup_math_alphabet: n {frac   }
796 \um_setup_math_alphabet: n {sf     }
797 \um_setup_math_alphabet: n {sfit   }
798 \um_setup_math_alphabet: n {tt     }
799 \um_setup_math_alphabet: n {bf     }
800 \um_setup_math_alphabet: n {bfup   }
801 \um_setup_math_alphabet: n {bfitt  }
802 \um_setup_math_alphabet: n {bfscr  }
803 \um_setup_math_alphabet: n {bffrak }
804 \um_setup_math_alphabet: n {bfsf   }
805 \um_setup_math_alphabet: n {bfsfup}
806 \um_setup_math_alphabet: n {bfsfitt}
807 }

```

: 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.

```

808 \cs_new:Npn \um_config_mathup: {
809   \um_set_mathalphabet_latin: Nnn{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
810   \um_set_mathalphabet_latin: Nnn{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
811   \um_set_mathalphabet_greek: Nnn{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
812   \um_set_mathalphabet_greek: Nnn{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
813   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}
814   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}
815   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}
816   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
817   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
818   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
819   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
820   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
821   \um_set_mathalphabet_char: Nnn{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
822 }

```

## 8.0.2 Italic: `\mathit`

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*  
*abcdefghijklmnopqrstuvwxyz*  
*ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ*  
*Θ*  
*αβγδεζηθικλμνξοπρστυφχψω*  
*εθϛφρω*

`\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}`  
`\mathit{abcdefghijklmnopqrstuvwxyz}`  
`\mathit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}`  
`\mathit{Θ}`  
`\mathit{αβγδεζηθικλμνξοπρστυφχψω}`

Roman:

```

823 \cs_new:Npn \um_config_mathit: {
824   \um_set_mathalphabet_latin: Nnn{\mathit}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@itLatin}
825   \um_set_mathalphabet_latin: Nnn{\mathit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@itlatin}
826   \um_set_mathalphabet_char: Nnn{\mathit}{`h, \um@usv@ith}{\um@usv@ith}

```

Greek:

```

827   \um_set_mathalphabet_greek: Nnn{\mathit}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@itGreek}
828   \um_set_mathalphabet_greek: Nnn{\mathit}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@itgreek}
829   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@itNabla}
830   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@partial, \um@usv@itpartial}{\um@usv@itpartial}
831   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@itvarTheta}
832   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@itvarepsilon}
833   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@itvartheta}
834   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@itvarkappa}
835   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@itvarphi}
836   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@itvarrho}
837   \um_set_mathalphabet_char: Nnn{\mathit}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@itvarpi}
838 }

```

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

*0123456789*  
*ABCDEFGHIJKLMNOPQRSTUVWXYZ*  
*abcdefghijklmnopqrstuvwxyz*

`\mathbb{0123456789}`  
`\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}`  
`\mathbb{abcdefghijklmnopqrstuvwxyz}`

Numbers:

```

839 \cs_new:Npn \um_config_mathbb: {
840   \um_set_mathalphabet_numbers: Nnn{\mathbb}{\um@usv@num}{\um@usv@bbnum}

```

Roman uppercase:

```

841   \um_set_mathalphabet_latin: Nnn{\mathbb}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bbLatin}
842   \um_set_mathalphabet_char: Nnn{\mathbb}{`C, "1D60A}{`"2102}
843   \um_set_mathalphabet_char: Nnn{\mathbb}{`H, "1D60F}{`"210D}
844   \um_set_mathalphabet_char: Nnn{\mathbb}{`N, "1D60F}{`"2115}

```

```

845 \um_set_mathalphabet_char: Nnn{\mathbb}{`P,"1D617}{ "2119}
846 \um_set_mathalphabet_char: Nnn{\mathbb}{`Q,"1D618}{ "211A}
847 \um_set_mathalphabet_char: Nnn{\mathbb}{`R,"1D619}{ "211D}
848 \um_set_mathalphabet_char: Nnn{\mathbb}{`Z,"1D621} { "2124}

```

Roman lowercase:

```

849 \um_set_mathalphabet_latin: Nnn{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bblatin}
850 }

```

## 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>

---

```

851 \cs_new:Npn \um_config_mathscr: {
852 \um_set_mathalphabet_latin: Nnn{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}
853 \um_set_mathalphabet_char: Nnn{\mathscr}{`B,"1D435}{ "212C}
854 \um_set_mathalphabet_char: Nnn{\mathscr}{`E,"1D438}{ "2130}
855 \um_set_mathalphabet_char: Nnn{\mathscr}{`F,"1D439}{ "2131}
856 \um_set_mathalphabet_char: Nnn{\mathscr}{`H,"1D43B}{ "210B}
857 \um_set_mathalphabet_char: Nnn{\mathscr}{`I,"1D43C}{ "2110}
858 \um_set_mathalphabet_char: Nnn{\mathscr}{`L,"1D43F}{ "2112}
859 \um_set_mathalphabet_char: Nnn{\mathscr}{`M,"1D440}{ "2133}
860 \um_set_mathalphabet_char: Nnn{\mathscr}{`R,"1D445}{ "211B}
861 \um_set_mathalphabet_latin: Nnn{\mathscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@scrlatin}
862 \um_set_mathalphabet_char: Nnn{\mathscr}{`e,"1D452}{ "212F}
863 \um_set_mathalphabet_char: Nnn{\mathscr}{`g,"1D454}{ "210A}
864 \um_set_mathalphabet_char: Nnn{\mathscr}{`o,"1D45C}{ "2134}
865 }

```

## 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*, *X*, *□*}:

```

866 \cs_new:Npn \um_config_mathfrak: {
867 \um_set_mathalphabet_latin: Nnn{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}
868 \um_set_mathalphabet_char: Nnn{\mathfrak}{`C,"1D436}{ "212D}
869 \um_set_mathalphabet_char: Nnn{\mathfrak}{`H,"1D43B}{ "210C}
870 \um_set_mathalphabet_char: Nnn{\mathfrak}{`I,"1D43C}{ "2111}
871 \um_set_mathalphabet_char: Nnn{\mathfrak}{`R,"1D445}{ "211C}
872 \um_set_mathalphabet_char: Nnn{\mathfrak}{`Z,"1D44D}{ "2128}

```

```

873 \um_set_mathalphabet_latin: Nnn{\mathfrak}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@fraklat
874 }

```

## 8.0.6 Sans serif: \mathsf

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

```

875 \cs_new:Npn \um_config_mathsf: {
876   \um_set_mathalphabet_numbers: Nnn{\mathsf}{\um@usv@num}{\um@usv@sfnun}
877   \um_set_mathalphabet_latin: Nnn{\mathsf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@sflatin}
878   \um_set_mathalphabet_latin: Nnn{\mathsf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@sflatin}
879 }

```

## 8.0.7 Sans serif italic: \mathsfit

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

```

880 \cs_new:Npn \um_config_mathsfitt: {
881   \um_set_mathalphabet_numbers: Nnn{\mathsfit}{\um@usv@num}{\um@usv@sfnun}
882   \um_set_mathalphabet_latin: Nnn{\mathsfit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@sfitlatin}
883   \um_set_mathalphabet_latin: Nnn{\mathsfit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@sfitlatin}
884 }

```

## 8.0.8 Typewriter or monospaced: \mathtt

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

```

885 \cs_new:Npn \um_config_mathtt: {
886   \um_set_mathalphabet_numbers: Nnn{\mathtt}{\um@usv@num}{\um@usv@ttnum}
887   \um_set_mathalphabet_latin: Nnn{\mathtt}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@ttlatin}
888   \um_set_mathalphabet_latin: Nnn{\mathtt}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@ttlatin}
889 }

```

## 8.1 Bold alphabets' character mappings

### 8.1.1 Bold: `\mathbf`

---

0123456789	$\mathbf{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathbf{abcdefghijklmnopqrstuvwxyz}$
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ	$\mathbf{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}$
Θ	$\mathbf{Θ}$
αβγδεζηθικλμνξοπρστυφχψω	$\mathbf{αβγδεζηθικλμνξοπρστυφχψω}$
εθκφρω	$\mathbf{εθκφρω}$

---

```

890 \cs_new:Npn \um_config_mathbf: {
891   \um_set_mathalphabet_numbers: Nnn{\mathbf}{\um@usv@num}{\um@usv@bnum}
892   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@bDigamma}
893   \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@digamma}{\um@usv@bdigamma}
894   \if@um@bfliteral
895     \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@upLatin}{\um@usv@bflatin}
896     \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@itLatin}{\um@usv@bfitLatin}
897     \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@uplatin}{\um@usv@bflatin}
898     \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@itlatin}{\um@usv@bfitlatin}
899     \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upGreek}{\um@usv@bfgreek}
900     \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@itGreek}{\um@usv@bfitGreek}
901     \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upgreek}{\um@usv@bfgreek}
902     \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@itgreek}{\um@usv@bfitgreek}
903     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
904     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}
905     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}
906     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@bDigamma}
907     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial}{\um@usv@bpartial}
908     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}
909     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}
910     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}
911     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}
912     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}
913     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}
914     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@digamma}{\um@usv@bdigamma}
915     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
916     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itNabla}{\um@usv@bfitNabla}
917     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}
918     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
919     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}
920     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
921     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}
922     \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}

```

```

923 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}
924 \else
925 \if@um@b fupLatin
926 \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfitLatin}
927 \else
928 \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfitLatin}
929 \fi
930 \if@um@b fuplatin
931 \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfitlatin}
932 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfuph}
933 \else
934 \um_set_mathalphabet_latin: Nnn{\mathbf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfitlatin}
935 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
936 \fi
937 \if@um@b fupGreek
938 \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfgreek}
939 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@b fvarTheta}
940 \else
941 \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@b fitGreek}
942 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@b fitvarTheta}
943 \fi
944 \if@um@b fupgreek
945 \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfgreek}
946 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
947 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@b fvartheta}
948 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@b fvarkappa}
949 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@b fvarphi}
950 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@b fvarrho}
951 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@b fvarpi}
952 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial, \um@usv@itpartial}{\um@usv@b fpartial}
953 \else
954 \um_set_mathalphabet_greek: Nnn{\mathbf}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@b fitgreek}
955 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
956 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@b fitvartheta}
957 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@b fitvarkappa}
958 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@b fitvarphi}
959 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@b fitvarrho}
960 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@b fitvarpi}
961 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial, \um@usv@itpartial}{\um@usv@b fitpartial}
962 \fi
963 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@b fNabla}
964 \um_set_mathalphabet_char: Nnn{\mathbf}{\um@usv@partial, \um@usv@itpartial}{\um@usv@b fpartial}
965 \fi
966 }

```



### 8.1.2 Bold Italic: `\mathbfit`

<b>0123456789</b>	<code>\$\mathbfit{0123456789}\$ \\\</code>
<b>ABCDEFGHIJKLMNOPQRSTUVWXYZ</b>	<code>\$\mathbfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
<b>abcdefghijklmnopqrstuvwxyz</b>	<code>\$\mathbfit{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>
<b>ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ</b>	<code>\$\mathbfit{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$\quad</code>
<b>αβγδεζηθικλμνξοπρστυφχψω εθκφρω</b>	<code>\$\mathbfit{αβγδεζηθικλμνξοπρστυφχψω}\$\quad</code>
	<code>\$\mathbfit{εθκφρω}\$ \\\</code>

```

967 \cs_new:Npn \um_config_mathbfit: {
968   \um_set_mathalphabet_numbers:Nnn{\mathbfit}{\um@usv@num}{\um@usv@bfnun}
969   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@upLatin,\um@usv@i tLatin}{\um@usv@b fitLatin}
970   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@uplatin,\um@usv@i tlatin}{\um@usv@b fitlatin}
971   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upGreek,\um@usv@i tGreek}{\um@usv@b fitGreek}
972   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upgreek,\um@usv@i tgreek}{\um@usv@b fitgreek}
973   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@b fLatin}{\um@usv@b fitLatin}
974   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@b flatin}{\um@usv@b fitlatin}
975   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@b fGreek}{\um@usv@b fitGreek}
976   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@b fgreek}{\um@usv@b fitgreek}
977   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varTheta,\um@usv@i tvarTheta}{\um@usv@b fit}
978   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@Nabla,\um@usv@i tNabla}{\um@usv@b fitNabla}
979   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@partial,\um@usv@i tpartial}{\um@usv@b fitpa}
980   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varepsilon,\um@usv@i tvarepsilon}{\um@usv@}
981   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@vartheta,\um@usv@i tvartheta}{\um@usv@b fit}
982   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varkappa,\um@usv@i tvarkappa}{\um@usv@b fit}
983   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varphi,\um@usv@i tvarphi}{\um@usv@b fitvarp}
984   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varrho,\um@usv@i tvarrho}{\um@usv@b fitvarr}
985   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varpi,\um@usv@i tvarpi}{\um@usv@b fitvarpi}
986 }

```

### 8.1.3 Bold Italic: `\mathbfup`

<b>0123456789</b>	<code>\$\mathbfup{0123456789}\$ \\\</code>
<b>ABCDEFGHIJKLMNOPQRSTUVWXYZ</b>	<code>\$\mathbfup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \\\</code>
<b>abcdefghijklmnopqrstuvwxyz</b>	<code>\$\mathbfup{abcdefghijklmnopqrstuvwxyz}\$ \\\</code>
<b>ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ</b>	<code>\$\mathbfup{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$\quad</code>
<b>αβγδεζηθικλμνξοπρστυφχψω εθκφρω</b>	<code>\$\mathbfup{αβγδεζηθικλμνξοπρστυφχψω}\$\quad</code>
	<code>\$\mathbfup{εθκφρω}\$ \\\</code>

```

987 \cs_new:Npn \um_config_mathbfup: {
988   \um_set_mathalphabet_numbers:Nnn{\mathbfup}{\um@usv@num}{\um@usv@bfnun}
989   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@upLatin,\um@usv@i tLatin}{\um@usv@b fLatin}
990   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@uplatin,\um@usv@i tlatin}{\um@usv@b flatin}
991   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upGreek,\um@usv@i tGreek}{\um@usv@b fGreek}

```

```

992 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfgreek}
993 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@bflatin}{\um@usv@bflatin}
994 \um_set_mathalphabet_latin: Nnn{\mathbfup}{\um@usv@bflatin}{\um@usv@bflatin}
995 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@bfgreek}{\um@usv@bfgreek}
996 \um_set_mathalphabet_greek: Nnn{\mathbfup}{\um@usv@bfgreek}{\um@usv@bfgreek}
997 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varTheta, \um@usv@itvarTheta}{\um@usv@bfvarTheta}
998 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@bfNabla}
999 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfpartial}
1000 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varepsilon, \um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
1001 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@vartheta, \um@usv@itvartheta}{\um@usv@bfvartheta}
1002 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varkappa, \um@usv@itvarkappa}{\um@usv@bfvarkappa}
1003 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varphi, \um@usv@itvarphi}{\um@usv@bfvarphi}
1004 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varrho, \um@usv@itvarrho}{\um@usv@bfvarrho}
1005 \um_set_mathalphabet_char: Nnn{\mathbfup}{\um@usv@varpi, \um@usv@itvarpi}{\um@usv@bfvarpi}
1006 }

```

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz*

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

```

1007 \cs_new:Npn \um_config_mathbffrak: {
1008   \um_set_mathalphabet_numbers: Nnn{\mathbffrak}{\um@usv@num}{\um@usv@bfnun}
1009   \um_set_mathalphabet_latin: Nnn{\mathbffrak}{\um@usv@upLatin, \um@usv@itLatin, \um@usv@frakL}
1010   \um_set_mathalphabet_latin: Nnn{\mathbffrak}{\um@usv@uplatin, \um@usv@itlatin, \um@usv@frakla}
1011 }

```

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz*

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

```

1012 \cs_new:Npn \um_config_mathbfscr: {
1013   \um_set_mathalphabet_numbers: Nnn{\mathbfscr}{\um@usv@num}{\um@usv@bfnun}
1014   \um_set_mathalphabet_latin: Nnn{\mathbfscr}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfscrL}
1015   \um_set_mathalphabet_latin: Nnn{\mathbfscr}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfscr1}
1016 }

```

### 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
$\mathbf{f}{\Theta}$ \
$\mathbf{f}{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbf{f}{\epsilon\delta\kappa\phi\rho\tau}$ \
```

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

```
1017 \cs_new:Npn \um_config_mathbf{f}{
1018   \um_set_mathalphabet_numbers:Nnn{\mathbf{f}}{\um@sv@num}{\um@sv@bfnun}
1019   \um_set_mathalphabet_latin:Nnn{\mathbf{f}}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@bfs flat
1020   \um_set_mathalphabet_latin:Nnn{\mathbf{f}}{\um@sv@uplatin,\um@sv@i tlatin}{\um@sv@bfs flat
1021   \um_set_mathalphabet_greek:Nnn{\mathbf{f}}{\um@sv@upGreek,\um@sv@i tGreek}{\um@sv@bfs fgRe
1022   \um_set_mathalphabet_greek:Nnn{\mathbf{f}}{\um@sv@upgreek,\um@sv@i tgreek}{\um@sv@bfs fgRe
```

Others:

```
1023   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varTheta,\um@sv@itvarTheta}{1D767}
1024   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@Nabla,\um@sv@i tNabla}{1D76F}
1025   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@partial,\um@sv@i tpartial}{1D789}
1026   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varepsilon,\um@sv@i tvarepsilon}{1D78A}
1027   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@vartheta,\um@sv@itvartheta}{1D78B}
1028   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varkappa,\um@sv@i tvarkappa}{1D78C}
1029   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varphi,\um@sv@i tvarphi}{1D78D}
1030   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varrho,\um@sv@i tvarrho}{1D78E}
1031   \um_set_mathalphabet_char:Nnn{\mathbf{f}}{\um@sv@varpi,\um@sv@i tvarpi}{1D78F}
1032 }
```

### 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
$\mathbf{fup}{\Theta}$ \
$\mathbf{fup}{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbf{fup}{\epsilon\delta\kappa\phi\rho\tau}$ \
```

Numbers (always upright) and letters:

```
1033 \cs_new:Npn \um_config_mathbf{fup}{
1034   \um_set_mathalphabet_numbers:Nnn{\mathbf{fup}}{\um@sv@num}{\um@sv@bfnun}
1035   \um_set_mathalphabet_latin:Nnn{\mathbf{fup}}{\um@sv@upLatin,\um@sv@i tLatin}{\um@sv@bfs fl
1036   \um_set_mathalphabet_latin:Nnn{\mathbf{fup}}{\um@sv@uplatin,\um@sv@i tlatin}{\um@sv@bfs fl
```

```

1037 \um_set_mathalphabet_greek: Nnn{\mathbfs fup}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfs fg
1038 \um_set_mathalphabet_greek: Nnn{\mathbfs fup}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfs fg

```

Others:

```

1039 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varTheta, \um@usv@itvarTheta}{1D767}
1040 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@Nabla, \um@usv@itNabla}{1D76F}
1041 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@partial, \um@usv@itpartial}{1D789}
1042 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varepsilon, \um@usv@itvarepsilon}{1D78A}
1043 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@vartheta, \um@usv@itvartheta}{1D78B}
1044 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varkappa, \um@usv@itvarkappa}{1D78C}
1045 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varphi, \um@usv@itvarphi}{1D78D}
1046 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varrho, \um@usv@itvarrho}{1D78E}
1047 \um_set_mathalphabet_char: Nnn{\mathbfs fup}{\um@usv@varpi, \um@usv@itvarpi}{1D78F}
1048 }

```

### 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
$\mathbfs fit{ }$ \
$\mathbfs fit{αβγδεζηθικλμνξοπρστυφχψω}$\quad
$\mathbfs fit{εδκφρϖ}$ \

```

```

1049 \cs_new:Npn \um_config_mathbfs fit: {
1050 \um_set_mathalphabet_numbers: Nnn{\mathbfs fit}{\um@usv@num}{\um@usv@bfnun}
1051 \um_set_mathalphabet_latin: Nnn{\mathbfs fit}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfs fi
1052 \um_set_mathalphabet_latin: Nnn{\mathbfs fit}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfs fi
1053 \um_set_mathalphabet_greek: Nnn{\mathbfs fit}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfs fi
1054 \um_set_mathalphabet_greek: Nnn{\mathbfs fit}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfs fi

```

Other symbols:

```

1055 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varTheta}{1D7A1}
1056 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@Nabla, \um@usv@itNabla}{\um@usv@bfs fitNa
1057 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@partial, \um@usv@itpartial}{\um@usv@bfs fi
1058 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varepsilon, \um@usv@itvarepsilon}{1D7C4}
1059 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@vartheta, \um@usv@itvartheta}{1D7C5}
1060 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varkappa, \um@usv@itvarkappa}{1D7C6}
1061 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varphi, \um@usv@itvarphi}{1D7C7}
1062 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varrho, \um@usv@itvarrho}{1D7C8}
1063 \um_set_mathalphabet_char: Nnn{\mathbfs fit}{\um@usv@varpi, \um@usv@itvarpi}{1D7C9}
1064 }

```

## 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  $\text{\LaTeX}$  ‘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.

```
1065 \begingroup
1066   \char_make_other:N \^
1067   \gdef\um@scancharlet#1="#2\@nil{
1068     \lowercase{
1069       \scantokens{\global\let#1=^^^^^#2}
1070     }
1071   }
1072   \gdef\um@scanactivedef"#1\@nil#2{
1073     \lowercase{
1074       \scantokens{\global\def^^^^^#1{#2}}
1075     }
1076   }
1077 \endgroup
1078 \let\unicodemathgobble\@gobble
```

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

```
1079 \begingroup
1080   \def\UnicodeMathSymbol#1#2#3#4{
1081     \um@scancharlet#2=#1\@nil
1082   }
1083   \@input{unicode-math-table.tex}
1084 \endgroup
```

## 8.3 Epilogue

Lots of little things to tidy up.

### 8.3.1 Unicode radicals

Undo the damage made to `\sqrt`:

```
1085 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}
```

### 8.3.2 Primes

$$\begin{array}{c} [x'] [x'''] [x'''''] \\ [x'] [x'''] [x'''''] \\ [x'] [x'''] [x'''''] \end{array}$$

```
\setmathfont{Cambria Math}
[$x\prime$] [$x\prime\prime\prime$]
[$x\prime\prime\prime$] [$x\prime\prime\prime\prime$]
[$x'$] [$x'''] [$x'''''] \\\sim
[$x\prime$] [$x\prime\prime\prime$] [$x\prime\prime\prime\prime$]
```

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 `sssty` feature is applied:

```
U+2032: PRIME in the ‘scriptstyle’ font: xʹ
```

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: `xʹ` and `xʹ`. 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 `x'''` vs. `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` & `\primetripel`.
- Ditto `pcount=4` & `\primequadruple`.
- If `pcount>4` or the glyph doesn’t exist, insert `pcount \primes` with `\primekern` between each.

```
1086 \muskip_new: N \g_um_primekern_muskip
1087 \muskip_gset: Nn \g_um_primekern_muskip { -\thinmuskip/2 }% arbitrary
1088 \num_new: N \l_um_primecount_num
```

```

1089 \cs_new:Nn \um_nprimes:n {
1090   \primesingle
1091   \prg_replicate:nn {#1-1} { \mskip \g_um_primekern_muskip \primesingle }
1092 }
1093 \cs_new:Nn \um_nprimes_select:n {
1094   \prg_case_int:nnn {#1}{
1095     {1} { \primesingle }
1096     {2} {
1097       \um_glyph_if_exist:NTF {"2033} {\primedouble} {\um_nprimes:n {#1}}
1098     }
1099     {3} {
1100       \um_glyph_if_exist:NTF {"2034} {\primetriple} {\um_nprimes:n {#1}}
1101     }
1102     {4} {
1103       \um_glyph_if_exist:NTF {"2057} {\primequadruple} {\um_nprimes:n {#1}}
1104     }
1105   }{
1106     \um_nprimes:n {#1}
1107   }
1108 }

```

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?

```

1109 \cs_new:Nn \um_scanprime: {
1110   \bgroup
1111   \num_zero:N \l_um_primecount_num
1112   \um_scanprime_collect:
1113 }
1114 \cs_new:Nn \um_scanprime_collect: {
1115   \num_incr:N \l_um_primecount_num
1116   \peek_charcode_remove:NTF ' {
1117     \um_scanprime_collect:
1118   }{
1119     \peek_meaning_remove:NTF \um_scanprime: {
1120       \um_scanprime_collect:
1121     }{
1122       \peek_charcode_remove:NTF ^^^^2032 {
1123         \um_scanprime_collect:
1124       }{
1125         \um_nprimes_select:n {\l_um_primecount_num}
1126         \egroup
1127       }
1128     }
1129 }

```

```

1130 }
1131 \cs_set_eq:NN \prime \um_scanprime:
1132 \group_begin:
1133   \char_make_active:N \'
1134   \char_make_active:n {"2032}
1135   \cs_gset_eq:NN ' \um_scanprime:
1136   \cs_gset_eq:NN ^^^^2032 \um_scanprime:
1137 \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`.

```

1138 \def\r@@t#1#2{
1139   \setbox\z@\hbox{$\m@th #1\sqrtsign{#2}$}
1140   \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}
1141   \raise \dimexpr(
1142     \um@fontdimen@percent{65}{\um@font}\ht\z@-
1143     \um@fontdimen@percent{65}{\um@font}\dp\z@
1144   )\relax
1145   \copy \rootbox
1146   \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}
1147   \box \z@
1148 }

```

### 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`:

```

1149 \def\operator@font{\um_setup_mathup:}
1150 \def\to{\rightarrow}
1151 \def\le{\leq}
1152 \def\ge{\geq}

```

`\mathcal`

```

1153 \def\mathcal{\mathscr}

```

`\mathrm`

```

1154 \def\mathrm{\mathup}

```

Overriding amsmath definitions:

```

1155 \AtBeginDocument{
1156   \def\@cdots{\mathinner{\cdots}}
1157 }

```



Interaction with beamer:

```
1158 \AtBeginDocument{
1159   \@ifpackageloaded{beamer}{
1160     \ifbeamer@suppressreplacements\else
1161       \PackageWarningNoLine{unicode-math}{
1162         Disabling~ beamer's~ math~ setup.^^}
1163       Please~ load~ beamer~ with~ the~ [professionalfonts]~ class~ option
1164     }
1165     \beamer@suppressreplacementstrue
1166   \fi
1167 }{}
1168 }

The end.
1169 \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://www.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));
```

```

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>/ )

```

```

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

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

```

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}/' '\

```

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

## A.1 Overview

**Maths symbol fonts** Fonts for symbols:  $\alpha$ ,  $\leq$ ,  $\rightarrow$

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

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

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

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

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

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

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

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

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

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

Delimiters and radicals use wrappers around  $\TeX$ ’s `\delimiter`/`\radical` primitives, which are re-designed in  $\XTeX$ . 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: $1/4$ 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: $4 \times$ 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 <{*test}
2 \documentclass{article}
3 \usepackage[a6paper]{geometry}
4 \usepackage{fontspec}
5 \setmainfont{FPL Neu}
6 \usepackage{unicode-math}
7 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
8 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
9 \def\upGreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ΣΤΥΦΧΨΩ}
10 \def\upgreek{αβγδεζηθικλμνξοπρςστυφχψω}
11 \def\itLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
12 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
13 \def\itGreek{ΑΒΓ ΔΕΖΗΘΙΚΛΜΝΞΟΠΡ ΣΤΥΦΧΨΩ}
14 \def\itgreek{αβγδεζηθικλμνξοπρςστυφχψω}
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{%

```

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.

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

66

```

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 </testbf>

```

# 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
\'	1133
\:::	606
\::f	606
\::n	606
\@cclvi	245
\@cdots	1156
\@elt	489–493, 496, 500, 502
\@empty	296, 297, 334, 417, 534, 541, 556, 577, 582
\@gobble	1078
\@ifnextchar	1085
\@ifpackageloaded	1159
\@ii	540, 541, 543, 545, 548, 553
\@input	351, 1083
\@marker	553, 572
\@nil	254, 398, 553, 566–569, 1067, 1072, 1081
\@preamblecmds	230
\@sqrt	1085
\@tempa	119, 162, 185, 203, 221, 307, 323, 532, 544, 568, 570, 582
\@tempb	119, 120, 162, 163, 185, 186, 203, 204, 221, 222, 532, 533, 571, 572, 577
\@tempswafalse	542
\@tempwattrue	546, 549, 574, 579, 584, 589
\@um@bfliteraltrue	158, 182
\@um@bfupGreekfalse	125, 164
\@um@bfupGreektrue	137, 149, 170, 176
\@um@bfupLatinfalse	128, 167
\@um@bfupLatintrue	140, 152, 173, 179
\@um@bfupgreekfalse	126, 138, 165, 171
\@um@bfupgreektrue	150, 177
\@um@bfuplatinfalse	129, 168
\@um@bfuplatintrue	141, 153, 174, 180
\@um@fontspec@featuretrue	298
\@um@literaltrue	157
\@um@ot@math@true	326
\@um@upGreekfalse	121
\@um@upGreektrue	133, 145
\@um@upLatinfalse	123, 135
\@um@upLatintrue	147
\@um@upNablafalse	130, 189
\@um@upNablatrue	142, 154, 187
\@um@upgreekfalse	122, 134
\@um@upgreektrue	146
\@um@uplatinfalse	124, 136, 148
\@um@uppartialfalse	127, 139, 166, 172, 207
\@um@uppartialtrue	151, 178, 205
\^	10–13, 17, 23–33, 72–74
\^	33, 1066
Numbers	
\0	23
\_	17–20, 23–32
A	
\A	24
\a	25
\addnolimits	495
\advance	30, 77
\alloc@	245
\Alpha	633
\alpha	658
\AtBeginDocument	631, 1155, 1158
\awint	491
B	
\B	853
\beamer@suppressreplacementstrue	1165
\begin	19, 66
\begingroup	251, 499, 780, 1065, 1079
\Beta	634
\beta	659
\bfitGreek	56
\bfitgreek	57
\bfitLatin	54

<code>\bfitlatin</code> .....	55	<code>\csname</code> .....	17, 18, 61–64, 248, 254, 257, 260, 292, 299, 412, 484, 763, 784
<code>\bfupgreek</code> .....	52	<b>D</b>	
<code>\bfupgreek</code> .....	53	<code>\DeclareDocumentCommand</code> .....	294
<code>\bfuplatin</code> .....	50	<code>\DeclareMathVersion</code> .....	301
<code>\bfuplatin</code> .....	51	<code>\DeclareRobustCommand</code> .....	1085
<code>\bgroup</code> .....	1110	<code>\DeclareSymbolFont</code> .....	349
<code>\bool_if:NTF</code> .....	663, 682, 688, 693	<code>\def</code> .....	7–15, 23–63, 65–117, 230, 231, 234, 245, 247, 249, 300, 335, 488, 496, 498, 500, 507, 509, 566, 568–571, 633–662, 665–681, 684–687, 690–692, 695, 696, 1074, 1080, 1138, 1149–1154, 1156
<code>\bool_new:N</code> .....	22	<code>\define@choicekey</code> .....	118, 162, 185, 203, 221, 532
<code>\bool_set_false:N</code> ...	131, 155, 159, 223	<code>\define@cmdkey</code> .....	528–531
<code>\bool_set_true:N</code> .....	143, 225	<code>\define@key</code> .....	511
<code>\box</code> .....	32, 33, 79, 80, 1147	<code>\Delta</code> .....	636
<b>C</b>		<code>\delta</code> .....	661
<code>\C</code> .....	842, 868	<code>\dimen</code> .....	29–31, 76–78
<code>\cdots</code> .....	1156	<code>\dimexpr</code> .....	232, 325, 1141
<code>\char_make_active:N</code> .....	1133	<code>\documentclass</code> .....	2, 37
<code>\char_make_active:n</code> .....	252, 1134	<code>\dp</code> .....	30, 77, 1143
<code>\char_make_other:N</code> .....	1066	<b>E</b>	
<code>\chardef</code> .....	245	<code>\E</code> .....	854
<code>\Chi</code> .....	655	<code>\e</code> .....	862
<code>\chi</code> .....	684	<code>\edef</code> .....	31, 78, 307, 343, 567, 568
<code>\cirfnint</code> .....	491	<code>\egroup</code> .....	1126
<code>\clist_map_inline:Nn</code> .....	474	<code>\else</code> ...	197, 215, 237, 240, 266, 270, 275, 280, 283, 327, 341, 377, 390, 421–425, 429, 432, 437, 443, 455, 465, 501, 514, 547, 552, 558, 576, 581, 586, 924, 927, 933, 940, 953, 1160
<code>\clist_map_inline:nn</code> ...	403, 469, 476	<code>\else:</code> .....	775
<code>\clist_map_variable:NNn</code> .....	540	<code>\encodingdefault</code> .....	350
<code>\clist_map_variable:nNn</code> .	597, 608, 614	<code>\end</code> .....	34, 81
<code>\copy</code> .....	1145	<code>\endcsname</code> ..	17, 18, 61–64, 248, 254, 257, 260, 292, 299, 412, 484, 763, 784
<code>\cs_gset:cpn</code> .....	259, 269	<code>\endgroup</code> .....	255, 505, 780, 1077, 1084
<code>\cs_gset:Npn</code> .....	272, 277	<code>\Epsilon</code> .....	637
<code>\cs_gset:Npx</code> .....	282	<code>\epsilon</code> .....	662
<code>\cs_gset_eq:NN</code> .....	1135, 1136	<code>\Eta</code> .....	639
<code>\cs_if_exist:cF</code> .....	778	<code>\eta</code> .....	666
<code>\cs_new:Nn</code> .....	358, 370, 397, 402, 407, 410, 607, 613, 622, 625, 628, 762, 777, 790, 1089, 1093, 1109, 1114	<code>\etex_iffontchar:D</code> .....	775
<code>\cs_new:Npn</code> .....	808, 823, 839, 851, 866, 875, 880, 885, 890, 967, 987, 1007, 1012, 1017, 1033, 1049	<code>\ExecuteOptionsX</code> .....	228
<code>\cs_set:cpn</code> .....	779		
<code>\cs_set:Nn</code> .....	192, 210, 416, 468, 473, 701, 712, 771		
<code>\cs_set:Npn</code> .....	606		
<code>\cs_set_eq:cN</code> .....	772		
<code>\cs_set_eq:NN</code> ..	339, 340, 346, 347, 1131		
<code>\cs_set_protected:cpn</code> .....	782		
<code>\cs_to_str:N</code> .....	269, 484		

$\backslash\exp\_args:Nnff$  . . . . . 606, 609, 616  
 $\backslash\exp\_args:No$  . . . . . 470, 477  
 $\backslash\expandafter$  . . . . .  
     248, 256, 261, 263, 267, 483, 496,  
     543, 545, 548, 553, 567, 568, 572, 784  
 $\backslash\ExplSyntaxOff$  . . . . . 1169  
 $\backslash\ExplSyntaxOn$  . . . . . 6

## F

$\backslash F$  . . . . . 855  
 $\backslash f@size$  . . . . . 299  
 $\backslash fi$  . . . . . 160, 183,  
     190, 201, 208, 219, 226, 242, 243,  
     264, 285–289, 333, 348, 380, 395,  
     422–426, 434, 439, 446, 463, 464,  
     466, 479, 503, 519, 535, 550, 551,  
     554, 560, 562, 563, 575, 580, 585,  
     590–594, 929, 936, 943, 962, 965, 1166  
 $\backslash fi:$  . . . . . 775  
 $\backslash fint$  . . . . . 491  
 $\backslash font$  . . . . . 324  
 $\backslash fontdimen$  . . . . . 232, 325, 1140, 1146

## G

$\backslash g$  . . . . . 863  
 $\backslash g@addto@macro$  . . . . . 483, 557, 559  
 $\backslash g\_um\_primekern\_muskip$  1086, 1087, 1091  
 $\backslash g\_um\_texgreek\_bool$  . 22, 131, 143,  
     155, 159, 223, 225, 663, 682, 688, 693  
 $\backslash Gamma$  . . . . . 635  
 $\backslash gamma$  . . . . . 660  
 $\backslash gdef$  . . . . . 1067, 1072  
 $\backslash ge$  . . . . . 1152  
 $\backslash geq$  . . . . . 1152  
 $\backslash glb@currsz$  . . . . . 295  
 $\backslash global$  . . . . . 253, 256, 273,  
     274, 278, 279, 284, 534, 1069, 1074  
 $\backslash group\_begin:$  . . . . . 1132  
 $\backslash group\_end:$  . . . . . 1137

## H

$\backslash H$  . . . . . 843, 856, 869  
 $\backslash h$  . . . . . 730, 734, 826  
 $\backslash hbox$  . . . . . 1139  
 $\backslash hoffset$  . . . . . 23, 70  
 $\backslash ht$  . . . . . 29, 76, 1142

## I

$\backslash I$  . . . . . 857, 870  
 $\backslash if@tempswa$  . . . . . 555  
 $\backslash if@um@bfliteral$  . . . . 15, 381, 427, 894  
 $\backslash if@um@bfupGreek$  . . . . . 16, 440, 937  
 $\backslash if@um@bfupgreek$  . . . . . 17, 447, 944  
 $\backslash if@um@bfupLatin$  . . . . . 18, 430, 925  
 $\backslash if@um@bfuplatin$  . . . . . 19, 435, 930  
 $\backslash if@um@fontspec@feature$  . . . . . 8, 512  
 $\backslash if@um@literal$  . . . . . 10, 372, 419  
 $\backslash if@um@ot@math@$  . . . . . 9  
 $\backslash if@um@upGreek$  . . . . . 11, 424  
 $\backslash if@um@upgreek$  . . . . . 12, 425  
 $\backslash if@um@upLatin$  . . . . . 13, 422  
 $\backslash if@um@uplatin$  . . . . . 14, 423  
 $\backslash if@um@upNabla$  . . . . . 20, 193  
 $\backslash if@um@uppartial$  . . . . . 21, 211  
 $\backslash ifbeamer@suppressreplacements$  . 1160  
 $\backslash ifcase$  . . . . . 120, 163, 186, 204, 222, 533  
 $\backslash ifdim$  . . . . . 325  
 $\backslash ifin@$  . . . . . 262, 268  
 $\backslash ifnum$  . . . . . 475, 573, 578, 583, 587, 588  
 $\backslash ifx$  . . . . . 235, 238,  
     250, 271, 276, 281, 334, 417, 501,  
     541, 544, 545, 548, 556, 572, 577, 582  
 $\backslash iiiint$  . . . . . 489  
 $\backslash iint$  . . . . . 489  
 $\backslash iint$  . . . . . 489  
 $\backslash in@$  . . . . . 261, 267  
 $\backslash int$  . . . . . 489  
 $\backslash intBar$  . . . . . 491  
 $\backslash intbar$  . . . . . 491  
 $\backslash intcap$  . . . . . 493  
 $\backslash intclockwise$  . . . . . 490  
 $\backslash intcup$  . . . . . 493  
 $\backslash intlarkh$  . . . . . 492  
 $\backslash intx$  . . . . . 492  
 $\backslash Iota$  . . . . . 641  
 $\backslash iota$  . . . . . 668  
 $\backslash itGreek$  . . . . . 13, 48  
 $\backslash itgreek$  . . . . . 14, 49  
 $\backslash itLatin$  . . . . . 11, 46  
 $\backslash itlatin$  . . . . . 12, 47

## K

$\backslash Kappa$  . . . . . 642  
 $\backslash kappa$  . . . . . 669

<code>\kern</code> . . . . .	1140, 1146	<code>\mathchar@type</code> . . . . .	258, 272, 274, 277, 279, 282, 284, 292, 411
<b>L</b>		<code>\mathclose</code> . . . . .	276
<code>\L</code> . . . . .	858	<code>\mathcode</code> . . . . .	253
<code>\l_um_inc_num</code> . . . . .	615, 617, 618	<code>\mathfrak</code> . . . . .	867–873
<code>\l_um_incr_num</code> . . . . .	598, 600, 602	<code>\mathgroup</code> . . . . .	245
<code>\l_um_input_num</code> . . . . .	597, 600, 608, 610, 614, 617	<code>\mathinner</code> . . . . .	1156
<code>\l_um_primecount_num</code> . . . . .	1088, 1111, 1115, 1125	<code>\mathit</code> . . . . .	824–837
<code>\l_um_script_features_tl</code> . . . . .	302, 313	<code>\mathop</code> . . . . .	250
<code>\l_um_script_font_tl</code> . . . . .	304, 312	<code>\mathopen</code> . . . . .	271, 509
<code>\l_um_sscript_features_tl</code> . . . . .	303, 317	<code>\mathord</code> . . . . .	373–376, 378, 379, 382–389, 391–394, 408
<code>\l_um_sscript_font_tl</code> . . . . .	305, 316	<code>\mathrm</code> . . . . .	<u>1154</u>
<code>\lambda</code> . . . . .	643	<code>\mathscr</code> . . . . .	852–864, 1153
<code>\lambda</code> . . . . .	670	<code>\mathsf</code> . . . . .	876–878
<code>\le</code> . . . . .	1151	<code>\mathsf fit</code> . . . . .	881–883
<code>\left</code> . . . . .	<u>508</u>	<code>\mathtt</code> . . . . .	886–888
<code>\left@primitive</code> . . . . .	508, 509	<code>\mathup</code> . . . . .	809–821, 1154
<code>\leq</code> . . . . .	1151	<code>\mddefault</code> . . . . .	350
<code>\let</code> . . . . .	50, 64, 246, 295–297, 337, 338, 344, 345, 508, 534, 1069, 1078	<code>\mitAlpha</code> . . . . .	633
<code>\linewidth</code> . . . . .	16, 60	<code>\mitalpha</code> . . . . .	658
<code>\lowercase</code> . . . . .	1068, 1073	<code>\mitBeta</code> . . . . .	634
<code>\lowint</code> . . . . .	493	<code>\mitbeta</code> . . . . .	659
<b>M</b>		<code>\mitChi</code> . . . . .	655
<code>\M</code> . . . . .	859	<code>\mitchi</code> . . . . .	684
<code>\m@th</code> . . . . .	1139	<code>\mitDelta</code> . . . . .	636
<code>\makebox</code> . . . . .	16–18, 60–64	<code>\mitdelta</code> . . . . .	661
<code>\mathaccent</code> . . . . .	281	<code>\mitEpsilon</code> . . . . .	637
<code>\mathalpha</code> . . . . .	485, 601	<code>\mitepsilon</code> . . . . .	663, 688
<code>\mathalphabet</code> . . . . .	58, 61, 62	<code>\mitEta</code> . . . . .	639
<code>\mathbb</code> . . . . .	840–849	<code>\miteta</code> . . . . .	666
<code>\mathbf</code> . . . . .	58, 891–893, 895–923, 926, 928, 931, 932, 934, 935, 938, 939, 941, 942, 945–952, 954–961, 963, 964	<code>\mitGamma</code> . . . . .	635
<code>\mathbffrac</code> . . . . .	1008–1010	<code>\mitgamma</code> . . . . .	660
<code>\mathbf fit</code> . . . . .	968–985	<code>\mitIota</code> . . . . .	641
<code>\mathbf fscr</code> . . . . .	1013–1015	<code>\mitiota</code> . . . . .	668
<code>\mathbf sf</code> . . . . .	1018–1031	<code>\mitKappa</code> . . . . .	642
<code>\mathbf s fit</code> . . . . .	1050–1063	<code>\mitkappa</code> . . . . .	669
<code>\mathbf s fup</code> . . . . .	1034–1047	<code>\mitLambda</code> . . . . .	643
<code>\mathbf fup</code> . . . . .	988–1005	<code>\mitlambda</code> . . . . .	670
<code>\mathbin</code> . . . . .	371	<code>\mitMu</code> . . . . .	644
<code>\mathcal</code> . . . . .	<u>1153</u>	<code>\mitmu</code> . . . . .	671
		<code>\mitNu</code> . . . . .	645
		<code>\mitnu</code> . . . . .	672
		<code>\mitOmega</code> . . . . .	657
		<code>\mitomega</code> . . . . .	686
		<code>\mitOmicron</code> . . . . .	647

<code>\mitomicron</code> .....	674	<code>\non@alpherr</code> .....	784
<code>\mitPhi</code> .....	654	<code>\npolint</code> .....	492
<code>\mitphi</code> .....	682, 693	<code>\Nu</code> .....	645
<code>\mitPi</code> .....	648	<code>\nu</code> .....	672
<code>\mitpi</code> .....	675	<code>\num_incr: N</code> .....	1115
<code>\mitPsi</code> .....	656	<code>\num_new: N</code> .....	1088
<code>\mitpsi</code> .....	685	<code>\num_zero: N</code> .....	1111
<code>\mitRho</code> .....	649	<code>\number</code> .....	610, 617, 618
<code>\mitrho</code> .....	676	<code>\numexpr</code> .....	578, 583, 587, 588, 600, 602, 610, 617, 618
<code>\mitSigma</code> .....	651	<b>O</b>	
<code>\mitsigma</code> .....	678	<code>\o</code> .....	864
<code>\mitTau</code> .....	652	<code>\oiint</code> .....	489
<code>\mittau</code> .....	679	<code>\oint</code> .....	489
<code>\mitTheta</code> .....	640	<code>\ointctrclockwise</code> .....	490
<code>\mittheta</code> .....	667	<code>\Omega</code> .....	657
<code>\mitUpsilon</code> .....	653	<code>\omega</code> .....	686
<code>\mitupsilon</code> .....	680	<code>\Omicron</code> .....	647
<code>\mitvarepsilon</code> .....	663, 688	<code>\omicron</code> .....	674
<code>\mitvarkappa</code> .....	691	<code>\operator@font</code> .....	1149
<code>\mitvarphi</code> .....	682, 693	<code>\or</code> .....	132, 144, 156, 169, 175, 181, 188, 206, 224
<code>\mitvarpi</code> .....	696	<b>P</b>	
<code>\mitvarrho</code> .....	695	<code>\P</code> .....	845
<code>\mitvarsigma</code> .....	677	<code>\PackageError</code> .....	515
<code>\mitvarTheta</code> .....	650	<code>\PackageInfo</code> .....	336
<code>\mitvartheta</code> .....	690	<code>\PackageWarning</code> .....	328, 768
<code>\mitXi</code> .....	646	<code>\PackageWarningNoLine</code> .....	1161
<code>\mitxi</code> .....	673	<code>\papersize</code> .....	31, 32, 78, 79
<code>\mitZeta</code> .....	638	<code>\parindent</code> .....	21, 68
<code>\mitzeta</code> .....	665	<code>\peek_charcode_remove: NTF</code> .....	1116, 1122
<code>\mode_if_math: F</code> .....	783	<code>\peek_meaning_remove: NTF</code> .....	1119
<code>\mskip</code> .....	1091	<code>\Phi</code> .....	654
<code>\Mu</code> .....	644	<code>\phi</code> .....	681
<code>\mu</code> .....	671	<code>\Pi</code> .....	648
<code>\muskip_gset: Nn</code> .....	1087	<code>\pi</code> .....	675
<code>\muskip_new: N</code> .....	1086	<code>\pointint</code> .....	492
<b>N</b>		<code>\prg_case_int: nnn</code> .....	1094
<code>\N</code> .....	844	<code>\prg_do_nothing:</code> .....	772
<code>\new@mathgroup</code> .....	245, 246	<code>\prg_new_conditional: Nnn</code> .....	774
<code>\newcommand</code> .....	291, 362, 365, 482, 495, 510, 539, 596, 632, 698, 722, 725, 728, 732, 736, 740, 744, 753	<code>\prg_replicate: nn</code> .....	1091
<code>\newcounter</code> .....	7	<code>\prg_return_false:</code> .....	775
<code>\newfam</code> .....	246	<code>\prg_return_true:</code> .....	775
<code>\newif</code> .....	8–21	<code>\prg_stepwise_variable: nnnNn</code> .....	598, 615
<code>\noexpand</code> .....	307, 502		
<code>\nolimits</code> .....	263		



<code>\prime</code> .....	1131	<code>\sigma</code> .....	678
<code>\primedouble</code> .....	1097	<code>\space</code> .....	784
<code>\primequadruple</code> .....	1103	<code>\special</code> .....	32, 79
<code>\primesingle</code> .... 408, 1090, 1091, 1095		<code>\sqint</code> .....	492
<code>\primetripel</code> .....	1100	<code>\sqrt</code> .....	507, 1085
<code>\ProcessOptionsX</code> .....	229	<code>\sqrtsign</code> .....	1085, 1139
<code>\protect</code> .....	518	<code>\stepcounter</code> .....	342
<code>\providecommand</code> .....	58	<code>\string</code> .... 254, 257, 259, 260, 567, 568	
<code>\ProvidesPackage</code> .....	1	<code>\strip@pt</code> .....	232
<code>\Psi</code> .....	656	<code>\sumint</code> .....	490
<code>\psi</code> .....	685		
<b>Q</b>			
<code>\Q</code> .....	846		
<b>R</b>			
<code>\R</code> .....	847, 860, 871		
<code>\r@@t</code> .....	<u>1138</u>		
<code>\raise</code> .....	1141		
<code>\relax</code> .....	120, 163, 186, 204, 222, 232, 250, 253, 258, 269, 271–274, 276–279, 281, 282, 284, 292, 295, 324, 325, 475, 533, 545, 548, 572, 573, 578, 583, 587, 588, 600, 602, 610, 617, 618, 1144		
<code>\removenolimits</code> .....	<u>498</u>		
<code>\RequirePackage</code> .....	3–5		
<code>\Rho</code> .....	649		
<code>\rho</code> .....	676		
<code>\rightarrow</code> .....	1150		
<code>\rootbox</code> .....	1145		
<code>\rppolint</code> .....	491		
<b>S</b>			
<code>\scan_stop:</code> .....	413, 414, 775		
<code>\scantokens</code> .....	1069, 1074		
<code>\scpolint</code> .....	492		
<code>\scriptscriptstyle</code> .....	238		
<code>\scriptstyle</code> .....	235		
<code>\setbox</code> .....	24, 32, 71, 79, 1139		
<code>\setkeys</code> .....	306		
<code>\setmainfont</code> .....	5, 40		
<code>\SetMathCode</code> .....	<u>291</u> , 404, 485, 599		
<code>\setmathfont</code> .....	20, 67, <u>294</u> , 518		
<code>\sf@size</code> .....	311, 315		
<code>\shipout</code> .....	33, 80		
<code>\Sigma</code> .....	651		
<b>T</b>			
<code>\Tau</code> .....	652		
<code>\tau</code> .....	679		
<code>\testmath</code> .....	15, 25–28, 59, 72–75		
<code>\tf@size</code> .....	310, 311		
<code>\the</code> .....	31, 78		
<code>\Theta</code> .....	640		
<code>\theta</code> .....	667		
<code>\theum@fam</code> .....	343		
<code>\thinmuskip</code> .....	1087		
<code>\tl_set:Nn</code> .....	194–196, 198–200, 212–214, 216–218, 302–305		
<code>\to</code> .....	1150		
<b>U</b>			
<code>\um@addto@mathmap</code> .....	470, 477, <u>482</u>		
<code>\um@backslash</code> .....	544, 567		
<code>\um@char@num@range</code> .....	297, 474, 556, 557, 559		
<code>\um@char@range</code> .....	296, 334, 417, 534, 537, 540		
<code>\um@def@itGreek</code> .....	424, <u>740</u>		
<code>\um@def@itgreek</code> .....	425, <u>753</u>		
<code>\um@def@itLatin</code> .....	422, <u>725</u>		
<code>\um@def@itlatin</code> .....	423, <u>728</u>		
<code>\um@def@numbers</code> .....	418, <u>698</u>		
<code>\um@def@upGreek</code> .....	424, <u>736</u>		
<code>\um@def@upgreek</code> .....	425, <u>744</u>		
<code>\um@def@upLatin</code> .....	422, <u>722</u>		
<code>\um@def@uplatin</code> .....	423, <u>732</u>		
<code>\um@firstchar</code> .....	543, 568		
<code>\um@firstof</code> .....	566–568		
<code>\um@font</code> .....	236, 239, 324, 325, 775, 1140, 1142, 1143, 1146		
<code>\um@fontdimen@percent</code> .....	<u>231</u> , 236, 239, 1142, 1143		
<code>\um@mathsymbol</code> .....	<u>247</u> , 363, 367		
<code>\um@mathsymbol@noparse</code> .....	337, 362		

<code>\um@mathsymbol@parse</code> . . . . .	344, <u>365</u>	<code>\um@usv@bfitvarkappa</code> . . . . .	
<code>\um@mversion</code> . . . . .	300, 301	. . . . .	102, 451, 459, 920, 957, 982
<code>\um@nolimits</code> . . . . .	261, <u>488</u> , 496, 504	<code>\um@usv@bfitvarphi</code> . . . . .	
<code>\um@parse@range</code> . . . . .	553, <u>569</u>	. . . . .	103, 452, 460, 921, 958, 983
<code>\um@parse@term</code> . . . . .	366, 398, <u>539</u>	<code>\um@usv@bfitvarpi</code> . . . . .	
<code>\um@radicals</code> . . . . .	267, <u>507</u>	. . . . .	105, 454, 462, 923, 960, 985
<code>\um@resolve@greek</code> . . . . .	<u>631</u>	<code>\um@usv@bfitvarrho</code> . . . . .	
<code>\um@scaled@apply</code> . . . . .	<u>234</u> , 1140, 1146	. . . . .	104, 453, 461, 922, 959, 984
<code>\um@scanactivedef</code> . . . . .	254, <u>1065</u>	<code>\um@usv@bfitvarTheta</code> . . . . .	
<code>\um@scancharlet</code> . . . . .	<u>1065</u> , 1081	. . . . .	99, 442, 445, 915, 942, 977
<code>\um@set@mathsymbol</code> . . . . .	248, <u>249</u>	<code>\um@usv@bfitvartheta</code> . . . . .	
<code>\um@setmathcode</code> . . . . .		. . . . .	101, 450, 458, 919, 956, 981
. . . . .	431, 433, 436, 438, 441,	<code>\um@usv@bflatin</code> . . . . .	48,
. . . . .	442, 444, 445, 448–454, 456–462,	. . . . .	431, 433, 713, 895, 926, 973, 989, 993
. . . . .	<u>596</u> , 699, 702–710, 713–720, 723,	<code>\um@usv@bflatin</code> . . . . .	49, 50,
. . . . .	726, 729, 730, 733, 734, 737,	. . . . .	436, 438, 714, 897, 931, 974, 990, 994
. . . . .	738, 741, 742, 745–751, 754–760	<code>\um@usv@bfnabla</code> . . . . .	
<code>\um@symfont</code> . . . . .	335, 343,	. . . . .	108, 195, 382, 391, 905, 998
. . . . .	349, 363, 367, 404, 412, 470, 477, 601	<code>\um@usv@bfnun</code> . . . . .	
<code>\um@usv@bblatin</code> . . . . .	33, 841	. . . . .	47, 891,
<code>\um@usv@bblatin</code> . . . . .	34, 849	. . . . .	968, 988, 1008, 1013, 1018, 1034, 1050
<code>\um@usv@bbnum</code> . . . . .	32, 840	<code>\um@usv@bfpartial</code> . . . . .	
<code>\um@usv@bFDigamma</code> . . . . .	81, 906	. . . . .	114, 213, 386, 393, 907, 952, 999
<code>\um@usv@bfdigamma</code> . . . . .	88, 914	<code>\um@usv@bfscrLatin</code> . . . . .	59, 1014
<code>\um@usv@bffrakLatin</code> . . . . .	57, 1009	<code>\um@usv@bfscrLatin</code> . . . . .	60, 1015
<code>\um@usv@bffraklatin</code> . . . . .	58, 1010	<code>\um@usv@bfsfGreek</code> . . . . .	65, 1021, 1037
<code>\um@usv@bfgreek</code> . . . . .	51,	<code>\um@usv@bfsfgreek</code> . . . . .	66, 1022, 1038
. . . . .	441, 444, 717, 899, 938, 975, 991, 995	<code>\um@usv@bfsfitGreek</code> . . . . .	69, 1053
<code>\um@usv@bfgreek</code> . . . . .	52,	<code>\um@usv@bfsfitgreek</code> . . . . .	70, 1054
. . . . .	448, 456, 718, 901, 945, 976, 992, 996	<code>\um@usv@bfsfitLatin</code> . . . . .	67, 1051
<code>\um@usv@bfitGreek</code> . . . . .		<code>\um@usv@bfsfitlatin</code> . . . . .	68, 1052
. . . . .	55, 441, 444, 719, 900, 941, 971, 975	<code>\um@usv@bfsfitNabla</code> . . . . .	
<code>\um@usv@bfitgreek</code> . . . . .		. . . . .	111, 200, 385, 392, 1056
. . . . .	56, 448, 456, 720, 902, 954, 972, 976	<code>\um@usv@bfsfitpartial</code> . . . . .	
<code>\um@usv@bfith</code> . . . . .	98, 903, 935	. . . . .	117, 218, 389, 394, 1057
<code>\um@usv@bfitLatin</code> . . . . .		<code>\um@usv@bfsfLatin</code> . . . . .	62, 1019, 1035
. . . . .	53, 431, 433, 715, 896, 928, 969, 973	<code>\um@usv@bfsflatin</code> . . . . .	63, 64, 1020, 1036
<code>\um@usv@bfitlatin</code> . . . . .		<code>\um@usv@bfsfNabla</code> . . . . .	110, 196, 384, 392
. . . . .	54, 436, 438, 716, 898, 934, 970, 974	<code>\um@usv@bfsfnun</code> . . . . .	61
<code>\um@usv@bfitNabla</code> . . . . .		<code>\um@usv@bfsfpartial</code> . . . . .	116, 214, 388, 394
. . . . .	109, 199, 383, 391, 916, 978	<code>\um@usv@bfsfuplatin</code> . . . . .	64
<code>\um@usv@bfitpartial</code> . . . . .		<code>\um@usv@bfuph</code> . . . . .	97, 932
. . . . .	115, 217, 387, 393, 917, 961, 979	<code>\um@usv@bfuplatin</code> . . . . .	50
<code>\um@usv@bfitvarepsilon</code> . . . . .		<code>\um@usv@bfvarepsilon</code> . . . . .	
. . . . .	100, 449, 457, 918, 955, 980	. . . . .	82, 449, 457, 908, 946, 1000
		<code>\um@usv@bfvarkappa</code> . . . . .	
		. . . . .	84, 451, 459, 910, 948, 1002

$\backslash\mathrm{um@usv@b fvarphi}$ . . . . .	. . . . 96, 751, 760, 821, 837, 923,
. . . . . 85, 452, 460, 911, 949, 1003	951, 960, 985, 1005, 1031, 1047, 1063
$\backslash\mathrm{um@usv@b fvarpi}$ . . . . .	$\backslash\mathrm{um@usv@itvarrho}$ . . . . .
. . . . . 87, 454, 462, 913, 951, 1005	. . . . 95, 750, 759, 820, 836, 922,
$\backslash\mathrm{um@usv@b fvarrho}$ . . . . .	950, 959, 984, 1004, 1030, 1046, 1062
. . . . . 86, 453, 461, 912, 950, 1004	$\backslash\mathrm{um@usv@itvarTheta}$ 90, 742, 815, 831,
$\backslash\mathrm{um@usv@b fvarTheta}$ . . . . .	915, 939, 942, 977, 997, 1023, 1039
. . . . . 80, 442, 445, 904, 939, 997	$\backslash\mathrm{um@usv@itvartheta}$ . . . . .
$\backslash\mathrm{um@usv@b fvartheta}$ . . . . .	. . . . 92, 747, 756, 817, 833, 919,
. . . . . 83, 450, 458, 909, 947, 1001	947, 956, 981, 1001, 1027, 1043, 1059
$\backslash\mathrm{um@usv@Digamma}$ . . . . . 72, 892, 906	$\backslash\mathrm{um@usv@Nabla}$ . . . . .
$\backslash\mathrm{um@usv@digamma}$ . . . . . 79, 893, 914	. . . 106, 194, 373, 378, 813, 829,
$\backslash\mathrm{um@usv@frakLatin}$ . . . . . 37, 867, 1009	905, 963, 978, 998, 1024, 1040, 1056
$\backslash\mathrm{um@usv@fraklatin}$ . . . . . 38, 873, 1010	$\backslash\mathrm{um@usv@num}$ . . . . .
$\backslash\mathrm{um@usv@itGreek}$ . . . . .	. 23, 699, 840, 876, 881, 886, 891,
. 30, 709, 737, 741, 811, 827, 900,	968, 988, 1008, 1013, 1018, 1034, 1050
938, 941, 971, 991, 1021, 1037, 1053	$\backslash\mathrm{um@usv@partial}$ . . . . . 112,
$\backslash\mathrm{um@usv@itgreek}$ . . . . .	212, 375, 379, 814, 830, 907, 952,
. . . . 31, 745, 754, 812, 828, 902,	961, 964, 979, 999, 1025, 1041, 1057
945, 954, 972, 992, 1022, 1038, 1054	$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 35, 852
$\backslash\mathrm{um@usv@ith}$ . . . . .	$\backslash\mathrm{um@usv@scrlatin}$ . . . . . 36, 861
. 89, 705, 730, 734, 826, 903, 932, 935	$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 42, 882
$\backslash\mathrm{um@usv@itLatin}$ . . . . . 26,	$\backslash\mathrm{um@usv@sfitlatin}$ . . . . . 43, 883
703, 723, 726, 809, 824, 841, 852,	$\backslash\mathrm{um@usv@sflatin}$ . . . . . 40, 877
867, 877, 882, 887, 896, 926, 928,	$\backslash\mathrm{um@usv@sflatin}$ . . . . . 41, 878
969, 989, 1009, 1014, 1019, 1035, 1051	$\backslash\mathrm{um@usv@sfnun}$ . . . . . 39, 876, 881
$\backslash\mathrm{um@usv@itlatin}$ . . . . . 27,	$\backslash\mathrm{um@usv@ttLatin}$ . . . . . 45, 887
704, 729, 733, 810, 825, 849, 861,	$\backslash\mathrm{um@usv@ttlatin}$ . . . . . 46, 888
873, 878, 883, 888, 898, 931, 934,	$\backslash\mathrm{um@usv@ttnum}$ . . . . . 44, 886
970, 990, 1010, 1015, 1020, 1036, 1052	$\backslash\mathrm{um@usv@upGreek}$ . . . . .
$\backslash\mathrm{um@usv@itNabla}$ . . . . .	. 28, 707, 737, 741, 811, 827, 899,
. . . 107, 198, 374, 378, 813, 829,	938, 941, 971, 991, 1021, 1037, 1053
916, 963, 978, 998, 1024, 1040, 1056	$\backslash\mathrm{um@usv@upgreek}$ . . . . .
$\backslash\mathrm{um@usv@itpartial}$ . . . . . 113,	. 29, 710, 745, 754, 812, 828, 901,
216, 376, 379, 814, 830, 917, 952,	945, 954, 972, 992, 1022, 1038, 1054
961, 964, 979, 999, 1025, 1041, 1057	$\backslash\mathrm{um@usv@upLatin}$ . . . . . 24,
$\backslash\mathrm{um@usv@itvarepsilon}$ . . . . .	702, 723, 726, 809, 824, 841, 852,
. . . . 91, 746, 755, 816, 832, 918,	867, 877, 882, 887, 895, 926, 928,
946, 955, 980, 1000, 1026, 1042, 1058	969, 989, 1009, 1014, 1019, 1035, 1051
$\backslash\mathrm{um@usv@itvarkappa}$ . . . . .	$\backslash\mathrm{um@usv@uplatin}$ . . . . . 25,
. . . . 93, 748, 757, 818, 834, 920,	706, 729, 733, 810, 825, 849, 861,
948, 957, 982, 1002, 1028, 1044, 1060	873, 878, 883, 888, 897, 931, 934,
$\backslash\mathrm{um@usv@itvarphi}$ . . . . .	970, 990, 1010, 1015, 1020, 1036, 1052
. . . . 94, 749, 758, 819, 835, 921,	$\backslash\mathrm{um@usv@varepsilon}$ . . . . .
949, 958, 983, 1003, 1029, 1045, 1061	. . . . 73, 746, 755, 816, 832, 908,
$\backslash\mathrm{um@usv@itvarpi}$ . . . . .	946, 955, 980, 1000, 1026, 1042, 1058

<code>\um@usv@varkappa</code> .....	<code>\um_init_alphabet:n</code> .....
..... 75, 748, 757, 818, 834, 910, 948, 957, 982, 1002, 1028, 1044, 1060	340, 771
<code>\um@usv@varphi</code> .....	<code>\um_make_mathactive:nNn</code> .... 408, <u>410</u>
..... 76, 749, 758, 819, 835, 911, 949, 958, 983, 1003, 1029, 1045, 1061	<code>\um_mathmap:Nnn</code> .... 338, 345, 609, 616
<code>\um@usv@varpi</code> .....	<code>\um_mathmap_noparse:Nnn</code> .... 338, 468
..... 78, 751, 760, 821, 837, 913, 951, 960, 985, 1005, 1031, 1047, 1063	<code>\um_mathmap_parse:Nnn</code> .....
<code>\um@usv@varrho</code> .....	345, <u>473</u>
..... 77, 750, 759, 820, 836, 912, 950, 959, 984, 1004, 1030, 1046, 1062	<code>\um_maybe_init_alphabet:n</code> 340, 347, 764
<code>\um@usv@varTheta</code> .....	<code>\um_Nabla_up_or_it_usv</code> .. 194, 198, 378
..... 71, 708, 738, 742, 815, 831, 904, 939, 942, 977, 997, 1023, 1039, 1055	<code>\um_nprimes:n</code> 1089, 1097, 1100, 1103, 1106
<code>\um@usv@vartheta</code> .....	<code>\um_nprimes_select:n</code> .....
..... 74, 747, 756, 817, 833, 909, 947, 956, 981, 1001, 1027, 1043, 1059	1093, 1125
<code>\um@zf@feature</code> .....	<code>\um_partial_up_or_it_usv</code> 212, 216, 379
..... <u>510</u> , 522, 525	<code>\um_prepare_alph:n</code> .....
<code>\um_bfNabla_up_or_it_usv</code> .....	765, <u>777</u>
..... 195, 199, 391, 963	<code>\um_remap_symbol:nnn</code> . 339, 346, 371, 373–376, 378, 379, 382–389, 391–394
<code>\um_bfpartial_up_or_it_usv</code> .....	<code>\um_remap_symbol_noparse:nnn</code> ....
..... 213, 217, 393, 964	..... 339, 399, 402
<code>\um_bfsfNabla_up_or_it_usv</code> .....	<code>\um_remap_symbol_parse:nnn</code> . 346, 397
..... 196, 200, 392	<code>\um_remap_symbols:</code> .....
<code>\um_bfsfpartial_up_or_it_usv</code> .....	353, <u>370</u>
..... 214, 218, 394	<code>\um_scanprime:</code> .....
<code>\um_config_mathbb:</code> .....	..... 1109, 1119, 1131, 1135, 1136
..... 839	<code>\um_scanprime_collect:</code> .....
<code>\um_config_mathbf:</code> .....	..... 1112, 1114, 1117, 1120, 1123
..... 890	<code>\um_set_mathalph_range:Nnn</code> .....
<code>\um_config_mathbffrak:</code> .....	613
..... 1007	<code>\um_set_mathalph_range:nNnn</code> ....
<code>\um_config_mathbfrit:</code> .....	..... 613, 623, 626, 629
..... 967	<code>\um_set_mathalphabet_char:Nnn</code> ...
<code>\um_config_mathbfscr:</code> .....	... 607, 813–821, 826, 829–837, 842–848, 853–860, 862–864, 868–872, 892, 893, 903–923, 932, 935, 939, 942, 946–952, 955–961, 963, 964, 977–985, 997–1005, 1023–1031, 1039–1047, 1055–1063
..... 1012	<code>\um_set_mathalphabet_char:Nnnn</code> ..
<code>\um_config_mathbfsf:</code> .....	606
..... 1017	<code>\um_set_mathalphabet_greek:Nnn</code> ..
<code>\um_config_mathbfsrit:</code> .....	..... 628, 811, 812, 827, 828, 899–902, 938, 941, 945, 954, 971, 972, 975, 976, 991, 992, 995, 996, 1021, 1022, 1037, 1038, 1053, 1054
..... 1049	<code>\um_set_mathalphabet_latin:Nnn</code> ..
<code>\um_config_mathbfsfup:</code> .....	..... 625, 809, 810, 824, 825, 841, 849, 852, 861, 867, 873, 877, 878, 882, 883, 887, 888, 895–898, 926, 928, 931, 934, 969, 970, 973, 974, 989, 990, 993, 994, 1009, 1010, 1014, 1015, 1019, 1020, 1035, 1036, 1051, 1052
..... 1033	
<code>\um_config_mathbfup:</code> .....	
..... 987	
<code>\um_config_mathfrak:</code> .....	
..... 866	
<code>\um_config_mathit:</code> .....	
..... 823	
<code>\um_config_mathscr:</code> .....	
..... 851	
<code>\um_config_mathsf:</code> .....	
..... 875	
<code>\um_config_mathsfrit:</code> .....	
..... 880	
<code>\um_config_mathstt:</code> .....	
..... 885	
<code>\um_config_mathup:</code> .....	
..... 808	
<code>\um_glyph_if_exist:n</code> .....	
..... 774	
<code>\um_glyph_if_exist:nTF</code> .....	
..... 763, <u>774</u> , 1097, 1100, 1103	

<code>\um_set_mathalphabet_numbers: Nnn</code>	<code>\varphi</code> . . . . . 692
. . . 622, 840, 876, 881, 886, 891,	<code>\varpi</code> . . . . . 696
968, 988, 1008, 1013, 1018, 1034, 1050	<code>\varrho</code> . . . . . 695
<code>\um_setup_alphabets: . . . . . 356, 790</code>	<code>\varsigma</code> . . . . . 677
<code>\um_setup_alphanum: . . . . . 355, 416</code>	<code>\varTheta</code> . . . . . 650
<code>\um_setup_bf_literals: . . . . . 428, 712</code>	<code>\vartheta</code> . . . . . 690
<code>\um_setup_literals: . . . . . 420, 701</code>	<code>\vbox</code> . . . . . 24, 32, 71, 79
<code>\um_setup_math_alphabet: n 762, 791–806</code>	<code>\voffset</code> . . . . . 22, 69
<code>\um_setup_mathactives: . . . . . 354, 407</code>	
<code>\um_setup_mathup: . . . . . 1149</code>	<b>W</b>
<code>\um_setup_nabla: . . . . . 192, 359</code>	<code>\wd</code> . . . . . 31, 78
<code>\um_setup_partial: . . . . . 210, 360</code>	
<code>\um_setup_shapes: . . . . . 352, 358</code>	<b>X</b>
<code>\unicodemathgobble . . . . . 1078</code>	<code>\xdef</code> . . . . . 504, 537
<code>\UnicodeMathSymbol . . . . . 337, 344, 1080</code>	<code>\XeTeXdelcode</code> . . . . . 273, 278
<code>\unless . . . . . 541</code>	<code>\XeTeXdelimiter</code> . . . . . 272, 277
<code>\updefault . . . . . 350</code>	<code>\XeTeXmathaccent</code> . . . . . 282
<code>\upGreek . . . . . 9, 44</code>	<code>\XeTeXmathchardef</code> . . . . . 256, 411
<code>\upgreek . . . . . 10, 45</code>	<code>\XeTeXmathcode</code> . . . . . 274, 279, 284, 292
<code>\upint . . . . . 493</code>	<code>\XeTeXmathcodenum</code> . . . . . 414
<code>\upLatin . . . . . 7, 42</code>	<code>\XeTeXradical</code> . . . . . 269
<code>\uplatin . . . . . 8, 43</code>	<code>\Xi</code> . . . . . 646
<code>\Upsilon . . . . . 653</code>	<code>\xi</code> . . . . . 673
<code>\upsilon . . . . . 680</code>	<code>\XKV@rm</code> . . . . . 320
<code>\use: c . . . . . 766, 780, 786</code>	
<code>\use_none: n . . . . . 347</code>	<b>Z</b>
<code>\usepackage . . . . . 3, 4, 6, 38, 39, 41</code>	<code>\Z</code> . . . . . 848, 872
	<code>\z@</code> . . . . . 1139, 1142, 1143, 1147
<b>V</b>	<code>\Zeta</code> . . . . . 638
<code>\varepsilon . . . . . 687</code>	<code>\zeta</code> . . . . . 665
<code>\varkappa . . . . . 691</code>	<code>\zf@family</code> . . . . . 350
<code>\varointclockwise . . . . . 490</code>	<code>\zf@fontspec</code> . . . . . 307
	<code>\zf@update@ff</code> . . . . . 523, 526