

# 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>1</b>	<b>5.2</b>	Overcoming <code>\@on-</code> <code>lypreamble</code>	13
<b>2</b>	<b>Specification</b>	<b>1</b>	<b>5.3</b>	Other things	13
2.1	Using multiple fonts	2	<b>6</b>	<b>Fundamentals</b>	<b>14</b>
2.2	Script and scriptscript fonts/features	2	6.1	Enlarging the number of maths families	14
<b>3</b>	<b>Maths input</b>	<b>2</b>	6.2	<code>\DeclareMathSymbol</code> for unicode ranges	14
3.1	Miscellanea	3	6.3	The main <code>\setmathfont</code> macro	16
<b>4</b>	<b>Package options</b>	<b>3</b>	6.4	(Big) operators	23
4.1	Math ‘style’	3	6.5	Radicals	27
4.2	Bold switching	4	6.6	Delimiters	27
4.3	Other upright vs. italic symbols	5	6.7	Maths accents	30
<b>I</b>	<b>The unicode-math pack- age</b>	<b>7</b>	<b>7</b>	<b>Font features</b>	<b>31</b>
<b>5</b>	<b>Things we need</b>	<b>7</b>	7.1	OpenType maths font features	31
5.1	Package options	10	7.2	Script and scriptscript font options	32
			7.3	Range processing	32

<b>II Maths alphabets mapping definitions</b>	<b>39</b>	<b>the NFSS</b>	<b>55</b>
7.4 Bold alphabets' character mappings	44	A.1 Overview	55
7.5 Definitions of the math symbols	49	<b>IV X<sub>Y</sub>TeX math font dimensions</b>	<b>57</b>
7.6 Epilogue	50	<b>V Some manner of unit testing</b>	<b>61</b>
<b>III stix table data extraction</b>	<b>54</b>	<b>B The regular weight alphabets</b>	<b>62</b>
<b>A Documenting maths support in</b>		<b>C The bold alphabets</b>	<b>63</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 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⟩}
```

Instances above of

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

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

## 2.1 Using multiple fonts

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

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

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

## 2.2 Script and scriptscript fonts/features

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

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

## 3 Maths input

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

: TODO : describe alphabet inputs

## 3.1 Miscellanea

### 3.1.1 Primes

Primes ( $x'$ ) may be input in several ways. You may use any combination of `ascii` straight quote (`'`), unicode prime (`'`), 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	
	$(a, z, B, X)$	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=ISO</code>	$(a, z, B, X)$	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=TeX</code>	$(a, z, B, X)$	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=French</code>	$(a, z, B, X)$	$(\alpha, \beta, \Gamma, \Xi)$

this; to follow from the example: `\mathup{g}`. Maths alphabets commands such as `\mathup` are detailed later.

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

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

## 4.2 Bold switching

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

Bold Greek letters have simply been bold variant glyphs of their regular weight, as in  $\xi = (\xi_r, \xi_\varphi, \xi_\theta)$ . Confusingly, the syntax in  $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$  has been different for these two examples: `\mathbf{M}` in the former (‘ $\mathbf{M}$ ’), and `\bm` (or `\boldsymbol`, deprecated) in the latter (‘ $\xi$ ’).

In `unicode-math`, the `\mathbf{M}` 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	
	$(a, z, B, X)$	$(\mathbb{A}, \mathbb{B}, \Gamma, \Xi)$
<code>bold-style=ISO</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=TeX</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=French</code>	$(\mathbf{a}, \mathbf{z}, \mathbf{B}, \mathbf{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$

Table 3: The various forms of nabla.

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

Note that `bold-style` is independent of `math-style`, although if the former is not specified then sensible defaults are chosen based on the latter.

The `bold-style` options' effects are shown in brief in table 2. Table ?? on page ?? shows every character under the effect of this package option.

### 4.3 Other upright vs. italic symbols

**Nabla** The symbol  $\nabla$  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`; `\mathit` and `\mathup` can be used to force one way or the other.

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

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

**Epsilon:  $\varepsilon$  vs.  $\epsilon$**   $\text{\TeX}$  defines `\epsilon` to look like  $\epsilon$  and `\varepsilon` to look like  $\varepsilon$ . 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  $\epsilon$ . This creates a problem.

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρρςστυφφχψω$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρρςστυφφχψω$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρρςστυφφχψω$

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

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρςστυφχψωεθκφρϖ$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρςστυφχψωεθκφρϖ$

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

$ABCDEFGHIJKLMNOPQRSTUVWXYZ$   
 $abcdefghijklmnopqrstuvwxyz$   
 $ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ$   
 $αβγδεζηθικλμνξοπρςστυφχψωεθκφρϖ$

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

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

People who use unicode input won't want their glyphs transforming; T<sub>E</sub>X users will be confused that what they think as 'normal epsilon' is actual the 'variant epsilon'.

:TODO: package option

**Phi:**  $\varphi$  vs.  $\phi$  The same problem for epsilon also exists for phi.

:TODO: package option

## 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 LATEX3 — 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 and partial:

```
20 \newif\if@um@upNabla
21 \newif\if@um@uppartial
22 \bool_new:N \g_um_vargreek_bool
```

## Programming niceties

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

## Shortcuts

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

### 5.0.1 Alphabet unicode positions

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

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

---

<sup>1</sup>'u.s.v.' stands for 'unicode scalar value'.

Bold:

```
52 \def\um@usv@bfnum{"1D7CE}  
53 \def\um@usv@bfLatin{"1D400}  
54 \def\um@usv@bfLatin{"1D41A}  
55 \let\um@usv@bfuplatin\um@usv@bfLatin  
56 \def\um@usv@bfGreek{"1D6A8}  
57 \def\um@usv@bfGreek{"1D6C2}  
58 \def\um@usv@bfItLatin{"1D468}  
59 \def\um@usv@bfItLatin{"1D482}  
60 \def\um@usv@bfItGreek{"1D71C}  
61 \def\um@usv@bfItGreek{"1D736}  
62 \def\um@usv@bffrakLatin{"1D56C}  
63 \def\um@usv@bffrakLatin{"1D586}  
64 \def\um@usv@bfscrLatin{"1D4D0}  
65 \def\um@usv@bfscrLatin{"1D4EA}  
66 \def\um@usv@bfsfnum{"1D7EC}  
67 \def\um@usv@bfsfLatin{"1D5D4}  
68 \def\um@usv@bfsfLatin{"1D5EE}  
69 \def\um@usv@bfsfGreek{"1D756}  
70 \def\um@usv@bfsfGreek{"1D770}  
71 \def\um@usv@bfsfitLatin{"1D63C}  
72 \def\um@usv@bfsfitLatin{"1D656}  
73 \def\um@usv@bfsfitGreek{"1D790}  
74 \def\um@usv@bfsfitGreek{"1D7AA}
```

Greek variants:

```
75 \def\um@usv@varTheta{"3F4}  
76 \def\um@usv@Digamma{"3DC}  
77 \def\um@usv@varepsilon{"3F5}  
78 \def\um@usv@vartheta{"3D1}  
79 \def\um@usv@varkappa{"3F0}  
80 \def\um@usv@varphi{"3D5}  
81 \def\um@usv@varrho{"3F1}  
82 \def\um@usv@varpi{"3D6}  
83 \def\um@usv@digamma{"3DD}
```

Bold:

```
84 \def\um@usv@bfvarTheta{"1D6B9}  
85 \def\um@usv@bfDigamma{"1D7CA}  
86 \def\um@usv@bfvarepsilon{"1D6DC}  
87 \def\um@usv@bfvartheta{"1D6DD}  
88 \def\um@usv@bfvarkappa{"1D6DE}  
89 \def\um@usv@bfvarphi{"1D6DF}  
90 \def\um@usv@bfvarrho{"1D6E0}  
91 \def\um@usv@bfvarpi{"1D6E1}  
92 \def\um@usv@bfdigamma{"1D7CB}
```

Italic Greek variants:

```

93 \def\um@usv@ith{"210E}
94 \def\um@usv@itvarTheta{"1D6F3}
95 \def\um@usv@itvarepsilon{"1D716}
96 \def\um@usv@itvartheta{"1D717}
97 \def\um@usv@itvarkappa{"1D718}
98 \def\um@usv@itvarphi{"1D719}
99 \def\um@usv@itvarrho{"1D71A}
100 \def\um@usv@itvarpi{"1D71B}

```

**Bold:**

```

101 \def\um@usv@bfuph{"1D421}
102 \def\um@usv@bfith{"1D489}
103 \def\um@usv@bfitvarTheta{"1D72D}
104 \def\um@usv@bfitvarepsilon{"1D750}
105 \def\um@usv@bfitvartheta{"1D751}
106 \def\um@usv@bfitvarkappa{"1D752}
107 \def\um@usv@bfitvarphi{"1D753}
108 \def\um@usv@bfitvarrho{"1D754}
109 \def\um@usv@bfitvarpi{"1D755}

```

**Nabla:**

```

110 \def\um@usv@Nabla{"2207}
111 \def\um@usv@itNabla{"1D6FB}
112 \def\um@usv@bfNabla{"1D6C1}
113 \def\um@usv@bfitNabla{"1D735}
114 \def\um@usv@bfsfNabla{"1D76F}
115 \def\um@usv@bfsfitNabla{"1D7A9}

```

**Partial:**

```

116 \def\um@usv@partial{"2202}
117 \def\um@usv@itpartial{"1D715}
118 \def\um@usv@bfpartial{"1D6DB}
119 \def\um@usv@bfitpartial{"1D74F}
120 \def\um@usv@bfsfpartial{"1D789}
121 \def\um@usv@bfsfitpartial{"1D7C3}

```

## 5.1 Package options

xkeyval's package support is used here.

### **math-style**

```

122 \define@choicekey*{unicode-math.sty}
123   {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{
124   \ifcase\@tempb\relax
125     \@um@upGreekfalse
126     \@um@upgreekfalse
127     \@um@upLatinfalse

```

```

128 \um@uplatinfalse
129 \um@bfupGreekfalse
130 \um@bfupgreekfalse
131 \um@bfupLatinfalse
132 \um@bfuplatinfalse
133 \um@upNablafalse
134 \um@uppartialfalse
135 \bool_set_false:N \g_um_vargreek_bool
136 \or
137 \um@upGreektrue
138 \um@upgreekfalse
139 \um@upLatinfalse
140 \um@uplatinfalse
141 \um@bfupGreektrue
142 \um@bfupgreekfalse
143 \um@bfupLatintrue
144 \um@bfuplatintrue
145 \um@upNabltrue
146 \um@uppartialtrue
147 \bool_set_true:N \g_um_vargreek_bool
148 \or
149 \um@upGreektrue
150 \um@upgreektrue
151 \um@upLatintrue
152 \um@uplatinfalse
153 \um@bfupGreektrue
154 \um@bfupgreektrue
155 \um@bfupLatintrue
156 \um@bfuplatintrue
157 \um@upNabltrue
158 \um@uppartialtrue
159 \bool_set_false:N \g_um_vargreek_bool
160 \or
161 \um@literaltrue
162 \um@bfliteraltrue
163 \bool_set_false:N \g_um_vargreek_bool
164 \fi
165 }

```

### **bold-style**

```

166 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,french,literal}{
167 \ifcase\@tempb\relax
168 \um@bfupGreekfalse
169 \um@bfupgreekfalse
170 \um@bfupLatinfalse
171 \um@bfuplatinfalse

```

```

172 \or
173 \um@bfupGreektrue
174 \um@bfupgreekfalse
175 \um@bfupLatintrue
176 \um@bfuplatintrue
177 \or
178 \um@bfupGreektrue
179 \um@bfupgreektrue
180 \um@bfupLatintrue
181 \um@bfuplatintrue
182 \or
183 \um@bfliteraltrue
184 \fi
185 }

```

### Symbol obliqueness

```

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

```

```

216 \else
217   \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@itpartial }
218   \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfitpartial }
219   \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfitpartial }
220 \fi
221 }

```

## Epsilon and phi shapes

```

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

229 \ExecuteOptionsX{math-style=TeX}
230 \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.

```

231 \def\@preamblecmds{}

```

## 5.3 Other things

`\um@fontdimen@percent` #1 : Font dimen number  
`\fontdimens` 10, 11, and 65 aren't actually dimensions, they're percentage values given in units of sp. This macro takes a font dimension number and outputs the decimal value of the associated parameter.

---

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

---

```

232 \def\um@fontdimen@percent#1#2{
233   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax
234 }

```

`\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.

```

235 \def\um@scaled@apply#1#2#3{
236   \ifx#1\scriptstyle
237     #2\um@fontdimen@percent{10}\um@font#3
238   \else
239     \ifx#1\scriptscriptstyle
240       #2\um@fontdimen@percent{11}\um@font#3
241     \else
242       #2#3%
243     \fi
244   \fi
245 }
```

## 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

```

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

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

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

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

```
251 \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`.

```
252 \begingroup
253 \char_make_active:n {#4}
254 \global\mathcode#4="8000\relax
255 \um@scanactivedef #4 \@nil { \csname\string#2@op\endcsname }
256 \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`.

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

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

```
260 \cs_gset:cpn { \string#2 @op } {
261 \csname\string#2@sym\endcsname
262 \expandafter\in@\expandafter#2\expandafter{\um@nolimits}
263 \ifin@
264 \expandafter\nolimits
265 \fi
266 }
```

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

```
267 \else
```

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

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



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

```

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

```

### Accents

```

282     \ifx\mathaccent#3\relax
283     \cs_gset:Npx #2 {\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}
284 \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.

```

285     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
286     \fi
287     \fi
288     \fi
289     \fi
290     \fi
291 }

```

`\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.

```

292 \newcommand\SetMathCode[4]{
293   \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax
294 }

```

---

A

```

\zf@fontspec{}\Cambria Math}
\let\glb@currsiz\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:

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

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

---

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

---

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

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

---

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

`\setmathfont` [#1]: font features  
 #2 : font name

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

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

```
296 \let\glb@currsizel\relax
```

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

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

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

- Tell fontspec that maths font features are actually allowed.

```
299 \@um@fontspec@featuretrue
```

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

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

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

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

```
302 \DeclareMathVersion{\um@mversion}
```

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

```

303 \tl_set:Nn \l_um_script_features_tl {ScriptStyle}
304 \tl_set:Nn \l_um_sscript_features_tl {ScriptScriptStyle}
305 \tl_set:Nn \l_um_script_font_tl      {#2}
306 \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.

```

307 \setkeys*[um]{options}{#1}
308 \edef\@tempa{\noexpand\zf@fontspec{
309     Script = Math,
310     SizeFeatures = {
311         {Size = \tf@size-} ,
312         {Size = \sf@size-\tf@size ,
313         Font = \l_um_script_font_tl ,
314         \l_um_script_features_tl
315         } ,
316         {Size = -\sf@size ,
317         Font = \l_um_sscript_font_tl ,
318         \l_um_sscript_features_tl
319         }
320     },
321     \XKV@rm
322 }{#2}
323 }
324 \@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`:

```

325 \font\um@font="#2"\relax
326 \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt\relax
327 \um@ot@math@true
328 \else
329 \um@PackageWarning{
330     The~ font~ '#2' ~is~ not~ a~ valid~ OpenType~ maths~ font.~
331     Some~ maths~ features~ will~ not~ be~ available~ or~ behave~
332     in~ a~ substandard~ manner.
333 }
334 \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

```

335 \ifx\um@char@range\@empty
336   \def\um@symfont{\um@allsym}
337   \um@PackageInfo{Defining~ the~ default~ maths~ font~ as~ '#2'}
338   \let \UnicodeMathSymbol \um@mathsymbol@noparse
339   \let \um_mathmap:Nnn \um_mathmap_noparse:Nnn
340   \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_noparse:nnn
341   \cs_set_eq:NN \um_maybe_init_alphabet:n \um_init_alphabet:n
342 \else
343   \stepcounter{um@fam}
344   \edef\um@symfont{\um@fam\theum@fam}
345   \let \UnicodeMathSymbol \um@mathsymbol@parse
346   \let \um_mathmap:Nnn \um_mathmap_parse:Nnn
347   \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_parse:nnn
348   \cs_set_eq:NN \um_maybe_init_alphabet:n \use_none:n
349 \fi

```

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

```

350 \DeclareSymbolFont{\um@symfont}
351   {\encodingdefault}{\zf@family}{\mddefault}{\updefault}

```

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

```

352 \@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.)

```

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

```

End of the `\setmathfont` macro.

```

358 }

359 \cs_new:Nn \um_setup_shapes: {
360   \um_setup_nabla:
361   \um_setup_partial:
362 }

```

### 6.3.1 Functions for setting up symbols with mathcodes

`\um@mathsymbol@noparse`

```
363 \newcommand\um@mathsymbol@noparse[4]{
364   \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
365 }
```

`\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.

```
366 \newcommand\um@mathsymbol@parse[4]{
367   \um@parse@term{#1}{#2}{#3}{
368     %\um@PackageInfo{Defining \string#2 as mathchar #1}
369     \um@mathsymbol{#2}{#3}{\um@symfont}{#1}
370   }
371 }
```

`\um_remap_symbols:` This function is used to define the mathcodes for those chars which should be mapped to a different glyph than themselves. The only example I have right now is ascii hyphen to real minus.

```
372 \cs_new:Nn \um_remap_symbols: {
373   \um_remap_symbol:nnn{"2D}{\mathbin}{"02212}% hyphen to minus
374 }
```

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

```
375 \cs_new:Nn \um_remap_symbol_parse:nnn {
376   \um@parse@term {#3} {\@nil} {#2} {
377     \um_remap_symbol_noparse:nnn {#1} {#2} {#3}
378   }
379 }
380 \cs_new:Nn \um_remap_symbol_noparse:nnn {
381   \SetMathCode {#1} {#2} {\um@symfont} {#3}
382 }
```

### 6.3.2 Active math characters

`\um_setup_mathactives:`

```
383 \cs_new:Nn \um_setup_mathactives: {
384   \um_make_mathactive:nnn {"2032} \primesingle \mathord
385 }
```

`\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!

```
386 \cs_new:Nn \um_make_mathactive:nnn {
387   \XeTeXmathchardef #2 = "\mathchar@type #3
388   \csname sym\um@symfont\endcsname
```

```

389             #1 \scan_stop:
390 \XeTeXmathcodenum #1 = "1FFFFF \scan_stop:
391 }

```

### 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`.

```

392 \cs_set:Nn \um_setup_alphanum: {
393   \ifx\um@char@range\@empty
394     \um@def@numbers

```

#### Normal weight

```

395   \ifum@literal
396     \um_setup_literals:
397   \else
398     \ifum@upLatin\um@def@upLatin\else\um@def@itLatin\fi
399     \ifum@uplatin\um@def@uplatin\else\um@def@itlatin\fi
400     \ifum@upGreek\um@def@upGreek\else\um@def@itGreek\fi
401     \ifum@upgreek\um@def@upgreek\else\um@def@itgreek\fi
402     \um@setmathcode{\um@usv@Nabla,\um@usv@itNabla}{\um_Nabla_up_or_it_usv}
403     \um@setmathcode{\um@usv@partial,\um@usv@itpartial}{\um_partial_up_or_it_usv}
404   \fi

```

#### Bold

```

405   \ifum@bfliteral
406     \um_setup_bf_literals:
407   \else
408     \ifum@bfupLatin
409     \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfitLatin}{\um@usv@bfLatin}
410     \else
411     \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfitLatin}{\um@usv@bfitLatin}
412     \fi
413     \ifum@bfuplatin
414     \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bflatin}
415     \else
416     \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{\um@usv@bfitlatin}
417     \fi
418     \ifum@bfupGreek
419     \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfitGreek}{\um@usv@bfGreek}
420     \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfvarTheta}
421   \else

```

```

422 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}
423 \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfitvarTheta}
424 \fi
425 \if@um@bfupgreek
426 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfgreek}
427 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}
428 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}
429 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}
430 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}
431 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}
432 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}
433 \else
434 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}
435 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}
436 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}
437 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}
438 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}
439 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}
440 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}
441 \fi
442 \um@setmathcode{\um@usv@bfNabla,\um@usv@bfitNabla}{\um@usv@bfitNabla}
443 \um@setmathcode{\um@usv@bfsfNabla,\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}
444 \um@setmathcode{\um@usv@bfpartial,\um@usv@bfitpartial}{\um@usv@bfitpartial}
445 \um@setmathcode{\um@usv@bfsfpartial,\um@usv@bfsfitpartial}{\um@usv@bfsfitpartial}
446 \fi
447 \else
: TODO : what is supposed to happen here?
448 \fi
449 }

```

### 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.

```

450 \cs_set:Nn \um_mathmap_noparse:Nnn {
451   \clist_map_inline:nn {#2} {
452     \exp_args:No \um@addto@mathmap \um@symfont {##1}{#1}{#3}
453   }
454 }

```

`\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`).

```

455 \cs_set:Nn \um_mathmap_parse:Nnn {
456   \clist_map_inline:Nn \um@char@num@range {
457     \ifnum##1=#3\relax
458       \clist_map_inline:nn {#2} {
459         \exp_args:No \um@addto@mathmap \um@symfont {####1}{#1}{#3}
460       }
461     \fi
462   }
463 }
```

`\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.

```

464 \newcommand\um@addto@mathmap[4]{
465   \expandafter\g@addto@macro
466     \csname um_setup_\cs_to_str:N #3:\endcsname{
467     \SetMathCode{#2}{\mathalpha}{#1}{#4}
468   }
469 }
```

## 6.4 (Big) operators

Turns out that  $\text{X}\text{\LaTeX}$  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  $\text{T}\text{\LaTeX}$  *etc.*, `\def\int{\intop\nolimits}`, so there needs to be a transformation from `\int` to `\intop` during the expansion of `\UnicodeMathSymbol` in the appropriate contexts.

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

USV	Ex.	Macro	Description
U+02140	$\sum\limits_0^1$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION



U+0220F	$\prod$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	$\oint$	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oiint$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oiint$	<code>\oiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\int$	<code>\int</code>	CLOCKWISE INTEGRAL
U+02232	$\oint$	<code>\oint</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\oint$	<code>\oint</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0	$\bigwedge$	<code>\bigwedge</code>	LOGICAL OR OPERATOR
U+022C1	$\bigvee$	<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2	$\bigcap$	<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3	$\bigcup$	<code>\bigcup</code>	UNION OPERATOR
U+027D5	$\Join$	<code>\Join</code>	LEFT OUTER JOIN
U+027D6	$\Join$	<code>\Join</code>	RIGHT OUTER JOIN
U+027D7	$\Join$	<code>\Join</code>	FULL OUTER JOIN
U+027D8	$\bot$	<code>\bot</code>	LARGE UP TACK
U+027D9	$\top$	<code>\top</code>	LARGE DOWN TACK
U+029F8	$/$	<code>\sol</code>	BIG SOLIDUS
U+029F9	$\backslash$	<code>\bsol</code>	BIG REVERSE SOLIDUS
U+02A00	$\odot$	<code>\odot</code>	N-ARY CIRCLED DOT OPERATOR
U+02A01	$\oplus$	<code>\oplus</code>	N-ARY CIRCLED PLUS OPERATOR
U+02A02	$\otimes$	<code>\otimes</code>	N-ARY CIRCLED TIMES OPERATOR

U+02A03		\bigcupdot	N-ARY UNION OPERATOR WITH DOT
U+02A04		\biguplus	N-ARY UNION OPERATOR WITH PLUS
U+02A05		\bigsqcap	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06		\bigsqcup	N-ARY SQUARE UNION OPERATOR
U+02A07		\conjquant	TWO LOGICAL AND OPERATOR
U+02A08		\disjquant	TWO LOGICAL OR OPERATOR
U+02A09		\bigtimes	N-ARY TIMES OPERATOR
U+02A0B		\sumint	SUMMATION WITH INTEGRAL
U+02A0C		\iiiint	QUADRUPLE INTEGRAL OPERATOR
U+02A0D		\intbar	FINITE PART INTEGRAL
U+02A0E		\intBar	INTEGRAL WITH DOUBLE STROKE
U+02A0F		\fint	INTEGRAL AVERAGE WITH SLASH
U+02A10		\cirfnint	CIRCULATION FUNCTION
U+02A11		\awint	ANTICLOCKWISE INTEGRATION LINE INTEGRATION WITH RECTANGULAR
U+02A12		\rppolint	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A13		\scpolint	PATH AROUND POLE LINE INTEGRATION NOT INCLUDING THE
U+02A14		\npolint	POLE
U+02A15		\pointint	INTEGRAL AROUND A POINT OPERATOR
U+02A16		\sqint	QUATERNION INTEGRAL OPERATOR INTEGRAL WITH LEFTWARDS ARROW
U+02A17		\intlarhk	WITH HOOK
U+02A18		\intx	INTEGRAL WITH TIMES SIGN
U+02A19		\intcap	INTEGRAL WITH INTERSECTION
U+02A1A		\intcup	INTEGRAL WITH UNION
U+02A1B		\upint	INTEGRAL WITH OVERBAR
U+02A1C		\lowint	INTEGRAL WITH UNDERBAR
U+02A1D		\Join	JOIN
U+02A1E		\bigtriangleleft	LARGE LEFT TRIANGLE OPERATOR
U+02A1F		\zcmp	Z NOTATION SCHEMA COMPOSITION

U+02A20	$\gg$	<code>\zpipe</code>	Z NOTATION SCHEMA PIPING
U+02A21	$\uparrow$	<code>\zproject</code>	Z NOTATION SCHEMA PROJECTION
U+02AFC	$\big $	<code>\biginterleave</code>	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFF	$\big $	<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 14). I’ve chosen essentially just the operators that look like integrals; hopefully a better mathematician can help me out here. I’ve a feeling that it’s more useful *not* to include the multiple integrals such as  $\iiint$ , but that might be a matter of preference.

```

470 \def\um@nolimits{
471   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oiint\@elt\oiint
472   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrackwise\@elt\sumint
473   \@elt\intbar\@elt\intBar\@elt\oint\@elt\cirfnint\@elt\awint\@elt\rppoint
474   \@elt\scpolint\@elt\ntpolint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
475   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint
476 }

```

`\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`.

```

477 \newcommand\addnolimits[1]{
478   \expandafter\def\expandafter\um@nolimits\expandafter{\um@nolimits\@elt#1}
479 }

```

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

```

480 \def\removenolimits#1{
481   \begingroup
482     \def\@elt##1{
483       \ifx##1#1\else
484         \noexpand\@elt\noexpand##1
485       \fi}
486     \xdef\um@nolimits{\um@nolimits}
487   \endgroup
488 }

```

$$\iiint_V$$

$$\iiint_V$$

$$\iiint_V$$

---

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

---

## 6.5 Radicals

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

Also, what about right-to-left square roots?

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

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

---

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

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

---

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

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

---

## 6.6 Delimiters

`\left` We redefine the primitive to be preceded by `\mathopen`; this gives much better spacing in cases such as `\sin\left....` Courtesy of Frank Mittelbach:

<http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3853&prlatex/3754>

```
490 \let\left@primitive\left
```

```
491 \def\left{\mathopen{}\left@primitive}
```

No re-definition is made for `\right` because I don't believe it to be necessary.

: TODO : 'fences', e.g., `\vert`

$$\left(\left(\left(\left((x)^1\right)^2\right)^3\right)^4\right)^5$$

$$\left[\left[\left[\left[y\right]^1\right]^2\right]^3\right]^4\right]^5$$

$$\left\{\left\{\left\{\left\{\left\{\mathbf{z}\right\}^1\right\}^2\right\}^3\right\}^4\right\}^5$$

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

Here are all `\mathopen` characters:

USV	Ex.	Macro	Description
U+00028	(	\lparen	LEFT PARENTHESIS
U+0005B	[	\lbrack	LEFT SQUARE BRACKET
U+0007B	{	\lbrace	LEFT CURLY BRACKET
			DOUBLE ANGLE QUOTATION MARK
U+000AB	«	\guillemotleft	(GUILLEMET), LEFT
U+02018	‘	\lq	SINGLE QUOTATION MARK, LEFT
U+0201A	,	\quotesinglbase	RISING SINGLE QUOTE, LEFT (LOW)
U+0201E	„	\quotdblbase	RISING DOUBLE QUOTE, LEFT (LOW)
			SINGLE ANGLE QUOTATION MARK
U+02039	<	\guilsinglleft	(GUILLEMET), LEFT
U+0221A	√	\sqrt	RADICAL
U+0221B	∛	\cuberoot	CUBE ROOT
U+0221C	∜	\fourthroot	FOURTH ROOT
U+02308	⌈	\lceil	LEFT CEILING
U+0230A	⌋	\lfloor	LEFT FLOOR
U+0231C	⌵	\ulcorner	UPPER LEFT CORNER
U+0231E	⌷	\llcorner	LOWER LEFT CORNER
			LIGHT LEFT TORTOISE SHELL BRACKET
U+02772		\lbrbrak	ORNAMENT
U+027C5	⎵	\lbag	LEFT S-SHAPED BAG DELIMITER
U+027CC	⎶	\longdivision	LONG DIVISION
			MATHEMATICAL LEFT WHITE SQUARE
U+027E6	⌌	\lBrack	BRACKET
U+027E8	⌠	\langle	MATHEMATICAL LEFT ANGLE BRACKET
			MATHEMATICAL LEFT DOUBLE ANGLE
U+027EA	⌡	\lAngle	BRACKET
			MATHEMATICAL LEFT WHITE TORTOISE
U+027EC		\Lbrbrak	SHELL BRACKET
U+02983	⎵	\lBrace	LEFT WHITE CURLY BRACKET
U+02985	(	\lParen	LEFT WHITE PARENTHESIS

U+02987	⟦	<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
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>\lblrbrak</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
U+0298E	⤵	<code>\rbracklrtick</code>	RIGHT SQUARE BRACKET WITH TICK IN BOTTOM CORNER
U+02990	⤵	<code>\rbrackurtick</code>	RIGHT SQUARE BRACKET WITH TICK IN TOP CORNER
U+02992	⤵	<code>\rangledot</code>	RIGHT ANGLE BRACKET WITH DOT
U+02994	⤵	<code>\rpargengtr</code>	RIGHT ARC GREATER-THAN BRACKET
U+02998	⤵	<code>\rblkbbrbrak</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	ˆ	<code>\acute</code>	ACUTE ACCENT
U+00302	ˆ	<code>\hat</code>	CIRCUMFLEX ACCENT
U+00303	˜	<code>\tilde</code>	TILDE
U+00304	ˉ	<code>\bar</code>	MACRON
U+00305	ˆ	<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	ˆ	<code>\ovhook</code>	COMBINING HOOK ABOVE
U+0030A	ˆ	<code>\ocirc</code>	RING
U+0030C	ˇ	<code>\check</code>	CARON
U+00310	ˆ	<code>\candra</code>	CANDRABINDU (NON-SPACING)
U+00312	ˆ	<code>\oturnedcomma</code>	COMBINING TURNED COMMA ABOVE
U+00313	ˆ	<code>\osmooth</code>	GREEK PSILI (SMOOTH BREATHING) (NON-SPACING)
U+00314	ˆ	<code>\orough</code>	GREEK DASIA (ROUGH BREATHING) (NON-SPACING)
U+00315	ˆ	<code>\ocommatopright</code>	COMBINING COMMA ABOVE RIGHT
U+0031A	ˆ	<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			COMBINING LONG VERTICAL LINE
U+020D2		<code>\vertoverlay</code>	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>\overleftrightarrow</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
U+020EC		<code>\underrightharpoondown</code>	COMBINING RIGHTWARDS HARPOON WITH BARB DOWNWARDS
U+020ED		<code>\underleftharpoondown</code>	COMBINING LEFTWARDS HARPOON WITH 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.

```

492 \newcommand\um@zf@feature[2]{
493   \define@key[zf]{options}{#1}[]{}
494   \if@um@fontspec@feature
495     #2
496   \else
497     \PackageError{fontspec/unicode-math}
498       {The ‘#1’ font feature can only be used for maths fonts}
499       {The feature you tried to use can only be in commands
500        like \protect\setmathfont}
501   \fi
502 }
503 }
```

### 7.1 OpenType maths font features

```

504 \um@zf@feature{ScriptStyle}{
505   \zf@update@ff{+ssty=0}
506 }
507 \um@zf@feature{ScriptScriptStyle}{
```



```

508 \zf@update@ff{+ssty=1}
509 }

```

## 7.2 Script and scriptscript font options

```

510 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
511 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
512 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
513 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

## 7.3 Range processing

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

```

514 \define@choicekey+[um]{options}{Range}[\@tempa\@tempb]{ALL}{
515   \ifcase\@tempb\relax
516     \global\let\um@char@range\@empty
517   \fi
518 }{
519   \xdef\um@char@range{#1}
520 }

```

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 8.

Table 8: Ranges accepted by `\um@parse@range`.

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

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

```

521 \newcommand\um@parse@term[4]{
522   \clist_map_variable:NNn \um@char@range \@ii {

```

```

523 \unless\ifx\@ii\@empty
524 \@tempswafalse

```

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

```

525 \expandafter\um@firstchar\expandafter{\@ii}
526 \ifx\@tempa\um@backslash
527 \expandafter\ifx\@ii#2\relax
528 \@tempswatrue
529 \else
530 \expandafter\ifx\@ii#3\relax
531 \@tempswatrue
532 \fi
533 \fi

```

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

```

534 \else
535 \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
536 \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.

```

537 \if@tempswa
538 \ifx\um@char@num@range\@empty
539 \g@addto@macro\um@char@num@range{#1}
540 \else
541 \g@addto@macro\um@char@num@range{,#1}
542 \fi
543 #4%
544 \fi
545 \fi
546 }
547 }
548 \def\um@firstof#1#2\@nil{#1}
549 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
550 \def\um@firstchar#1{\edef\@tempa{\expandafter\um@firstof\string#1\@nil}}

```

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

```

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

```

`\um@parse@range` Weird syntax. As shown previously in table 8, this macro can be passed four different input types via `\um@parse@term`.

```

551 \def\um@parse@range#1-#2-#3\@nil#4\@nil{

```

```

552 \def\@tempa{#1}
553 \def\@tempb{#2}

```

---

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

---

```

554 \expandafter\ifx\expandafter\@marker\@tempb\relax
555 \ifnum#4=#1\relax
556 \@tempswatrue
557 \fi
558 \else

```

---

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

---

```

559 \ifx\@empty\@tempb
560 \ifnum#4>\numexpr#1-1\relax
561 \@tempswatrue
562 \fi
563 \else

```

---

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

---

```

564 \ifx\@empty\@tempa
565 \ifnum#4<\numexpr#2+1\relax
566 \@tempswatrue
567 \fi

```

---

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

---

```

568 \else
569 \ifnum#4>\numexpr#1-1\relax
570 \ifnum#4<\numexpr#2+1\relax
571 \@tempswatrue
572 \fi
573 \fi
574 \fi
575 \fi
576 \fi
577 }

```

\um@setmathcode #1 : Starting input char(s)

#2 : Number of iterations  
 #3 : Starting output char  
 Loops through character ranges setting \mathcode.

```

578 \newcommand\um@setmathcode[3][1]{
579   \clist_map_variable:nNn {#2} \l_um_input_num {
580     \prg_stepwise_variable:nnnNn{1}{1}{#1} \l_um_incr_num {
581       \SetMathCode
582         {\numexpr \l_um_incr_num+ \l_um_input_num - 1\relax}
583         {\mathalpha}{\um@symfont}
584         {\numexpr \l_um_incr_num + #3 - 1\relax}
585     }
586   }
587 }

```

```

\um_set_mathalphabet_char:Nnnn #1 : Maths alphabet
#2 : Input char(s)
#3 : Output char
Loops through character ranges setting \mathcode.
588 \cs_set:Npn \exp_args:Nnff {\::n\::f\::f\:::}
589 \cs_new:Nn \um_set_mathalphabet_char:Nnn {
590   \clist_map_variable:nNn {#2} \l_um_input_num {
591     \exp_args:Nnff \um_mathmap:Nnn {#1}
592     {\number\numexpr\l_um_input_num\relax} {\number\numexpr#3\relax}
593   }
594 }

```

\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.

```

595 \cs_new:Nn \um_set_mathalph_range:nNnn {
596   \clist_map_variable:nNn {#3} \l_um_input_num {
597     \prg_stepwise_variable:nnnNn {0}{1}{#1} \l_um_inc_num {
598       \exp_args:Nnff \um_mathmap:Nnn {#2}
599       {\number\numexpr \l_um_inc_num + \l_um_input_num \relax}
600       {\number\numexpr \l_um_inc_num + #4 \relax}
601     }
602   }
603 }
604 \cs_new:Nn \um_set_mathalphabet_numbers:Nnn {
605   \um_set_mathalph_range:nNnn {9}{#1}{#2}{#3}
606 }
607 \cs_new:Nn \um_set_mathalphabet_latin:Nnn {
608   \um_set_mathalph_range:nNnn {25}{#1}{#2}{#3}
609 }
610 \cs_new:Nn \um_set_mathalphabet_greek:Nnn {

```

```

611 \um_set_mathalph_range:nNnn {24}{#1}{#2}{#3}
612 }

```

---

**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.

```

613 \AtBeginDocument{\um@resolve@greek}
614 \newcommand\um@resolve@greek{
615   \def\Alpha{\mitAlpha}
616   \def\Beta{\mitBeta}
617   \def\Gamma{\mitGamma}
618   \def\Delta{\mitDelta}
619   \def\Epsilon{\mitEpsilon}
620   \def\Zeta{\mitZeta}
621   \def\Eta{\mitEta}
622   \def\Theta{\mitTheta}
623   \def\Iota{\mitIota}
624   \def\Kappa{\mitKappa}
625   \def\Lambda{\mitLambda}
626   \def\Mu{\mitMu}
627   \def\Nu{\mitNu}
628   \def\Xi{\mitXi}
629   \def\Omicron{\mitOmicron}
630   \def\Pi{\mitPi}
631   \def\Rho{\mitRho}
632   \def\varTheta{\mitvarTheta}
633   \def\Sigma{\mitSigma}
634   \def\Tau{\mitTau}
635   \def\Upsilon{\mitUpsilon}
636   \def\Phi{\mitPhi}
637   \def\Chi{\mitChi}
638   \def\Psi{\mitPsi}
639   \def\Omega{\mitOmega}

```

Lowercase:

```

640   \def\alpha{\mitalpha}
641   \def\beta{\mitbeta}
642   \def\gamma{\mitgamma}
643   \def\delta{\mitdelta}
644   \def\epsilon{\mitepsilon}

```

```

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

```

\um@def@numbers

```

672 \newcommand\um@def@numbers{
673   \um@setmathcode[10]{\um@usv@num}{\um@usv@num}
674 }

```

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

```

675 \cs_set:Nn \um_setup_literals: {
676   \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}
677   \um@setmathcode[26]{\um@usv@itLatin}{\um@usv@itLatin}
678   \um@setmathcode[26]{\um@usv@itlatin}{\um@usv@itlatin}
679   \um@setmathcode{\um@usv@ith}{\um@usv@ith}
680   \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}
681   \um@setmathcode[25]{\um@usv@upGreek}{\um@usv@upGreek}
682   \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}
683   \um@setmathcode[25]{\um@usv@itGreek}{\um@usv@itGreek}
684   \um@setmathcode[25]{\um@usv@upgreek}{\um@usv@upgreek}
685   \um@setmathcode{\um@usv@Nabla}{\um@usv@Nabla}
686   \um@setmathcode{\um@usv@itNabla}{\um@usv@itNabla}
687   \um@setmathcode{\um@usv@partial}{\um@usv@partial}

```

```

688 \um@setmathcode{\um@usv@itpartial}{\um@usv@itpartial}
689 }

\um_setup_bf_literals: TODO: other literal symbols

690 \cs_set:Nn \um_setup_bf_literals: {
691   \um@setmathcode[26]{\um@usv@bfLatin}{\um@usv@bfLatin}
692   \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}
693   \um@setmathcode[26]{\um@usv@bfitLatin}{\um@usv@bfitLatin}
694   \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}
695   \um@setmathcode[25]{\um@usv@bfGreek}{\um@usv@bfGreek}
696   \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}
697   \um@setmathcode[25]{\um@usv@bfitGreek}{\um@usv@bfitGreek}
698   \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}
699 }

\um@def@upLatin

700 \newcommand\um@def@upLatin{
701   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
702 }

\um@def@itLatin

703 \newcommand\um@def@itLatin{
704   \um@setmathcode[26]{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
705 }

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

706 \newcommand\um@def@itlatin{
707   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
708   \um@setmathcode{\`h,\um@usv@ith}{\um@usv@ith}
709 }

\um@def@uplatin

710 \newcommand\um@def@uplatin{
711   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
712   \um@setmathcode{\um@usv@ith}{\`h}
713 }

\um@def@upGreek

714 \newcommand\um@def@upGreek{
715   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
716   \um@setmathcode{\um@usv@varTheta,"1D6F3}{\um@usv@varTheta}
717 }

\um@def@itGreek

718 \newcommand\um@def@itGreek{

```

```

719 \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
720 \um@setmathcode{\um@usv@varTheta}{\um@usv@itvarTheta}
721 }

```

\um@def@upgreek

```

722 \newcommand\um@def@upgreek{
723   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
724   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
725   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
726   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
727   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
728   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
729   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
730 }

```

\um@def@itgreek

```

731 \newcommand\um@def@itgreek{
732   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
733   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
734   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
735   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
736   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
737   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
738   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
739 }

```

## File II

# 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.

```

1 \cs_new:Nn \um_setup_math_alphabet:n {
2   \um_glyph_if_exist:nTF {\csname \um@usv@#1latin \endcsname}{
3     \um_maybe_init_alphabet:n {#1}

```



```

4 \um_prepare_alph:n {#1}
5 \use:c {um_config_math#1:}
6 }{
7 \um@PackageWarning{Math~ alphabet~ "#1"~ not~ found~ with~ this~ font}
8 }
9 }
10 \cs_set:Nn \um_init_alphabet:n {
11 \cs_set_eq:CN {um_setup_math#1:} \prg_do_nothing:
12 }

```

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

```

13 \prg_new_conditional:Nnn \um_glyph_if_exist:n {p,TF,T,F} {
14 \etex_iffontchar:D \um@font #1 \scan_stop: \prg_return_true: \else: \prg_return_false: \fi:
15 }

```

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

```

16 \cs_new:Nn \um_prepare_alph:n {
17 \cs_if_exist:cF {um_math#1:n} {
18 \cs_set:cpn {um_math#1:n} ##1 {
19 \begingroup \use:c {um_setup_math#1:} ##1 \endgroup
20 }
21 \cs_set_protected:cpn {math#1} {
22 \mode_if_math:F {
23 \expandafter\non@alpherr\expandafter{\csname math#1\endcsname\space}
24 }
25 \use:c {um_math#1:n}
26 }
27 }
28 }
29 \cs_new:Nn \um_setup_alphabets: {
30 \um_setup_math_alphabet:n {up }
31 \um_setup_math_alphabet:n {it }
32 \um_setup_math_alphabet:n {bb }
33 \um_setup_math_alphabet:n {scr }
34 \um_setup_math_alphabet:n {frak }
35 \um_setup_math_alphabet:n {sf }
36 \um_setup_math_alphabet:n {sfit }
37 \um_setup_math_alphabet:n {tt }
38 \um_setup_math_alphabet:n {bf }
39 \um_setup_math_alphabet:n {bfup }
40 \um_setup_math_alphabet:n {bfit }
41 \um_setup_math_alphabet:n {bfscr }
42 \um_setup_math_alphabet:n {bffrak}

```

```

43 \um_setup_math_alphabet:n {bfsf }
44 \um_setup_math_alphabet:n {bfsfit}
45 }

```

: TODO : nested alphabets?

### 7.3.1 Upright: `\mathup`

$ \begin{array}{l} \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\ \text{abcdefghijklmnopqrstuvwxyz} \\ \text{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ} \\ \text{αβγδεζηθικλμνξοπρστυφχψω εθκφρϖ} \end{array} $	<pre> \$ \mathup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \ \$ \mathup{abcdefghijklmnopqrstuvwxyz}\$ \ \$ \mathup{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$ \quad\$ \mathup{0}\$ \ \$ \mathup{αβγδεζηθικλμνξοπρστυφχψω}\$ \quad\$ \mathup{000000}\$ \ </pre>
--	---

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

```

46 \cs_new:Npn \um_config_mathup: {
47   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
48   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
49   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
50   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
51   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}
52   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}
53   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}
54   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
55   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
56   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
57   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
58   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
59   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
60 }

```

### 7.3.2 Italic: `\mathit`

$ \begin{array}{l} \text{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\ \text{abcdefghijklmnopqrstuvwxyz} \\ \text{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ} \\ \text{αβγδεζηθικλμνξοπρστυφχψω εθκφρϖ} \end{array} $	<pre> \$ \mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \ \$ \mathit{abcdefghijklmnopqrstuvwxyz}\$ \ \$ \mathit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$ \quad\$ \mathit{0}\$ \ \$ \mathit{αβγδεζηθικλμνξοπρστυφχψω}\$ \quad\$ \mathit{000000}\$ \ </pre>
--	---

Roman:

```

61 \cs_new:Npn \um_config_mathit: {
62   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
63   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
64   \um_set_mathalphabet_char:Nnn{\mathit}{\h,\um@usv@ith}{\um@usv@ith}

```

Greek:

```

65 \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
66 \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
67 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}
68 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}
69 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarThe
70 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itva
71 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvarthe
72 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkap
73 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
74 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
75 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
76 }

```

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

---

$0123456789$	<code>\$\mathbb{0123456789}\$ \\\</code>
$ABCDEFGHIJKLMN OPQRSTUVWXYZ$	<code>\$\mathbb{ABCDEFGHIJKLMN OPQRSTUVWXYZ}\$ \\\</code>
$abcdefghijklmn opqrstuvwxyz$	<code>\$\mathbb{abcdefghijklmn opqrstuvwxyz}\$ \\\</code>

---

Numbers:

```

77 \cs_new:Npn \um_config_mathbb: {
78 \um_set_mathalphabet_numbers:Nnn{\mathbb}{\um@usv@num}{\um@usv@bbnum}

```

Roman uppercase:

```

79 \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bbLatin}
80 \um_set_mathalphabet_char:Nnn{\mathbb}{`C,"1D60A}{`"2102}
81 \um_set_mathalphabet_char:Nnn{\mathbb}{`H,"1D60F}{`"210D}
82 \um_set_mathalphabet_char:Nnn{\mathbb}{`N,"1D60F}{`"2115}
83 \um_set_mathalphabet_char:Nnn{\mathbb}{`P,"1D617}{`"2119}
84 \um_set_mathalphabet_char:Nnn{\mathbb}{`Q,"1D618}{`"211A}
85 \um_set_mathalphabet_char:Nnn{\mathbb}{`R,"1D619}{`"211D}
86 \um_set_mathalphabet_char:Nnn{\mathbb}{`Z,"1D621}{`"2124}

```

Roman lowercase:

```

87 \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bblatin}
88 }

```

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

---

$\mathscr{ABCDEFGHIJKLMN OPQRSTUVWXYZ}$	<code>\$\mathscr{ABCDEFGHIJKLMN OPQRSTUVWXYZ}\$ \\\</code>
$\mathscr{abcdefghijklmn opqrstuvwxyz}$	<code>\$\mathscr{abcdefghijklmn opqrstuvwxyz}\$ \\\</code>

---

```

89 \cs_new:Npn \um_config_mathscr: {
90   \um_set_mathalphabet_latin:Nnn{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}
91   \um_set_mathalphabet_char:Nnn{\mathscr}{`\B,"1D435}{\um@usv@scrLatin}
92   \um_set_mathalphabet_char:Nnn{\mathscr}{`\E,"1D438}{\um@usv@scrLatin}
93   \um_set_mathalphabet_char:Nnn{\mathscr}{`\F,"1D439}{\um@usv@scrLatin}
94   \um_set_mathalphabet_char:Nnn{\mathscr}{`\H,"1D43B}{\um@usv@scrLatin}
95   \um_set_mathalphabet_char:Nnn{\mathscr}{`\I,"1D43C}{\um@usv@scrLatin}
96   \um_set_mathalphabet_char:Nnn{\mathscr}{`\L,"1D43F}{\um@usv@scrLatin}
97   \um_set_mathalphabet_char:Nnn{\mathscr}{`\M,"1D440}{\um@usv@scrLatin}
98   \um_set_mathalphabet_char:Nnn{\mathscr}{`\R,"1D445}{\um@usv@scrLatin}
99   \um_set_mathalphabet_latin:Nnn{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}
100  \um_set_mathalphabet_char:Nnn{\mathscr}{`\e,"1D452}{\um@usv@scrLatin}
101  \um_set_mathalphabet_char:Nnn{\mathscr}{`\g,"1D454}{\um@usv@scrLatin}
102  \um_set_mathalphabet_char:Nnn{\mathscr}{`\o,"1D45C}{\um@usv@scrLatin}
103 }

```

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

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

Letters, with exceptions { $\mathfrak{C}$ ,  $\mathfrak{S}$ ,  $\mathfrak{Z}$ ,  $\mathfrak{R}$ ,  $\mathfrak{J}$ }:

```

104 \cs_new:Npn \um_config_mathfrak: {
105   \um_set_mathalphabet_latin:Nnn{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}
106   \um_set_mathalphabet_char:Nnn{\mathfrak}{`\C,"1D436}{\um@usv@frakLatin}
107   \um_set_mathalphabet_char:Nnn{\mathfrak}{`\H,"1D43B}{\um@usv@frakLatin}
108   \um_set_mathalphabet_char:Nnn{\mathfrak}{`\I,"1D43C}{\um@usv@frakLatin}
109   \um_set_mathalphabet_char:Nnn{\mathfrak}{`\R,"1D445}{\um@usv@frakLatin}
110   \um_set_mathalphabet_char:Nnn{\mathfrak}{`\Z,"1D44D}{\um@usv@frakLatin}
111   \um_set_mathalphabet_latin:Nnn{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}
112 }

```

### 7.3.6 Sans serif: `\mathsf`

$\mathsf{0123456789}$	<code>\$\mathsf{0123456789}\$ \</code>
$\mathsf{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$	<code>\$\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
$\mathsf{a b c d e f g h i j k l m n o p q r s t u v w x y z}$	<code>\$\mathsf{abcdefghijklmnopqrstuvwxyz}\$ \</code>

```

113 \cs_new:Npn \um_config_mathsf: {
114   \um_set_mathalphabet_numbers:Nnn{\mathsf}{\um@usv@num}{\um@usv@sfnum}
115   \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfLatin}
116   \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sflatin}
117 }

```

### 7.3.7 Sans serif italic: `\mathsf`

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

```

118 \cs_new:Npn \um_config_mathsf: {
119   \um_set_mathalphabet_numbers:Nnn{\mathsf}{\um@usv@num}{\um@usv@sfnm}
120   \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLatin}
121   \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfitlatin}
122 }

```

### 7.3.8 Typewriter or monospaced: `\mathtt`

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

```

123 \cs_new:Npn \um_config_mathtt: {
124   \um_set_mathalphabet_numbers:Nnn{\mathtt}{\um@usv@num}{\um@usv@ttnum}
125   \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}
126   \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}
127 }

```

## 7.4 Bold alphabets' character mappings

### 7.4.1 Bold: `\mathbf`

0123456789	<code>\mathbf{0123456789}</code>
ABCDEFGHIJKLMNOPQRSTUVWXYZ	<code>\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}</code>
abcdefghijklmnopqrstuvwxyz	<code>\mathbf{abcdefghijklmnopqrstuvwxyz}</code>
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	<code>\mathbf{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}</code>
Θ	<code>\mathbf{Θ}</code>
αβγδεζηθικλμνξοπρστυφχψω	<code>\mathbf{αβγδεζηθικλμνξοπρστυφχψω}</code>
εθκφρϑ	<code>\mathbf{εθκφρϑ}</code>

```

128 \cs_new:Npn \um_config_mathbf: {
129   \um_set_mathalphabet_numbers:Nnn{\mathbf}{\um@usv@num}{\um@usv@bfnum}
130   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Digamma}{1D7CA}
131   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@digamma}{1D7CB}
132   \ifum@bfliteral

```

```

133 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin}{\um@usv@bfLatin}
134 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@itLatin}{\um@usv@bfitLatin}
135 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin}{\um@usv@bflatin}
136 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@itLatin}{\um@usv@bfitlatin}
137 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek}{\um@usv@bfGreek}
138 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itGreek}{\um@usv@bfitGreek}
139 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek}{\um@usv@bfGreek}
140 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itgreek}{\um@usv@bfitgreek}
141 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
142 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}
143 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}
144 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@bfDigamma}
145 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial}{\um@usv@bfpartial}
146 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}
147 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}
148 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}
149 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}
150 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}
151 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}
152 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@digamma}{\um@usv@bfdigamma}
153 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
154 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itNabla}{\um@usv@bfitNabla}
155 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}
156 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
157 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}
158 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
159 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}
160 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}
161 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}
162 \else
163 \if@um@bfupLatin
164 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfLatin}
165 \else
166 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}
167 \fi
168 \if@um@bfuplatin
169 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bflatin}
170 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfuph}
171 \else
172 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
173 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
174 \fi
175 \if@um@bfupGreek
176 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfGreek}
177 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvar}
178 \else

```

```

179 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGree
180 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitv
181 \fi
182 \if@um@bfupgreek
183 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfgreek}
184 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@b
185 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfvar
186 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfvar
187 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
188 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
189 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
190 \else
191 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgree
192 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@b
193 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitv
194 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitv
195 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarph
196 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrh
197 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
198 \fi
199 \fi
200 }
201 % \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla}{\um@usv@bfitNabla}
202 % \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial}{\um@usv@bfitpartial}
203 % \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itNabla}{\um@usv@bfititNabla}
204 % \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itpartial}{\um@usv@bfititpartial}

```

#### 7.4.2 Bold Italic: `\mathbfit`

---

**0123456789**  
**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ**  
**αβγδεζηθικλμνξοπρστυφχψω εθκφρτσ**

---

```

$ \mathbfit{0123456789}$ \
$ \mathbfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \
$ \mathbfit{abcdefghijklmnopqrstuvwxyz}$ \
$ \mathbfit{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}$ \quad
$ \mathbfit{0}$ \
$ \mathbfit{00000000000000000000000000000000}$ \quad
$ \mathbfit{000000}$ \

```

---

```

205 \cs_new:Npn \um_config_mathbfit: {
206   \um_set_mathalphabet_numbers:Nnn{\mathbfit}{\um@usv@num}{\um@usv@bfnun}
207   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}
208   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
209   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}
210   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}
211   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@bfLatin}{\um@usv@bfitLatin}
212   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@bflatin}{\um@usv@bfitlatin}
213   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@bfGreek}{\um@usv@bfitGreek}

```

```

214 \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@bfgreek}{\um@usv@bfitgreek}
215 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitva
216 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfitNabla}
217 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpart
218 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bf
219 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitva
220 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitva
221 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi
222 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}
223 \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
224 }

```

### 7.4.3 Bold Italic: \mathbfup

**0123456789**  
**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ**  
**αβγδεζηθικλμνξοπρστυφχψω εθκφρψ**

```

$\mathbfup{0123456789}$ \
$\mathbfup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \
$\mathbfup{abcdefghijklmnopqrstuvwxyz}$ \
$\mathbfup{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad
$\mathbfup{ }$ \
$\mathbfup{00000000000000000000000000000000}$\quad
$\mathbfup{0000000}$ \

```

```

225 \cs_new:Npn \um_config_mathbfup: {
226 \um_set_mathalphabet_numbers:Nnn{\mathbfup}{\um@usv@num}{\um@usv@bnum}
227 \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bLatin}
228 \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@blatin}
229 \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bGreek}
230 \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfgreek}
231 \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@bLatin}{\um@usv@bLatin}
232 \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@blatin}{\um@usv@blatin}
233 \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@bGreek}{\um@usv@bGreek}
234 \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@bfgreek}{\um@usv@bfgreek}
235 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvarT
236 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfNabla}
237 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfpartia
238 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bf
239 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfvart
240 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfvark
241 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
242 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
243 \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
244 }

```



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

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**

```
\setmathfont{Cambria Math}
$\mathbffrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbffrak{abcdefghijklmnopqrstuvwxyz}$ \\
```

```
245 \cs_new:Npn \um_config_mathbffrak: {
246   \um_set_mathalphabet_numbers:Nnn{\mathbffrak}{\um@usv@num}{\um@usv@bfnum}
247   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@upLatin, \um@usv@itLatin, \um@usv@frakLat
248   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@uplatin, \um@usv@itlatin, \um@usv@fraklati
249 }
```

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

***ABCDEFGHIJKLMNOPQRSTUVWXYZ***  
***abcdefghijklmnopqrstuvwxyz***

```
\setmathfont{Cambria Math}
$\mathbfscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbfscr{abcdefghijklmnopqrstuvwxyz}$ \\
```

```
250 \cs_new:Npn \um_config_mathbfscr: {
251   \um_set_mathalphabet_numbers:Nnn{\mathbfscr}{\um@usv@num}{\um@usv@bfnum}
252   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfscrLat
253   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfscrLat
254 }
```

#### 7.4.6 Bold sans serif: `\mathbfsf`

**0123456789**  
**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ**  
**αβγδεζηθικλμνξοπρστυφχψω εδκφρϖ**

```
\setmathfont{STIXGeneral-Bold}
$\mathbfsf{0123456789}$ \\
$\mathbfsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbfsf{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathbfsf{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad
$\mathbfsf{ }$ \\
$\mathbfsf{00000000000000000000000000000000}$\quad
$\mathbfsf{000000}$ \\
```

Numbers (always upright) and letters:

```
255 \cs_new:Npn \um_config_mathbfsf: {
256   \um_set_mathalphabet_numbers:Nnn{\mathbfsf}{\um@usv@num}{\um@usv@bfnum}
257   \um_set_mathalphabet_latin:Nnn{\mathbfsf}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfsfLatin
258   \um_set_mathalphabet_latin:Nnn{\mathbfsf}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfsflatin
259   \um_set_mathalphabet_greek:Nnn{\mathbfsf}{\um@usv@upGreek, \um@usv@itGreek}{\um@usv@bfsfGreek
260   \um_set_mathalphabet_greek:Nnn{\mathbfsf}{\um@usv@upgreek, \um@usv@itgreek}{\um@usv@bfsfgreek
```

Others:

```
261 \um_set_mathalphabet_char:Nnn{\mathbfsf}{\um@usv@varTheta, \um@usv@itvarTheta}{1D767}
```

#### 7.4.7 Bold italic sans serif: `\mathbfsfit`

[illegible]

Other symbols:

## 7.5 Definitions of the math symbols

50

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.

```

287 \begingroup
288   \char_make_other:N \^
289   \gdef\um@scancharlet#1="#2\@nil{
290     \lowercase{
291       \scantokens{\global\let#1=^^^^^#2}
292     }
293   }
294   \gdef\um@scanactivedef"#1\@nil#2{
295     \lowercase{
296       \scantokens{\global\def^^^^^#1{#2}}
297     }
298   }
299 \endgroup
300 \let\unicodemathgobble\@gobble

```

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

```

301 \begingroup
302   \def\UnicodeMathSymbol#1#2#3#4{
303     \um@scancharlet#2=#1\@nil
304   }
305   \@input{unicode-math-table.tex}
306 \endgroup

```

## 7.6 Epilogue

Lots of little things to tidy up.

### 7.6.1 Unicode radicals

Undo the damage made to \sqrt:

```

307 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}

```

### 7.6.2 Primes

---

```

[x'] [x'''] [x''''']
[x'] [x'''] [x''''']
[x'] [x'''] [x''''']

```

---

```

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

```

---

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 `ssty` 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.

```
308 \muskip_new:N \g_um_primekern_muskip
309 \muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }% arbitrary
310 \num_new:N \l_um_primecount_num
311 \cs_new:Nn \um_nprimes:n {
312   \primesingle
313   \prg_replicate:nn {#1-1} { \mskip \g_um_primekern_muskip \primesingle }
314 }
315 \cs_new:Nn \um_nprimes_select:n {
316   \prg_case_int:nnn {#1}{
317     {1} { \primesingle }
```

```

318 {2} {
319   \um_glyph_if_exist:nTF {"2033} {\primedouble} {\um_nprimes:n {#1}}
320 }
321 {3} {
322   \um_glyph_if_exist:nTF {"2034} {\primetriples} {\um_nprimes:n {#1}}
323 }
324 {4} {
325   \um_glyph_if_exist:nTF {"2057} {\primequadruple} {\um_nprimes:n {#1}}
326 }
327 }{
328   \um_nprimes:n {#1}
329 }
330 }

```

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?

```

331 \cs_new:Nn \um_scanprime: {
332   \bgroup
333   \num_zero:N \l_um_primecount_num
334   \um_scanprime_collect:
335 }
336 \cs_new:Nn \um_scanprime_collect: {
337   \num_incr:N \l_um_primecount_num
338   \peek_charcode_remove:NTF ' {
339     \um_scanprime_collect:
340   }{
341     \peek_meaning_remove:NTF \um_scanprime: {
342       \um_scanprime_collect:
343     }{
344       \peek_charcode_remove:NTF ^^^^2032 {
345         \um_scanprime_collect:
346       }{
347         \um_nprimes_select:n {\l_um_primecount_num}
348       } \egroup
349     }
350   }
351 }
352 }
353 \cs_set_eq:NN \prime \um_scanprime:
354 \group_begin:
355   \char_make_active:N \'
356   \char_make_active:n {"2032}
357   \cs_gset_eq:NN ' \um_scanprime:

```

```

358 \cs_gset_eq:NN ^^^^2032 \um_scanprime:
359 \group_end:

```

### 7.6.3 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`.

```

360 \def\r@@t#1#2{
361   \setbox\z@\hbox{$\m@th #1\sqrtsign{#2}$}
362   \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}
363   \raise \dimexpr(
364     \um@fontdimen@percent{65}{\um@font}\ht\z@-
365     \um@fontdimen@percent{65}{\um@font}\dp\z@
366   )\relax
367   \copy \rootbox
368   \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}
369   \box \z@
370 }

```

### 7.6.4 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`:

```

371 \def\operator@font{\um_setup_mathup:}
372 \def\to{\rightarrow}
373 \def\le{\leq}
374 \def\ge{\geq}

```

`\mathcal`

```

375 \def\mathcal{\mathscr}

```

`\mathrm`

```

376 \def\mathrm{\mathup}

```

Overriding amsmath definitions:

```

377 \AtBeginDocument{
378   \def\@cdots{\mathinner{\cdots}}
379 }

```

Interaction with beamer:

```

380 \AtBeginDocument{
381   \ifpackageloaded{beamer}{
382     \ifbeamer@suppressreplacements\else

```

```

383 \PackageWarningNoLine{unicode-math}{
384   Disabling~ beamer's~ math~ setup.^{}
385   Please~ load~ beamer~ with~ the~ [professionalfonts]~ class~ option
386 }
387 \beamer@suppressreplacementstrue
388 \fi
389 }{}
390 }

```

The end.

```

391 \ExplSyntaxOff

```

## File III

# 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));

```

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

```

10   if (texname ~ /[\\]/ &&
11       substr(texname,0,5) != "\\text" &&

```

```

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

```

Print the actual entry corresponding to the unicode character:

```

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

```

Now replace the STIX class abbreviations with their T<sub>E</sub>X macro names.

```

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

```

A ‘fence’ defined by the STIX table is something like `\vert`; in X<sub>Y</sub>T<sub>E</sub>X this is just a `\mathord` that will grow with the magic of `\XeTeXmathchardef`.

```

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

```

Fixing up a couple of things in the STIX table.

```

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

```

## A Documenting maths support in the NFSS

### A.1 Overview

In the following, *<NFSS decl.>* stands for something like `{T1}{lmr}{m}{n}`.

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

```
\DeclareSymbolFont{<name>}<NFSS decl.>
```

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

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

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



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

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

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

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

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

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

Delimiters and radicals use wrappers around TeX’s `\delimiter`/`\radical` primitives, which are re-designed in X<sub>Y</sub>TeX. The syntax used in L<sup>A</sup>TeX’s NFSS is therefore not so relevant here.

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

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

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

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

Accents are not included yet.

**Summary** For symbols, something like:

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

For characters, something like:

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

{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}}

## File IV

# 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>DELIMITEDSUBFORMULA-MINHEIGHT</code>	Minimum height required for a delimited expression to be treated as a subformula. Suggested value: normal line height $\times 1.5$ .
13	<code>DISPLAYOPERATORMIN-HEIGHT</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.
17	<code>FLATTENEDACCENTBASE-HEIGHT</code>	Maximum (ink) height of accent base that does not require flattening the accents. Suggested: cap height of the font ( <code>os2.sCapHeight</code> ).

\fontdimen	Dimension name	Description
18	SUBSCRIPTSHIFTDOWN	The standard shift down applied to subscript elements. Positive for moving in the downward direction. Suggested: $os2.ySubscriptYOffset$ .
19	SUBSCRIPTTOPMAX	Maximum allowed height of the (ink) top of subscripts that does not require moving subscripts further down. Suggested: $/5$ x-height.
20	SUBSCRIPTBASELINEDROP-MIN	Minimum allowed drop of the baseline of subscripts relative to the (ink) bottom of the base. Checked for bases that are treated as a box or extended shape. Positive for subscript baseline dropped below the base bottom.
21	SUPERSCRIPSHIFTUP	Standard shift up applied to superscript elements. Suggested: $os2.ySuperscriptYOffset$ .
22	SUPERSCRIPSHIFTUP-CRAMPED	Standard shift of superscripts relative to the base, in cramped style.
23	SUPERSCRIPBOTTOMMIN	Minimum allowed height of the (ink) bottom of superscripts that does not require moving subscripts further up. Suggested: $1/4$ 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: $4 \times \text{default rule thickness}$ .
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.
27	SPACEAFTERSCRIPT	Extra white space to be added after each subscript and superscript. Suggested: 0.5pt for a 12 pt font.

\fontdimen	Dimension name	Description
28	UPPERLIMITGAPMIN	Minimum gap between the (ink) bottom of the upper limit, and the (ink) top of the base operator.
29	UPPERLIMITBASELINERISE-MIN	Minimum distance between baseline of upper limit and (ink) top of the base operator.
30	LOWERLIMITGAPMIN	Minimum gap between (ink) top of the lower limit, and (ink) bottom of the base operator.
31	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	STACKTOPDISPLAYSTYLE-SHIFTUP	Standard shift up applied to the top element of a stack in display style.
34	STACKBOTTOMSHIFTDOWN	Standard shift down applied to the bottom element of a stack. Positive for moving in the downward direction.
35	STACKBOTTOMDISPLAY-STYLESHIFTDOWN	Standard shift down applied to the bottom element of a stack in display style. Positive for moving in the downward direction.
36	STACKGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element. Suggested: 3×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	STRETCHSTACKBOTTOM-SHIFTDOWN	Standard shift down applied to the bottom element of the stretch stack. Positive for moving in the downward direction.
40	STRETCHSTACKGAPABOVE-MIN	Minimum gap between the ink of the stretched element, and the (ink) bottom of the element above. Suggested: UpperLimitGapMin

\fontdimen	Dimension name	Description
41	STRETCHSTACKGAPBELOW-MIN	Minimum gap between the ink of the stretched element, and the (ink) top of the element below. Suggested: LowerLimitGapMin.
42	FRACTIONNUMERATOR-SHIFTUP	Standard shift up applied to the numerator.
43	FRACTIONNUMERATOR-DISPLAYSTYLESHIFTUP	Standard shift up applied to the numerator in display style. Suggested: StackTopDisplayStyleShiftUp.
44	FRACTIONDENOMINATOR-SHIFTDOWN	Standard shift down applied to the denominator. Positive for moving in the downward direction.
45	FRACTIONDENOMINATOR-DISPLAYSTYLESHIFTDOWN	Standard shift down applied to the denominator in display style. Positive for moving in the downward direction. Suggested: StackBottomDisplayStyleShiftDown.
46	FRACTIONNUMERATORGAP-MIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar. Suggested: default rule thickness
47	FRACTIONNUMDISPLAY-STYLEGAPMIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
48	FRACTIONRULETHICKNESS	Thickness of the fraction bar. Suggested: default rule thickness.
49	FRACTIONDENOMINATOR-GAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar. Suggested: default rule thickness
50	FRACTIONDENOMDISPLAY-STYLEGAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
51	SKEWEDFRACTION-HORIZONTALGAP	Horizontal distance between the top and bottom elements of a skewed fraction.
52	SKEWEDFRACTIONVERTICAL-GAP	Vertical distance between the ink of the top and bottom elements of a skewed fraction.

\fontdimen	Dimension name	Description
53	OVERBARVERTICALGAP	Distance between the overbar and the (ink) top of the base. Suggested: 3×default rule thickness.
54	OVERBARRULETHICKNESS	Thickness of overbar. Suggested: default rule thickness.
55	OVERBAREXTRAASCENDER	Extra white space reserved above the overbar. Suggested: default rule thickness.
56	UNDERBARVERTICALGAP	Distance between underbar and (ink) bottom of the base. Suggested: 3×default rule thickness.
57	UNDERBARRULETHICKNESS	Thickness of underbar. Suggested: default rule thickness.
58	UNDERBAREXTRA-DESCENDER	Extra white space reserved below the underbar. Always positive. Suggested: default rule thickness.
59	RADICALVERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: 1¼ default rule thickness.
60	RADICALDISPLAYSTYLE- VERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: default rule thickness + ¼ x-height.
61	RADICALRULETHICKNESS	Thickness of the radical rule. This is the thickness of the rule in designed or constructed radical signs. Suggested: default rule thickness.
62	RADICALEXTRAASCENDER	Extra white space reserved above the radical. Suggested: RadicalRuleThickness.
63	RADICALKERNBEFORE- DEGREE	Extra horizontal kern before the degree of a radical, if such is present. Suggested: 5/18 of em.
64	RADICALKERNAFTERDEGREE	Negative kern after the degree of a radical, if such is present. Suggested: −10/18 of em.
65	RADICALDEGREEBOTTOM- RAISEPERCENT	Height of the bottom of the radical degree, if such is present, in proportion to the ascender of the radical sign. Suggested: 60%.

## File V

# Some manner of unit testing

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

## B The regular weight alphabets

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

```
1 (*test)
2 \documentclass{article}
3 \usepackage[a6paper]{geometry}
4 \usepackage{fontspec}
5 \setmainfont{FPL Neu}
6 \usepackage{unicode-math}
7 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
8 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
9 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
10 \def\upgreek{αβγδεζηθικλμνξοπρστυφχψω}
11 \def\itLatin{abcdefghijklmnopqrstuvwxyz}
12 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
13 \def\itGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
14 \def\itgreek{αβγδεζηθικλμνξοπρστυφχψω}
15 \def\testmath#1{
16   \makebox[\linewidth][l]{
17     \makebox[0pt][l]{$\csname up#1\endcsname$}
18     \makebox[0pt][l]{$\csname it#1\endcsname$}}
19 \begin{document}
20 \setmathfont[Colour=2255FF99]{Cambria Math}
21 \parindent=0pt
22 \voffset=-1in
23 \hoffset=-1in
24 \setbox0=\vbox{
25   \testmath{Latin}\\
26   \testmath{latin}\\
27   \testmath{Greek}\\
28   \testmath{greek}}
29 \dimen0=\ht0
30 \advance\dimen0\dp0
31 \edef\papersize{papersize=\the\wd0,\the\dimen0}
32 \setbox255=\vbox{\special{\papersize}\box0}
```

```

33 \shipout\box255
34 \end{document}
35 (/test)

```

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

## C The bold alphabets

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

```

36 (*testbf)
37 \documentclass{article}
38 \usepackage[a6paper]{geometry}
39 \usepackage{fontspec}
40 \setmainfont{FPL Neu}
41 \usepackage{unicode-math}
42 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
43 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
44 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}
45 \def\upgreek{αβγδεζηθικλμνξοπρστυφχψω}
46 \def\itLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
47 \def\itlatin{abcdefghijklmnopqrstuvwxyz}
48 \def\itGreek{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}
49 \def\itgreek{αβγδεζηθικλμνξοπρστυφχψω}
50 \def\bfupLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
51 \def\bfuplatin{abcdefghijklmnopqrstuvwxyz}
52 \def\bfupGreek{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}
53 \def\bfupgreek{αβγδεζηθικλμνξοπρστυφχψω}
54 \def\bfitalatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
55 \def\bfitalatin{abcdefghijklmnopqrstuvwxyz}
56 \def\bfitalatin{ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ}
57 \def\bfitalatin{αβγδεζηθικλμνξοπρστυφχψω}
58 \providecommand\mathalphabet{\mathbf}
59 \def\testmath#1{
60   \makebox[\linewidth][l]{
61     \makebox[0pt][l]{$\mathalphabet{\csname up#1\endcsname}$}
62     \makebox[0pt][l]{$\mathalphabet{\csname it#1\endcsname}$}
63     \makebox[0pt][l]{$\csname bfup#1\endcsname$}
64     \makebox[0pt][l]{$\csname bfit#1\endcsname$}
65   }
66 \begin{document}
67 \setmathfont[Colour=2255FF55]{Cambria Math}
68 \parindent=0pt

```



```

69 \voffset=-1in
70 \hoffset=-1in
71 \setbox0=\vbox{
72 \testmath{Latin}\\
73 \testmath{latin}\\
74 \testmath{Greek}\\
75 \testmath{greek}}
76 \dimen0=\ht0
77 \advance\dimen0\dp0
78 \edef\papersize{papersize=\the\wd0,\the\dimen0}
79 \setbox255=\vbox{\special{\papersize}\box0}
80 \shipout\box255
81 \end{document}
82 </testbf>

```

# 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	\@um@upLatinfalse	127, 139
\'	355	\@um@upLatintrue	151
\:::	579	\@um@upNablafalse	133, 190
\::f	579	\@um@upNabltrue	145, 157, 188
\::n	579	\@um@upgreekfalse	126, 138
\@cclvi	224	\@um@upgreektrue	150
\@cdots	378	\@um@uplatinfalse	128, 140, 152
\@elt	462–466, 469, 473, 475	\@um@uppartialfalse	134, 197
\@empty	275,	\@um@uppartialtrue	146, 158, 195
	276, 313, 366, 507, 514, 529, 550, 555	\\	10–13, 17, 23–33, 72–74
\@gobble	300	\^	33, 288
\@ifnextchar	307	Numbers	
\@ifpackageloaded	381	\0	28
\@ii	513, 514, 516, 518, 521, 526		
\@input	305, 330	\_	17–20, 23–32
\@marker	526, 545	A	
\@nil	233, 289, 294, 303, 349, 526, 539–542	\A	29
\@preamblecmds	209	\a	30
\@sqrt	307	\addnolimits	468
\@tempa	123, 166, 186, 193,	\advance	30, 77
	200, 286, 302, 505, 517, 541, 543, 555	\alloc@	224
\@tempb	123, 124, 166, 167, 186, 187, 193,	\Alpha	606
	194, 200, 201, 505, 506, 544, 545, 550	\alpha	631
\@tempswafalse	515	\AtBeginDocument	377, 380, 604
\@tempwattrue	519, 522, 547, 552, 557, 562	\awint	464
\@um@bfliteraltrue	162, 183	B	
\@um@bfupGreekfalse	129, 168	\B	91
\@um@bfupGreektrue	141, 153, 173, 178	\beamer@suppressreplacementstrue	387
\@um@bfupLatinfalse	131, 170	\begin	19, 66
\@um@bfupLatintrue	143, 155, 175, 180	\begingroup	19, 230, 287, 301, 472
\@um@bfupgreekfalse	130, 142, 169, 174	\Beta	607
\@um@bfupgreektrue	154, 179	\beta	632
\@um@bfuplatinfalse	132, 171	\bfitGreek	56
\@um@bfuplatintrue	144, 156, 176, 181	\bfitgreek	57
\@um@fontspec@featuretrue	277	\bfitLatin	54
\@um@literaltrue	161	\bfitlatin	55
\@um@ot@math@true	305	\bfupGreek	52
\@um@upGreekfalse	125		
\@um@upGreektrue	137, 149		

<code>\bfupgreek</code> .....	53	<code>\DeclareRobustCommand</code> .....	307
<code>\bfuplatin</code> .....	50	<code>\DeclareSymbolFont</code> .....	328
<code>\bfuplatin</code> .....	51	<code>\def</code> 7–15, 23, 24, 28–121, 209, 210, 213,	
<code>\bgroup</code> .....	332	224, 226, 228, 279, 296, 302, 314,	
<code>\bool_new:N</code> .....	22	360, 371–376, 378, 461, 469, 471,	
<code>\bool_set_false:N</code> ...	135, 159, 163, 202	473, 480, 482, 539, 541–544, 606–661	
<code>\bool_set_true:N</code> .....	147, 204	<code>\define@choicekey</code> .....	
<code>\box</code> .....	32, 33, 79, 80, 369	..... 122, 166, 186, 193, 200, 505	
<b>C</b>			
<code>\C</code> .....	80, 106	<code>\define@cmdkey</code> .....	501–504
<code>\cdots</code> .....	378	<code>\define@key</code> .....	484
<code>\char_make_active:N</code> .....	355	<code>\Delta</code> .....	609
<code>\char_make_active:n</code> .....	231, 356	<code>\delta</code> .....	634
<code>\char_make_other:N</code> .....	288	<code>\dimen</code> .....	29–31, 76–78
<code>\chardef</code> .....	224	<code>\dimexpr</code> .....	211, 304, 363
<code>\Chi</code> .....	628	<code>\documentclass</code> .....	2, 37
<code>\chi</code> .....	653	<code>\dp</code> .....	30, 77, 365
<code>\cirfnint</code> .....	464	<b>E</b>	
<code>\clist_map_inline:Nn</code> .....	447	<code>\E</code> .....	92
<code>\clist_map_inline:nn</code> .....	442, 449	<code>\e</code> .....	100
<code>\clist_map_variable:NNn</code> .....	513	<code>\edef</code> .....	31, 78, 286, 322, 540, 541
<code>\clist_map_variable:nNn</code> .	570, 581, 587	<code>\egroup</code> .....	348
<code>\copy</code> .....	367	<code>\else</code> .....	162, 165, 171, 178, 190,
<code>\cs_gset:cpn</code> .....	238, 248	216, 219, 245, 249, 254, 259, 262,	
<code>\cs_gset:Npn</code> .....	251, 256	306, 320, 370–374, 377, 382, 388,	
<code>\cs_gset:Npx</code> .....	261	391, 396, 402, 414, 426, 433, 438,	
<code>\cs_gset_eq:NN</code> .....	357, 358	474, 487, 520, 525, 531, 549, 554, 559	
<code>\cs_if_exist:cF</code> .....	17	<code>\else:</code> .....	14
<code>\cs_new:Nn</code> .....	1,	<code>\encodingdefault</code> .....	329
16, 29, 311, 315, 331, 336, 345, 348,		<code>\end</code> .....	34, 81
353, 356, 359, 580, 586, 595, 598, 601		<code>\endcsname</code> ....	2, 17, 18, 23, 61–64,
<code>\cs_new:Npn</code> 46, 61, 77, 89, 104, 113, 118,		227, 233, 236, 239, 271, 278, 361, 457	
123, 128, 205, 225, 245, 250, 255, 271		<code>\endgroup</code> .....	19, 234, 299, 306, 478
<code>\cs_set:cpn</code> .....	18	<code>\Epsilon</code> .....	610
<code>\cs_set:Nn</code> ...	10, 365, 441, 446, 666, 681	<code>\epsilon</code> .....	635
<code>\cs_set:Npn</code> .....	579	<code>\Eta</code> .....	612
<code>\cs_set_eq:cN</code> .....	11	<code>\eta</code> .....	637
<code>\cs_set_eq:NN</code> ...	318, 319, 325, 326, 353	<code>\etex_iffontchar:D</code> .....	14
<code>\cs_set_protected:cpn</code> .....	21	<code>\ExecuteOptionsX</code> .....	207
<code>\cs_to_str:N</code> .....	248, 457	<code>\exp_args:Nnff</code> .....	579, 582, 589
<code>\csname</code> .....	2, 17, 18, 23, 61–64,	<code>\exp_args:No</code> .....	443, 450
227, 233, 236, 239, 271, 278, 361, 457		<code>\expandafter</code> .....	
<b>D</b>			
<code>\DeclareDocumentCommand</code> .....	273	. 23, 227, 235, 240, 242, 246, 456,	
<code>\DeclareMathVersion</code> .....	280	469, 516, 518, 521, 526, 540, 541, 545	
		<code>\ExplSyntaxOff</code> .....	391
		<code>\ExplSyntaxOn</code> .....	6

<b>F</b>	
\F .....	93
\f@size .....	278
\fi .....	164, 167, 174, 181, 184, 191, 198, 199, 205, 221, 222, 243, 264–268, 312, 327, 371–374, 379, 384, 385, 388, 393, 398, 405, 422, 429, 436, 437, 439, 452, 476, 492, 508, 523, 524, 527, 533, 535, 536, 548, 553, 558, 563–567
\fi: .....	14
\fint .....	464
\font .....	303
\fontdimen .....	211, 304, 362, 368
<b>G</b>	
\g .....	101
\g@addto@macro .....	456, 530, 532
\g_um_primekern_muskip ..	308, 309, 313
\g_um_vargreek_bool .....	22, 135, 147, 159, 163, 202, 204
\Gamma .....	608
\gamma .....	633
\gdef .....	289, 294
\ge .....	374
\geq .....	374
\glb@currsiz .....	274
\global .....	232, 235, 252, 253, 257, 258, 263, 291, 296, 507
\group_begin: .....	354
\group_end: .....	359
<b>H</b>	
\H .....	81, 94, 107
\h .....	64, 699, 703
\hbox .....	361
\hoffset .....	23, 70
\ht .....	29, 76, 364
<b>I</b>	
\I .....	95, 108
\if@temp .....	528
\if@um@bfliteral .....	15, 132, 386
\if@um@bfupGreek .....	16, 175, 399
\if@um@bfupgreek .....	17, 182, 406
\if@um@bfupLatin .....	18, 163, 389
\if@um@bfuplatin .....	19, 168, 394
\if@um@fontspec@feature .....	8, 485
\if@um@literal .....	10, 368
\if@um@ot@math@ .....	9
\if@um@upGreek .....	11, 373
\if@um@upgreek .....	12, 374
\if@um@upLatin .....	13, 371
\if@um@uplatin .....	14, 372
\if@um@upNabla .....	20, 375, 423
\if@um@uppartial .....	21, 380, 430
\ifbeamer@suppressreplacements ..	382
\ifcase .....	124, 167, 187, 194, 201, 506
\ifdim .....	304
\ifin@ .....	241, 247
\ifnum .....	448, 546, 551, 556, 560, 561
\ifx .....	214, 217, 229, 250, 255, 260, 313, 366, 474, 514, 517, 518, 521, 529, 545, 550, 555
\iiiint .....	462
\iiint .....	462
\iint .....	462
\in@ .....	240, 246
\int .....	462
\intBar .....	464
\intbar .....	464
\intcap .....	466
\intclockwise .....	463
\intcup .....	466
\intlarhk .....	465
\intx .....	465
\Iota .....	614
\iota .....	639
\itGreek .....	13, 48
\itgreek .....	14, 49
\itLatin .....	11, 46
\itlatin .....	12, 47
<b>K</b>	
\Kappa .....	615
\kappa .....	640
\kern .....	362, 368
<b>L</b>	
\L .....	96
\L_um_inc_num .....	588, 590, 591
\L_um_incr_num .....	571, 573, 575
\L_um_input_num .....	570, 573, 581, 583, 587, 590
\L_um_primecount_num ..	310, 333, 337, 347

<code>\l_um_script_features_tl</code> . . .	281, 292	<code>\mathrm</code> . . . . .	376
<code>\l_um_script_font_tl</code> . . . . .	283, 291	<code>\mathscr</code> . . . . .	90–102, 375
<code>\l_um_sscript_features_tl</code> . . .	282, 296	<code>\mathsf</code> . . . . .	114–116
<code>\l_um_sscript_font_tl</code> . . . . .	284, 295	<code>\mathsfit</code> . . . . .	119–121
<code>\Lambda</code> . . . . .	616	<code>\mathtt</code> . . . . .	124–126
<code>\lambda</code> . . . . .	641	<code>\mathup</code> . . . . .	47–59, 376
<code>\le</code> . . . . .	373	<code>\mddefault</code> . . . . .	329
<code>\left</code> . . . . .	481	<code>\mitAlpha</code> . . . . .	606
<code>\left@primitive</code> . . . . .	481, 482	<code>\mitalpha</code> . . . . .	631
<code>\leq</code> . . . . .	373	<code>\mitBeta</code> . . . . .	607
<code>\let</code> . . . . .	55, 225, 274–276, 291, 300, 316, 317, 323, 324, 481, 507	<code>\mitbeta</code> . . . . .	632
<code>\linewidth</code> . . . . .	16, 60	<code>\mitChi</code> . . . . .	628
<code>\lowercase</code> . . . . .	290, 295	<code>\mitchi</code> . . . . .	653
<code>\lowint</code> . . . . .	466	<code>\mitDelta</code> . . . . .	609
<b>M</b>			
<code>\M</code> . . . . .	97	<code>\mitdelta</code> . . . . .	634
<code>\m@th</code> . . . . .	361	<code>\mitEpsilon</code> . . . . .	610
<code>\makebox</code> . . . . .	16–18, 60–64	<code>\mitepsilon</code> . . . . .	635
<code>\mathaccent</code> . . . . .	260	<code>\mitEta</code> . . . . .	612
<code>\mathalpha</code> . . . . .	458, 574	<code>\miteta</code> . . . . .	637
<code>\mathalphabet</code> . . . . .	58, 61, 62	<code>\mitGamma</code> . . . . .	608
<code>\mathbb</code> . . . . .	78–87	<code>\mitgamma</code> . . . . .	633
<code>\mathbf</code> . . . . .	58, 129–131, 133–161, 164, 166, 169, 170, 172, 173, 176, 177, 179, 180, 183–189, 191–197, 201–204	<code>\mitIota</code> . . . . .	614
<code>\mathbffrac</code> . . . . .	246–248	<code>\mitiota</code> . . . . .	639
<code>\mathbfit</code> . . . . .	206–223	<code>\mitKappa</code> . . . . .	615
<code>\mathbfscr</code> . . . . .	251–253	<code>\mitkappa</code> . . . . .	640
<code>\mathbfsf</code> . . . . .	256–269	<code>\mitLambda</code> . . . . .	616
<code>\mathbfsfit</code> . . . . .	272–285	<code>\mitlambda</code> . . . . .	641
<code>\mathbfup</code> . . . . .	226–243	<code>\mitMu</code> . . . . .	617
<code>\mathbin</code> . . . . .	346	<code>\mitmu</code> . . . . .	642
<code>\mathcal</code> . . . . .	375	<code>\mitNu</code> . . . . .	618
<code>\mathchar@type</code> . . . . .	237, 251, 253, 256, 258, 261, 263, 271, 360	<code>\mitnu</code> . . . . .	643
<code>\mathclose</code> . . . . .	255	<code>\mitOmega</code> . . . . .	630
<code>\mathcode</code> . . . . .	232	<code>\mitomega</code> . . . . .	655
<code>\mathfrac</code> . . . . .	105–111	<code>\mitOmicron</code> . . . . .	620
<code>\mathgroup</code> . . . . .	224	<code>\mitomicron</code> . . . . .	645
<code>\mathinner</code> . . . . .	378	<code>\mitPhi</code> . . . . .	627
<code>\mathit</code> . . . . .	62–75	<code>\mitphi</code> . . . . .	652
<code>\mathop</code> . . . . .	229	<code>\mitPi</code> . . . . .	621
<code>\mathopen</code> . . . . .	250, 482	<code>\mitpi</code> . . . . .	646
<code>\mathord</code> . . . . .	357	<code>\mitPsi</code> . . . . .	629
		<code>\mitpsi</code> . . . . .	654
		<code>\mitRho</code> . . . . .	622
		<code>\mitrho</code> . . . . .	647
		<code>\mitSigma</code> . . . . .	624
		<code>\mitsigma</code> . . . . .	649
		<code>\mitTau</code> . . . . .	625

\mittau .....	650	<b>O</b>	
\mitTheta .....	613	\o .....	102
\mittheta .....	638	\oiint .....	462
\mitUpsilon .....	626	\oiint .....	462
\mitupsilon .....	651	\oint .....	462
\mitvarepsilon .....	656	\ointctrclockwise .....	463
\mitvarkappa .....	658	\Omega .....	630
\mitvarphi .....	659	\omega .....	655
\mitvarpi .....	661	\Omicron .....	620
\mitvarrho .....	660	\omicron .....	645
\mitvarsigma .....	648	\operator@font .....	371
\mitvarTheta .....	623	\or .....	136,
\mitvartheta .....	657		148, 160, 172, 177, 182, 189, 196, 203
\mitXi .....	619	<b>P</b>	
\mitxi .....	644	\P .....	83
\mitZeta .....	611	\PackageError .....	25, 488
\mitzeta .....	636	\PackageInfo .....	27
\mode_if_math:F .....	22	\PackageWarning .....	26
\mskip .....	313	\PackageWarningNoLine .....	383
\Mu .....	617	\papersize .....	31, 32, 78, 79
\mu .....	642	\parindent .....	21, 68
\muskip_gset:Nn .....	309	\peek_charcode_remove:NTF ...	338, 344
\muskip_new:N .....	308	\peek_meaning_remove:NTF .....	341
<b>N</b>		\Phi .....	627
\N .....	82	\phi .....	652
\new@mathgroup .....	224, 225	\Pi .....	621
\newcommand ....	25–27, 270, 336, 339,	\pi .....	646
	455, 468, 483, 512, 569, 605, 663,	\pointint .....	465
	691, 694, 697, 701, 705, 709, 713, 722	\prg_case_int:nnn .....	316
\newcounter .....	7	\prg_do_nothing: .....	11
\newfam .....	225	\prg_new_conditional:Nnn .....	13
\newif .....	8–21	\prg_replicate:nn .....	313
\noexpand .....	286, 475	\prg_return_false: .....	14
\nolimits .....	242	\prg_return_true: .....	14
\non@alpherr .....	23	\prg_stepwise_variable:nnnNn	571, 588
\npolint .....	465	\prime .....	353
\Nu .....	618	\primedouble .....	319
\nu .....	643	\primequadruple .....	325
\num_incr:N .....	337	\primesingle .....	312, 313, 317, 357
\num_new:N .....	310	\primetripel .....	322
\num_zero:N .....	333	\ProcessOptionsX .....	208
\number .....	583, 590, 591	\protect .....	491
\numexpr .....	551,	\providecommand .....	58
	556, 560, 561, 573, 575, 583, 590, 591	\ProvidesPackage .....	1
		\Psi .....	629
		\psi .....	654

Q		T	
\Q	84	\Tau	625
R		\tau	650
\R	85, 98, 109	\testmath	15, 25–28, 59, 72–75
\r@@t	360	\tf@size	289, 290
\raise	363	\the	31, 78
\relax	124, 167, 187, 194, 201, 211, 229, 232, 237, 248, 250–253, 255–258, 260, 261, 263, 271, 274, 303, 304, 366, 448, 506, 518, 521, 545, 546, 551, 556, 560, 561, 573, 575, 583, 590, 591	\Theta	613
\removenolimits	471	\theta	638
\RequirePackage	3–5	\theum@fam	322
\Rho	622	\thinmuskip	309
\rho	647	\tl_set:Nn	281–284
\rightarrow	372	\to	372
\rootbox	367	U	
\rppolint	464	\um@addto@mathmap	443, 450, 455
S		\um@backslash	517, 540
\scan_stop:	14, 362, 363	\um@Break	23
\scantokens	291, 296	\um@char@num@range	276, 447, 529, 530, 532
\scpolint	465	\um@char@range	275, 313, 366, 507, 510, 513
\scriptscriptstyle	217	\um@def@itGreek	373, 709
\scriptstyle	214	\um@def@itgreek	374, 722
\setbox	24, 32, 71, 79, 361	\um@def@itLatin	371, 694
\setkeys	285	\um@def@itlatin	372, 697
\setmainfont	5, 40	\um@def@numbers	367, 663
\SetMathCode	270, 354, 458, 572	\um@def@upGreek	373, 705
\setmathfont	20, 67, 273, 491	\um@def@upgreek	374, 713
\sf@size	290, 294	\um@def@upLatin	371, 691
\shipout	33, 80	\um@def@uplatin	372, 701
\Sigma	624	\um@firstchar	516, 541
\sigma	649	\um@firsttof	539–541
\space	23	\um@font	14, 215, 218, 303, 304, 362, 364, 365, 368
\special	32, 79	\um@fontdimen@percent	210, 215, 218, 364, 365
\sqint	465	\um@Loop	23
\sqrt	307, 480	\um@mathsymbol	226, 337, 342
\sqrtsign	307, 361	\um@mathsymbol@noparse	316, 336
\stepcounter	321	\um@mathsymbol@parse	323, 339
\string	233, 236, 238, 239, 341, 540, 541	\um@mversion	279, 280
\strip@pt	211	\um@nolimits	240, 461, 469, 477
\sumint	463	\um@PackageError	25
		\um@PackageInfo	27, 315, 341
		\um@PackageWarning	7, 26, 307
		\um@parse@range	526, 542
		\um@parse@term	340, 349, 512

$\backslash\mathrm{um@Pool}$ .....	23, 24	$\backslash\mathrm{um@usv@bfitvarpi}$ .....	109, 161, 197, 223, 413, 421
$\backslash\mathrm{um@radicals}$ .....	246, 480	$\backslash\mathrm{um@usv@bfitvarrho}$ .....	108, 160, 196, 222, 412, 420
$\backslash\mathrm{um@resolve@greek}$ .....	604	$\backslash\mathrm{um@usv@bfitvarTheta}$ .....	103, 153, 180, 215, 401, 404
$\backslash\mathrm{um@scaled@apply}$ .....	213, 362, 368	$\backslash\mathrm{um@usv@bfitvarthetaeta}$ .....	105, 157, 193, 219, 409, 417
$\backslash\mathrm{um@scanactivedef}$ .....	233, 287	$\backslash\mathrm{um@usv@bfLatin}$ .....	53, 133, 164, 211, 227, 231, 390, 392, 682
$\backslash\mathrm{um@scancharlet}$ .....	287, 303	$\backslash\mathrm{um@usv@bflatin}$ .....	54, 55, 135, 169, 212, 228, 232, 395, 397, 683
$\backslash\mathrm{um@set@mathsymbol}$ .....	227, 228	$\backslash\mathrm{um@usv@bfNabla}$ .....	112, 143, 236, 424, 427
$\backslash\mathrm{um@setmathcode}$ .....	376, 378, 381, 383, 390, 392, 395, 397, 400, 401, 403, 404, 407–413, 415–421, 424, 425, 427, 428, 431, 432, 434, 435, 569, 664, 667–679, 682–689, 692, 695, 698, 699, 702, 703, 706, 707, 710, 711, 714–720, 723–729	$\backslash\mathrm{um@usv@bfnum}$ .....	52, 129, 206, 226, 246, 251, 256, 272
$\backslash\mathrm{um@symfont}$ .....	314, 322, 328, 337, 342, 354, 361, 443, 450, 574	$\backslash\mathrm{um@usv@bfpartial}$ .....	118, 145, 237, 431, 434
$\backslash\mathrm{um@usv@bbLatin}$ .....	38, 79	$\backslash\mathrm{um@usv@bfscrLatin}$ .....	64, 252
$\backslash\mathrm{um@usv@bblatin}$ .....	39, 87	$\backslash\mathrm{um@usv@bfscrLatin}$ .....	65, 253
$\backslash\mathrm{um@usv@bbnum}$ .....	37, 78	$\backslash\mathrm{um@usv@bfsgreek}$ .....	69, 259
$\backslash\mathrm{um@usv@bfDigamma}$ .....	85, 144	$\backslash\mathrm{um@usv@bfsgreek}$ .....	70, 260
$\backslash\mathrm{um@usv@bfdigamma}$ .....	92, 152	$\backslash\mathrm{um@usv@bfsgreek}$ .....	73, 275
$\backslash\mathrm{um@usv@bffrakLatin}$ .....	62, 247	$\backslash\mathrm{um@usv@bfsgreek}$ .....	74, 276
$\backslash\mathrm{um@usv@bffrakLatin}$ .....	63, 248	$\backslash\mathrm{um@usv@bfsgreek}$ .....	71, 273
$\backslash\mathrm{um@usv@bfGreek}$ .....	56, 137, 176, 213, 229, 233, 400, 403, 686	$\backslash\mathrm{um@usv@bfsgreek}$ .....	72, 274
$\backslash\mathrm{um@usv@bfgreek}$ .....	57, 139, 183, 214, 230, 234, 407, 415, 687	$\backslash\mathrm{um@usv@bfsgreek}$ .....	115, 278, 425, 428
$\backslash\mathrm{um@usv@bfitGreek}$ .....	60, 138, 179, 209, 213, 400, 403, 688	$\backslash\mathrm{um@usv@bfsgreek}$ .....	121, 279, 432, 435
$\backslash\mathrm{um@usv@bfitgreek}$ .....	61, 140, 191, 210, 214, 407, 415, 689	$\backslash\mathrm{um@usv@bfsgreek}$ .....	67, 257
$\backslash\mathrm{um@usv@bfith}$ .....	102, 141, 173	$\backslash\mathrm{um@usv@bfsgreek}$ .....	68, 258
$\backslash\mathrm{um@usv@bfitLatin}$ .....	58, 134, 166, 207, 211, 390, 392, 684	$\backslash\mathrm{um@usv@bfsgreek}$ .....	114, 425, 428
$\backslash\mathrm{um@usv@bfitlatin}$ .....	59, 136, 172, 208, 212, 395, 397, 685	$\backslash\mathrm{um@usv@bfsgreek}$ .....	66
$\backslash\mathrm{um@usv@bfitNabla}$ .....	113, 154, 216, 424, 427	$\backslash\mathrm{um@usv@bfsgreek}$ .....	120, 432, 435
$\backslash\mathrm{um@usv@bfitpartial}$ .....	119, 155, 217, 431, 434	$\backslash\mathrm{um@usv@bfsgreek}$ .....	101, 170
$\backslash\mathrm{um@usv@bfitvarepsilon}$ .....	104, 156, 192, 218, 408, 416	$\backslash\mathrm{um@usv@bfsgreek}$ .....	55
$\backslash\mathrm{um@usv@bfitvarkappa}$ .....	106, 158, 194, 220, 410, 418	$\backslash\mathrm{um@usv@bfsgreek}$ .....	86, 146, 184, 238, 408, 416
$\backslash\mathrm{um@usv@bfitvarphi}$ .....	107, 159, 195, 221, 411, 419	$\backslash\mathrm{um@usv@bfsgreek}$ .....	88, 148, 186, 240, 410, 418
		$\backslash\mathrm{um@usv@bfsgreek}$ .....	89, 149, 187, 241, 411, 419
		$\backslash\mathrm{um@usv@bfsgreek}$ .....	91, 151, 189, 243, 413, 421
		$\backslash\mathrm{um@usv@bfsgreek}$ .....	90, 150, 188, 242, 412, 420
		$\backslash\mathrm{um@usv@bfsgreek}$ .....	84, 142, 177, 235, 401, 404
		$\backslash\mathrm{um@usv@bfsgreek}$ .....	87, 147, 185, 239, 409, 417



$\backslash\mathrm{um@usv@Digamma}$ . . . . .	76, 130, 144	$\backslash\mathrm{um@usv@sfnum}$ . . . . .	44, 114, 119
$\backslash\mathrm{um@usv@digamma}$ . . . . .	83, 131, 152	$\backslash\mathrm{um@usv@ttLatin}$ . . . . .	50, 125
$\backslash\mathrm{um@usv@frakLatin}$ . . . . .	42, 105, 247	$\backslash\mathrm{um@usv@ttlLatin}$ . . . . .	51, 126
$\backslash\mathrm{um@usv@frakLatin}$ . . . . .	43, 111, 248	$\backslash\mathrm{um@usv@ttnum}$ . . . . .	49, 124
$\backslash\mathrm{um@usv@itGreek}$ 35, 49, 65, 138, 176, 179, 209, 229, 259, 275, 674, 706, 710		$\backslash\mathrm{um@usv@upGreek}$ 33, 49, 65, 137, 176, 179, 209, 229, 259, 275, 672, 706, 710	
$\backslash\mathrm{um@usv@itgreek}$ . . . . 36, 50, 66, 140, 183, 191, 210, 230, 260, 276, 714, 723		$\backslash\mathrm{um@usv@upgreek}$ 34, 50, 66, 139, 183, 191, 210, 230, 260, 276, 675, 714, 723	
$\backslash\mathrm{um@usv@ith}$ . . . . .		$\backslash\mathrm{um@usv@upLatin}$ 29, 47, 62, 79, 90, 105, 115, 120, 125, 133, 164, 166, 207, 227, 247, 252, 257, 273, 667, 692, 695	
$\backslash\mathrm{um@usv@itLatin}$ 31, 47, 62, 79, 90, 105, 115, 120, 125, 134, 164, 166, 207, 227, 247, 252, 257, 273, 668, 692, 695		$\backslash\mathrm{um@usv@upLatin}$ 30, 48, 63, 87, 99, 111, 116, 121, 126, 135, 169, 172, 208, 228, 248, 253, 258, 274, 671, 698, 702	
$\backslash\mathrm{um@usv@itLatin}$ 32, 48, 63, 87, 99, 111, 116, 121, 126, 136, 169, 172, 208, 228, 248, 253, 258, 274, 669, 698, 702		$\backslash\mathrm{um@usv@varepsilon}$ . . 54, 70, 77, 146, 184, 192, 218, 238, 264, 280, 715, 724	
$\backslash\mathrm{um@usv@itNabla}$ . . . 51, 67, 111, 154, 203, 216, 236, 262, 278, 376, 378, 677		$\backslash\mathrm{um@usv@varkappa}$ . . . 56, 72, 79, 148, 186, 194, 220, 240, 266, 282, 717, 726	
$\backslash\mathrm{um@usv@itpartial}$ . 52, 68, 117, 155, 204, 217, 237, 263, 279, 381, 383, 679		$\backslash\mathrm{um@usv@varphi}$ . . . . 57, 73, 80, 149, 187, 195, 221, 241, 267, 283, 718, 727	
$\backslash\mathrm{um@usv@itvarepsilon}$ 54, 70, 95, 156, 184, 192, 218, 238, 264, 280, 715, 724		$\backslash\mathrm{um@usv@varpi}$ . . . . . 59, 75, 82, 151, 189, 197, 223, 243, 269, 285, 720, 729	
$\backslash\mathrm{um@usv@itvarkappa}$ . . 56, 72, 97, 158, 186, 194, 220, 240, 266, 282, 717, 726		$\backslash\mathrm{um@usv@varrho}$ . . . . . 58, 74, 81, 150, 188, 196, 222, 242, 268, 284, 719, 728	
$\backslash\mathrm{um@usv@itvarphi}$ . . . 57, 73, 98, 159, 187, 195, 221, 241, 267, 283, 718, 727		$\backslash\mathrm{um@usv@varTheta}$ 53, 69, 75, 142, 177, 180, 215, 235, 261, 277, 673, 707, 711	
$\backslash\mathrm{um@usv@itvarpi}$ . . . 59, 75, 100, 161, 189, 197, 223, 243, 269, 285, 720, 729		$\backslash\mathrm{um@usv@vartheta}$ . . . 55, 71, 78, 147, 185, 193, 219, 239, 265, 281, 716, 725	
$\backslash\mathrm{um@usv@itvarrho}$ . . . 58, 74, 99, 160, 188, 196, 222, 242, 268, 284, 719, 728		$\backslash\mathrm{um@zf@feature}$ . . . . . 483, 495, 498	
$\backslash\mathrm{um@usv@itvarTheta}$ . . . . . 53, 69, 94, 153, 177, 180, 215, 235, 261, 711		$\backslash\mathrm{um\_config\_mathbb}$ : . . . . . 77	
$\backslash\mathrm{um@usv@itvartheta}$ . . 55, 71, 96, 157, 185, 193, 219, 239, 265, 281, 716, 725		$\backslash\mathrm{um\_config\_mathbf}$ : . . . . . 128	
$\backslash\mathrm{um@usv@Nabla}$ . . . . . 51, 67, 110, 143, 201, 216, 236, 262, 278, 376, 378, 676		$\backslash\mathrm{um\_config\_mathbbfrak}$ : . . . . . 245	
$\backslash\mathrm{um@usv@num}$ . . . . 28, 78, 114, 119, 124, 129, 206, 226, 246, 251, 256, 272, 664		$\backslash\mathrm{um\_config\_mathbf fit}$ : . . . . . 205	
$\backslash\mathrm{um@usv@partial}$ . . . 52, 68, 116, 145, 202, 217, 237, 263, 279, 381, 383, 678		$\backslash\mathrm{um\_config\_mathbfscr}$ : . . . . . 250	
$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 40, 90		$\backslash\mathrm{um\_config\_mathbfssf}$ : . . . . . 255	
$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 41, 99		$\backslash\mathrm{um\_config\_mathbfssfit}$ : . . . . . 271	
$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 47, 120		$\backslash\mathrm{um\_config\_mathbfsup}$ : . . . . . 225	
$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 48, 121		$\backslash\mathrm{um\_config\_mathfrak}$ : . . . . . 104	
$\backslash\mathrm{um@usv@sLatin}$ . . . . . 45, 115		$\backslash\mathrm{um\_config\_mathit}$ : . . . . . 61	
$\backslash\mathrm{um@usv@sLatin}$ . . . . . 46, 116		$\backslash\mathrm{um\_config\_mathscr}$ : . . . . . 89	
		$\backslash\mathrm{um\_config\_mathsf}$ : . . . . . 113	
		$\backslash\mathrm{um\_config\_mathsf fit}$ : . . . . . 118	
		$\backslash\mathrm{um\_config\_maththt}$ : . . . . . 123	
		$\backslash\mathrm{um\_config\_mathup}$ : . . . . . 46	
		$\backslash\mathrm{um\_glyph\_if\_exist:n}$ . . . . . 13	

<code>\um_glyph_if_exist:nTF</code> .....	<code>\um_setup_bf_literals:</code> .....
..... 2, 13, 319, 322, 325	..... 387, 681
<code>\um_init_alphabet:n</code> .....	<code>\um_setup_literals:</code> .....
..... 10, 319	..... 369, 666
<code>\um_make_mathactive:nNN</code> .....	<code>\um_setup_math_alphabet:n</code> ...
..... 357, 359	..... 1, 30–44
<code>\um_mathmap:Nnn</code> .....	<code>\um_setup_mathactives:</code> .....
..... 317, 324, 582, 589	..... 332, 356
<code>\um_mathmap_noparse:Nnn</code> .....	<code>\um_setup_mathup:</code> .....
..... 317, 441	..... 371
<code>\um_mathmap_parse:Nnn</code> .....	<code>\unicodemathgobble</code> .....
..... 324, 446	..... 300
<code>\um_maybe_init_alphabet:n</code> ..	<code>\UnicodeMathSymbol</code> .....
..... 3, 319, 326	..... 302, 316, 323
<code>\um_nprimes:n</code> ...	<code>\unless</code> .....
..... 311, 319, 322, 325, 328	..... 514
<code>\um_nprimes_select:n</code> .....	<code>\updefault</code> .....
..... 315, 347	..... 329
<code>\um_prepare_alph:n</code> .....	<code>\upGreek</code> .....
..... 4, 16	..... 9, 44
<code>\um_remap_symbol:nnn</code> .....	<code>\upgreek</code> .....
..... 318, 325, 346	..... 10, 45
<code>\um_remap_symbol_noparse:nnn</code> .....	<code>\upint</code> .....
..... 318, 350, 353	..... 466
<code>\um_remap_symbol_parse:nnn</code> ..	<code>\upLatin</code> .....
..... 325, 348	..... 8, 42
<code>\um_remap_symbols:</code> .....	<code>\uplatin</code> .....
..... 331, 345	..... 7, 43
<code>\um_scanprime:</code> ..	<code>\Upsilon</code> .....
..... 331, 341, 353, 357, 358	..... 626
<code>\um_scanprime_collect:</code> .....	<code>\upsilon</code> .....
..... 334, 336, 339, 342, 345	..... 651
<code>\um_set_mathalph_range:Nnn</code> .....	<code>\use:c</code> .....
..... 586	..... 5, 19, 25
<code>\um_set_mathalph_range:nNnn</code> .....	<code>\use_none:n</code> .....
..... 586, 596, 599, 602	..... 326
<code>\um_set_mathalphabet_char:Nnn</code> ...	<code>\usepackage</code> .....
..... 51–59,	..... 3, 4, 6, 38, 39, 41
64, 67–75, 80–86, 91–98, 100–102,	
106–110, 130, 131, 141–161,	
170, 173, 177, 180, 184–189,	
192–197, 201–204, 215–223,	
235–243, 261–269, 277–285, 580	
<code>\um_set_mathalphabet_char:Nnnn</code> ..	
..... 579	
<code>\um_set_mathalphabet_greek:Nnn</code> ..	
..... 49, 50, 65, 66, 137–140, 176, 179,	
183, 191, 209, 210, 213, 214, 229,	
230, 233, 234, 259, 260, 275, 276, 601	
<code>\um_set_mathalphabet_latin:Nnn</code> ..	
..... 47, 48,	
62, 63, 79, 87, 90, 99, 105, 111, 115,	
116, 120, 121, 125, 126, 133–136,	
164, 166, 169, 172, 207, 208, 211,	
212, 227, 228, 231, 232, 247, 248,	
252, 253, 257, 258, 273, 274, 598	
<code>\um_set_mathalphabet_numbers:Nnn</code> .	
..... 78, 114, 119, 124,	
129, 206, 226, 246, 251, 256, 272, 595	
<code>\um_setup_alphabets:</code> .....	
..... 29, 334	
<code>\um_setup_alphanum:</code> .....	
..... 333, 365	

## V

<code>\varepsilon</code> .....	..... 656
<code>\varkappa</code> .....	..... 658
<code>\varointclockwise</code> .....	..... 463
<code>\varphi</code> .....	..... 659
<code>\varpi</code> .....	..... 661
<code>\varrho</code> .....	..... 660
<code>\varsigma</code> .....	..... 648
<code>\varTheta</code> .....	..... 623
<code>\vartheta</code> .....	..... 657
<code>\vbox</code> .....	..... 24, 32, 71, 79
<code>\voffset</code> .....	..... 22, 69

## W

<code>\wd</code> .....	..... 31, 78
------------------------	--------------

## X

<code>\xdef</code> .....	..... 477, 510
<code>\XeTeXdelcode</code> .....	..... 252, 257
<code>\XeTeXdelim�ter</code> .....	..... 251, 256
<code>\XeTeXmathaccent</code> .....	..... 261
<code>\XeTeXmathchardef</code> .....	..... 235, 360
<code>\XeTeXmathcode</code> .....	..... 253, 258, 263, 271
<code>\XeTeXmathcodenum</code> .....	..... 363
<code>\XeTeXradical</code> .....	..... 248
<code>\Xi</code> .....	..... 619
<code>\xi</code> .....	..... 644
<code>\XKV@rm</code> .....	..... 299

<b>Z</b>		
\Z .....	86, 110	\zetaa ..... 636
\z@ .....	361, 364, 365, 369	\zf@family ..... 329
\Zeta .....	611	\zf@fontspec ..... 286
		\zf@update@ff ..... 496, 499