

Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2008/01/21 v0.3

Abstract

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

Contents

1	Introduction	2	6	Fundamentals	13
2	Specification	2	6.1	Enlarging the number of maths families	13
2.1	Using multiple fonts	3	6.2	<code>\DeclareMathSymbol</code> for unicode ranges	13
2.2	Script and scriptscript fonts/features	3	6.3	User interface to <code>\DeclareSymbolFont</code>	15
3	Maths input	3	6.4	Maths alphabets' character mapping	18
4	Package options	4	6.5	(Big) operators	23
4.1	Math 'style'	4	6.6	Radicals	27
4.2	Bold switching	5	6.7	Delimiters	27
4.3	Other upright vs. italic symbols	5	6.8	Maths accents	30
I	The unicode-math package	6	7	Font features	31
5	Things we need	7	7.1	OpenType maths font features	31
5.1	Package options	11	7.2	Script and scriptscript font options	31
5.2	Overcoming <code>\@on-</code> <code>lypreamble</code>	13	7.3	Range processing	31
			II	Maths alphabets map-	

ping definitions	37	A.2 Detailed code investigation	49
7.4 Bold alphabets' character mappings	41	A.3 Maths symbols	49
		A.4 Delimiters	51
		A.5 Symbol fonts	54
III stix table data extraction	47	IV Some manner of unit testing	58
A Documenting maths support in the NFSS	48	B The regular weight alphabets	58
A.1 Overview	48	C The bold alphabets	59

1 Introduction

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

As of X_YTeX v.0.995, maths characters can be accessed in unicode ranges. Now, a proper method must be invented for real unicode maths support. Before any code is written, I'm writing a specification in order to work out what is required. Fairly significant pieces of the NFSS may have to be re-written, and I'm a little unsure where to start.

2 Specification

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

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

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

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

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

Finally, maths versions must also be provided for. While I guess version selection in L^AT_EX 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

$$[\langle font\ features \rangle] \{ \langle font\ name \rangle \}$$
follow from my fontspec package, and therefore any additional $\langle font\ features \rangle$ 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. 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:

$$\setmathfont[Range=\langle unicode\ range \rangle, \langle font\ features \rangle] \{ \langle font\ name \rangle \}$$

where $\langle unicode\ range \rangle$ 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 $\langle unicode\ range \rangle$ 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 no doubt use entirely separate fonts. Both of these options must be taken into account. I hope this will be mostly automatic from the users’ points of view. The `+ssty` feature can be detected and applied automatically, and appropriate optical size information embedded in the fonts will ensure this latter case. Fine tuning should be possible automatically with fontspec options. We might have to wait until MnMath, for example, before we really know.

3 Maths input

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

: TODO : describe alphabet inputs

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

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

4 Package options

4.1 Math ‘style’

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

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

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

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

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

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

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

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

4.2 Bold switching

Similar as in the previous section, ISO standards differ somewhat to \TeX 's conventions (and classical typesetting) for ‘boldness’ in mathematics. In the past, it has been customary to use bold *upright* letters to denote things like vectors and matrices. For example, $\mathbf{M} = (M_x, M_y, M_z)$. Presumably, this was due to the relatively scarcity of bold italic fonts in the pre-digital typesetting era. It has been suggested that *italic* bold symbols are used nowadays instead.

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

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

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

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

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

4.3 Other upright vs. italic symbols

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

Table 3: The various forms of nabla.

Description		Glyph
Upright	Serif	∇
	Bold serif	∇
	Bold sans	∇
Italic	Serif	∇
	Bold serif	∇
	Bold sans	∇

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεεζηθθικκλμνξοπωρρςστυφφχψω

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεεζηθθικκλμνξοπωρρςστυφφχψω

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ
αβγδεεζηθθικκλμνξοπωρρςστυφφχψω

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

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

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

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

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΘΣΤΥΦΧΨΩ
αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ

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

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

File I

The unicode-math package

This is the package.

```
1 \ProvidesPackage{unicode-math}
2 [2008/01/21 v0.3 Unicode maths in XeLaTeX]
```

5 Things we need

Packages

```
3 \RequirePackage{fontspec}
```

Counters and conditionals

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

For math-style:

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

For bold-style:

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

For nabla and partial:

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

Programming niceties

`\def@cn`

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

`\um@Loop` See Kees van der Laan's various articles on T_EX programming:

```
\um@Break 21 \def\um@Loop#1\um@Pool{#1\um@Loop#1\um@Pool}
22 \def\um@Break#1\um@Pool{}
```

`\um@FOR` A simple 'for' loop implemented with the above. Takes a (predefined) counter control sequence and increments it between two integers, iterating as we go.

```
23 \long\def\um@FOR#1=[#2:#3]\do#4{%
24   #1=#2\relax
25   \um@Loop #4%
26   \advance#1\@ne
27   \ifnum#1>#3\relax
28     \expandafter\um@Break
29   \fi
30   \um@Pool}
```

`g/h/i/j/k/l/m/`

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

Shortcuts

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


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

```

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

```

Bold:

```

58 \def\um@usv@bfnum{"1D7CE}
59 \def\um@usv@bfLatin{"1D400}
60 \def\um@usv@bflatin{"1D41A}
61 \def\um@usv@bfGreek{"1D6A8}
62 \def\um@usv@bfgreek{"1D6C2}
63 \def\um@usv@bfitLatin{"1D468}
64 \def\um@usv@bfitlatin{"1D482}
65 \def\um@usv@bfitGreek{"1D71C}
66 \def\um@usv@bfitgreek{"1D736}
67 \def\um@usv@bffrakLatin{"1D56C}
68 \def\um@usv@bffraklatin{"1D586}
69 \def\um@usv@bfscrLatin{"1D4D0}
70 \def\um@usv@bfscrLatin{"1D4EA}
71 \def\um@usv@bfsfnun{"1D7EC}
72 \def\um@usv@bfsflatin{"1D5D4}
73 \def\um@usv@bfsflatin{"1D5EE}
74 \def\um@usv@bfsfGreek{"1D756}

```

¹'u.s.v.' stands for 'unicode scalar value'.

```

75 \def\um@usv@bfsfgreek{"1D770}
76 \def\um@usv@bfsfitLatin{"1D63C}
77 \def\um@usv@bfsfitlatin{"1D656}
78 \def\um@usv@bfsfitGreek{"1D790}
79 \def\um@usv@bfsfitgreek{"1D7AA}

```

Greek variants:

```

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

```

Bold:

```

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

```

Italic Greek variants:

```

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

```

Bold:

```

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

```

Nabla:

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

Partial:

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

5.1 Package options

xkeyval's package support is used here.

math-style

```
127 \define@choicekey*{unicode-math.sty}  
128   {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{%  
129   \ifcase\@tempb\relax  
130     \@um@upGreekfalse  
131     \@um@upgreekfalse  
132     \@um@upLatinfalse  
133     \@um@uplatinfalse  
134     \@um@bfupGreekfalse  
135     \@um@bfupgreekfalse  
136     \@um@bfupLatinfalse  
137     \@um@bfuplatinfalse  
138     \@um@upNablafalse  
139     \@um@uppartialfalse  
140   \or  
141     \@um@upGreektrue  
142     \@um@upgreekfalse  
143     \@um@upLatinfalse  
144     \@um@uplatinfalse  
145     \@um@bfupGreektrue  
146     \@um@bfupgreekfalse  
147     \@um@bfupLatintrue  
148     \@um@bfuplatintrue  
149     \@um@upNablatrue  
150     \@um@uppartialtrue  
151   \or
```

```

152 \um@upGreektrue
153 \um@upgreektrue
154 \um@upLatintrue
155 \um@uplatinfalse
156 \um@bfupGreektrue
157 \um@bfupgreektrue
158 \um@bfupLatintrue
159 \um@bfuplatintrue
160 \um@upNablatrue
161 \um@uppartialtrue
162 \or
163 \um@literaltrue
164 \fi}

```

bold-style

```

165 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,french,literal}{%
166 \ifcase\@tempb\relax
167 \um@bfupGreekfalse
168 \um@bfupgreekfalse
169 \um@bfupLatinfalse
170 \um@bfuplatinfalse
171 \or
172 \um@bfupGreektrue
173 \um@bfupgreekfalse
174 \um@bfupLatintrue
175 \um@bfuplatintrue
176 \or
177 \um@bfupGreektrue
178 \um@bfupgreektrue
179 \um@bfupLatintrue
180 \um@bfuplatintrue
181 \or
182 \um@bfliteraltrue
183 \fi}

```

Symbol obliqueness

```

184 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright,italic}{%
185 \ifcase\@tempb\relax
186 \um@upNablatrue
187 \or
188 \um@upNablafalse
189 \fi}

190 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{%
191 \ifcase\@tempb\relax
192 \um@uppartialtrue

```

```

193 \or
194 \um@uppartialfalse
195 \fi}
196 \ExecuteOptionsX{math-style=iso}
197 \ProcessOptionsX

```

5.2 Overcoming \@onlypreamble

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

```

198 \def\@preamblecmds{}

```

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 ltfssbas.dtx) we want to redefine

```

199 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cclvi}
200 \let\newfam\new@mathgroup

```

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

6.2 \DeclareMathSymbol for unicode ranges

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

```

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

```

The final macros that actually define the maths symbol with X_YTeX 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.

```

203 \def\um@set@mathsymbol#1#2#3#4{%

```

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

```
204 \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`.

```
205 \begingroup
206 \catcode#4=\active
207 \global\mathcode#4="8000\relax
208 \um@scanactivedef#4\@nil{\csname\string#2@op\endcsname}%
209 \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`.

```
210 \expandafter\global\expandafter\XeTeXmathchardef
211 \csname\string#2@sym\endcsname
212 ="\mathchar@type#3 #1 #4\relax
```

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

```
213 \expandafter\gdef\csname\string#2@op\endcsname{%
214 \csname\string#2@sym\endcsname
215 \expandafter\in@\expandafter#2\expandafter{\um@nolimits}%
216 \ifin@\expandafter\nolimits\fi}%
```

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

```
217 \else
```

Radicals

```
218 \expandafter\in@\expandafter#2\expandafter{\um@radicals,}%
219 \ifin@
220 \gdef#2{\XeTeXradical#1 #4\relax}%
221 \else
```

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

```
222 \ifx\mathopen#3\relax
223 \gdef#2{\XeTeXdelimiter "\mathchar@type#3 #1 #4}%
224 \global\XeTeXdelcode#4=#1 #4\relax
225 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
226 \else
```

```

227 \ifx\mathclose#3\relax
228 \gdef#2{\XeTeXdelimeter "\mathchar@type#3 #1 #4}%
229 \global\XeTeXdelcode#4=#1 #4\relax
230 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
231 \else

```

Accents

```

232 \ifx\mathaccent#3\relax
233 \xdef#2{\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}%
234 \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.

```

235 \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
236 \fi
237 \fi
238 \fi
239 \fi
240 \fi}

```

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

```

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

```

A

```

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

```

6.3 User interface to `\DeclareSymbolFont`

Here's the simplest usage:

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

```

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

```

TODO And an example of the Range feature:

$(a, a, \textcolor{blue}{a}, a, \alpha)$

```

\setmathfont{Cambria Math}
$(a, \textit{a}, \mathbf{a}, \bfa, \alpha)$

```

An interesting example of the Range feature:

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

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

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

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

```
243 \newcommand\setmathfont[2][]{%
```

Init

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

```
244 \let\glb@currsizel\relax
```

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

```
245 \let\um@char@range\@empty
246 \let\um@char@num@range\@empty
247 \@um@initfalse
```

- Tell fontspec that maths font features are actually allowed.

```
248 \@um@fontspec@featuretrue
```

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

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

- Set the name of the math version being defined

```
250 \def\um@mversion{normal}%
251 \DeclareMathVersion{\um@mversion}%
```


Define default font features for the script and scriptscript font.

```

252 \def\um@ScriptFeatures{ScriptStyle}%
253 \def\um@ScriptScriptFeatures{ScriptScriptStyle}%
254 \def\um@ScriptFont{#2}%
255 \def\um@ScriptScriptFont{#2}%

```

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

```

256 \setkeys*[um]{options}{#1}%
257 %\rule{1.5ex}{1.5ex}%
258 \edef\@tempa{\noexpand\zf@fontspec{%
259   Script=Math,SizeFeatures={%
260     {Size=\tf@size-},%
261     {Size=\sf@size-\tf@size,%
262       Font=\um@ScriptFont,%
263       \um@ScriptFeatures},%
264     {Size=-\sf@size,%
265       Font=\um@ScriptScriptFont,%
266       \um@ScriptScriptFeatures}},%
267   \XKV@rm}{#2}}\@tempa
268 %\rule{1.5ex}{1.5ex}%

```

Probably want to check there that we're not creating multiple symbol fonts with the same NFSS declaration. On that note, fontspec doesn't seem to be keeping track of that, either :C (check that out!)

```

269 \ifx\um@char@range\@empty
270   \def\um@symfont{\um@allsym}%
271   \um@PackageInfo{Defining the default maths font as '#2'}%
272   \let\UnicodeMathSymbol\um@mathsymbol@noparse
273 \else
274   \stepcounter{um@fam}%
275   \edef\um@symfont{\um@fam\theum@fam}%
276   \let\UnicodeMathSymbol\um@mathsymbol@parse
277 \fi
278 \DeclareSymbolFont{\um@symfont}
279   {\encodingdefault}{\zf@family}{\mddefault}{\updefault}%

```

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

```

280 \if@um@init
281   \input unicode-math.tex\relax
282   \input unicode-math-add.tex\relax
283 \else
284   \unless\ifx\um@char@range\@empty
285     \input unicode-math.tex\relax
286     \input unicode-math-add.tex\relax

```

```

287 \fi
288 \fi

```

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

```

289 \ifx\um@char@range\@empty
290 \let\um@mathbb\@empty
291 \let\um@mathbf\@empty
292 \let\um@mathfrak\@empty
293 \let\um@mathup\@empty
294 \let\um@mathit\@empty
295 \let\um@mathscr\@empty
296 \let\um@mathsf\@empty
297 \let\um@mathsfitt\@empty
298 \let\um@mathtt\@empty
299 \let\um@mathbf\@empty
300 \let\um@mathbfup\@empty
301 \let\um@mathbfitt\@empty
302 \let\um@mathbffrak\@empty
303 \let\um@mathbfscr\@empty
304 \let\um@mathbfsf\@empty
305 \let\um@mathbfsfitt\@empty
306 \let\um@setsinglemathalph\um@mathmap@noparse
307 \else
308 \let\um@setsinglemathalph\um@mathmap@parse
309 \fi

```

6.4 Maths alphabets' character mapping

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

Numbers:

```

310 \ifx\um@char@range\@empty
311 \um@def@numbers

```

Normal weight

```

312 \if@um@literal
313 \um@setmathcode[26]{\um@usv@upLatin}{\um@usv@upLatin}%
314 \um@setmathcode[26]{\um@usv@itLatin}{\um@usv@itLatin}%
315 \um@setmathcode[26]{\um@usv@itlatin}{\um@usv@itlatin}%
316 \um@setmathcode{\um@usv@ith}{\um@usv@ith}%
317 \um@setmathcode[26]{\um@usv@uplatin}{\um@usv@uplatin}%
318 \um@setmathcode[25]{\um@usv@upGreek}{\um@usv@upGreek}%

```

```

319 \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}%
320 \um@setmathcode[25]{\um@usv@itGreek}{\um@usv@itGreek}%
321 \um@setmathcode[25]{\um@usv@upgreek}{\um@usv@upgreek}%

```

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

```

322 \um@setmathcode{\um@usv@Nabla}{\um@usv@Nabla}%
323 \um@setmathcode{\um@usv@itNabla}{\um@usv@itNabla}%
324 \um@setmathcode{\um@usv@partial}{\um@usv@partial}%
325 \um@setmathcode{\um@usv@itpartial}{\um@usv@itpartial}%
326 \else

```

Latin letters:

```

327 \if@um@upLatin\um@def@upLatin\else\um@def@itLatin\fi
328 \if@um@uplatin\um@def@uplatin\else\um@def@itlatin\fi

```

0123456789	$\setmathfont{\text{Cambria Math}}$
<i>ABCDEFGHIJKLMNOPQRSTUVWXYZ</i>	$\$0123456789\$ \backslash$
<i>abcdefghijklmnopqrstuvwxyz</i>	$\$ABCDEFGHIJKLMNOPQRSTUVWXYZ\$ \backslash$
	$\$abcdefghijklmnopqrstuvwxyz\$ \backslash$

Normal weight Greek, italic uppercase and lowercase respectively:

```

329 \if@um@upGreek\um@def@upGreek\else\um@def@itGreek\fi
330 \if@um@upgreek\um@def@upgreek\else\um@def@itgreek\fi

```

Nabla and partial:

```

331 \if@um@upNabla
332 \um@setmathcode{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
333 \else
334 \um@setmathcode{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}%
335 \fi
336 \if@um@uppartial
337 \um@setmathcode{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
338 \else
339 \um@setmathcode{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}%
340 \fi
341 \fi

```

Bold

```

342 \if@um@bfliteral
343 \um@setmathcode[26]{\um@usv@bfLatin}{\um@usv@bfLatin}%
344 \um@setmathcode[26]{\um@usv@bflatin}{\um@usv@bflatin}%
345 \um@setmathcode[26]{\um@usv@bfitLatin}{\um@usv@bfitLatin}%
346 \um@setmathcode[26]{\um@usv@bfitlatin}{\um@usv@bfitlatin}%
347 \um@setmathcode[25]{\um@usv@bfGreek}{\um@usv@bfGreek}%
348 \um@setmathcode[25]{\um@usv@bfgreek}{\um@usv@bfgreek}%
349 \um@setmathcode[25]{\um@usv@bfitGreek}{\um@usv@bfitGreek}%
350 \um@setmathcode[25]{\um@usv@bfitgreek}{\um@usv@bfitgreek}%

```

```

351 \else
352 \um@setmathcode[26]{\um@usv@bfLatin,\um@usv@bfitLatin}{%
353 \ifum@bfupLatin\um@usv@bfLatin\else\um@usv@bfitLatin\fi}%
354 \um@setmathcode[26]{\um@usv@bflatin,\um@usv@bfitlatin}{%
355 \ifum@bfuplatin\um@usv@bflatin\else\um@usv@bfitlatin\fi}%
356 \ifum@bfupgreek
357 \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfitGreek}{\um@usv@bfGreek}%
358 \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfvarTheta}%
359 \else
360 \um@setmathcode[25]{\um@usv@bfGreek,\um@usv@bfitGreek}{\um@usv@bfitGreek}%
361 \um@setmathcode{\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfitvarTheta}%
362 \fi
363 \ifum@bfupgreek
364 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfgreek}%
365 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfvarepsilon}%
366 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}%
367 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}%
368 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}%
369 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}%
370 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}%
371 \else
372 \um@setmathcode[25]{\um@usv@bfgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}%
373 \um@setmathcode{\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}%
374 \um@setmathcode{\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}%
375 \um@setmathcode{\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}%
376 \um@setmathcode{\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}%
377 \um@setmathcode{\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}%
378 \um@setmathcode{\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}%
379 \fi

```

Bold nabla and partial symbols:

```

380 \ifum@upNabla
381 \um@setmathcode{\um@usv@bfNabla}{\um@usv@bfNabla}%
382 \um@setmathcode{\um@usv@bfitNabla}{\um@usv@bfitNabla}%
383 \um@setmathcode{\um@usv@bfsfNabla}{\um@usv@bfsfNabla}%
384 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}%
385 \else
386 \um@setmathcode{\um@usv@bfNabla}{\um@usv@bfitNabla}%
387 \um@setmathcode{\um@usv@bfitNabla}{\um@usv@bfitNabla}%
388 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}%
389 \um@setmathcode{\um@usv@bfsfitNabla}{\um@usv@bfsfitNabla}%
390 \fi
391 \ifum@uppartial
392 \um@setmathcode{\um@usv@bfpartial}{\um@usv@bfpartial}%
393 \um@setmathcode{\um@usv@bfitpartial}{\um@usv@bfpartial}%
394 \um@setmathcode{\um@usv@bfsfpartial}{\um@usv@bfsfpartial}%
395 \um@setmathcode{\um@usv@bfsfitpartial}{\um@usv@bfsfitpartial}%

```



```

424 \um@mk@math{sf}
425 \um@mk@math{sfit}
426 \um@mk@math{tt}

```

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

```

427 \um@mk@math{bf}
428 \um@mk@math{bfup}
429 \um@mk@math{bfitt}
430 \um@mk@math{bfscr}
431 \um@mk@math{bffrak}
432 \um@mk@math{bfsf}
433 \um@mk@math{bfsfit}

```

`\mathcal`

```

434 \let\mathcal\mathscr

```

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

```

435 \newcommand\um@mathmap@noparse[3]{%
436   \@for\@ii:=#2\do{%
437     \expandafter\expandafter
438     \expandafter\um@addto@mathmap
439     \expandafter\expandafter
440     \expandafter{%
441       \expandafter\um@symfont
442       \expandafter}%
443     \expandafter{\@ii}{#1}{#3}%
444   }}%

```

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

```

445 \newcommand\um@mathmap@parse[3]{%
446   \@for\@ii:=\um@char@num@range\do{%
447     \ifnum\@ii=#3\relax
448     \@for\@jj:=#2\do{%
449       \expandafter\expandafter
450       \expandafter\um@addto@mathmap

```

```

451 \expandafter\expandafter
452 \expandafter{%
453 \expandafter\um@symfont
454 \expandafter}%
455 \expandafter{\@jj}{#1}{#3}}%
456 \fi}}%

```

`\um@addto@mathmap` #1 : Math symbol font, always/usually the expansion of `\um@symfont`
 #2 : Input slot, *e.g.*, the slot for ‘A’
 #3 : Maths alphabet, *e.g.*, `\mathbb`
 #4 : Output slot, *e.g.*, the slot for ‘A’
 This macro is used so that `\um@symfont` can be expanded before entering the `\g@addto@macro` command.

```

457 \newcommand\um@addto@mathmap[4]{%
458 \expandafter\g@addto@macro
459 \csname um\expandafter\@gobble\string#3\endcsname{%
460 \SetMathCode{#2}{\mathalpha}{#1}{#4}}}

```

6.5 (Big) operators























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

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

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

USV	Ex.	Macro	Description
U+02140	$\sum\limits_{0\atop 1}$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod\limits_{0\atop 1}$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod\limits_{0\atop 1}$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum\limits_{0\atop 1}$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int\limits_{0\atop 1}$	<code>\int</code>	INTEGRAL OPERATOR

U+0222C	$\int_0^1 \int_0^1$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\int_0^1 \int_0^1 \int_0^1$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	\oint_0^1	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oint_0^1 \oint_0^1$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oint_0^1 \oint_0^1 \oint_0^1$	<code>\oiiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\int_0^1 \curvearrowright$	<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232	$\int_0^1 \curvearrowleft$	<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\int_0^1 \curvearrowright$	<code>\ointctrclockwise</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	$\int_0^1 \boxplus$	<code>\leftouterjoin</code>	LEFT OUTER JOIN
U+027D6	$\int_0^1 \boxminus$	<code>\rightouterjoin</code>	RIGHT OUTER JOIN
U+027D7	$\int_0^1 \boxtimes$	<code>\fullouterjoin</code>	FULL OUTER JOIN
U+027D8	$\int_0^1 \Uparrow$	<code>\biguparrow</code>	LARGE UP TACK
U+027D9	$\int_0^1 \Downarrow$	<code>\bigdownarrow</code>	LARGE DOWN TACK
U+029F8	$\int_0^1 \bigodot$	<code>\xsol</code>	BIG SOLIDUS
U+029F9	$\int_0^1 \bigotimes$	<code>\xbsol</code>	BIG REVERSE SOLIDUS
U+02A00	\bigodot	<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR
U+02A01	\bigoplus	<code>\bigoplus</code>	N-ARY CIRCLED PLUS OPERATOR
U+02A02	\bigotimes	<code>\bigotimes</code>	N-ARY CIRCLED TIMES OPERATOR

U+02A03		\bigcupdot	N-ARY UNION OPERATOR WITH DOT
U+02A04		\bigcupplus	N-ARY UNION OPERATOR WITH PLUS
U+02A05		\bigsqcap	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06		\bigsqcup	N-ARY SQUARE UNION OPERATOR
U+02A07		\conjquant	TWO LOGICAL AND OPERATOR
U+02A08		\disjquant	TWO LOGICAL OR OPERATOR
U+02A09		\bigtimes	N-ARY TIMES OPERATOR
U+02A0B		\sumint	SUMMATION WITH INTEGRAL
U+02A0C		\iiint	QUADRUPLE INTEGRAL OPERATOR
U+02A0D		\intbar	FINITE PART INTEGRAL
U+02A0E		\intBar	INTEGRAL WITH DOUBLE STROKE
U+02A0F		\fint	INTEGRAL AVERAGE WITH SLASH
U+02A10		\cirfnint	CIRCULATION FUNCTION
U+02A11		\awint	ANTICLOCKWISE INTEGRATION
U+02A12		\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	\int	<code>\zpipe</code>	Z NOTATION SCHEMA PIPING
U+02A21	\int	<code>\zproject</code>	Z NOTATION SCHEMA PROJECTION
U+02AFC	\int	<code>\biginterleave</code>	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFF	\int	<code>\bigtalloblong</code>	N-ARY WHITE VERTICAL BAR

`\um@nolimits` This macro is a commalist containing those maths operators that require a `\no-limits` suffix. This list is used when processing `unicode-math.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 13). 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 \int , but that might be a matter of preference.

```

461 \def\um@nolimits{%
462   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oint\@elt\oiint
463   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrackwise\@elt\sumint
464   \@elt\intbar\@elt\intBar\@elt\oint\@elt\cirfnint\@elt\awint\@elt\rppoint
465   \@elt\scpoint\@elt\ncpoint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
466   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint}

```

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

```

467 \newcommand\addnolimits[1]{%
468   \expandafter\def
469   \expandafter\um@nolimits
470   \expandafter{\um@nolimits\@elt#1}}

```

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

```

471 \def\removenolimits#1{%
472   \begingroup
473     \def\@elt##1{%
474       \ifx##1#1\else
475         \noexpand\@elt\noexpand##1
476       \fi}
477     \xdef\um@nolimits{\um@nolimits}%
478   \endgroup}

```

$$\iiint_V$$

$$\iiint_V$$

$$\iiint_V$$

```
\setmathfont{Cambria Math} \[\iiint_V\]
\removenolimits\iiint
\setmathfont{Cambria Math} \[\iiint_V\]
\addnolimits\iiint
\setmathfont{Cambria Math} \[\iiint_V\]
```

6.6 Radicals

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

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

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

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

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

6.7 Delimiters

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

<http://www.latex-project.org/cgi-bin/ltbugs2html?pr=latex/3853&prlatex/3754>

```
480 \let\left@primitive\left
```

```
481 \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`

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











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

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


















```
\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\{z
\right\}^1\right\}\right)^2\right\}\right)^3\right\}\right)^4\right\}\right)^5 \]
```

Here are all `\mathopen` characters:

USV	Ex.	Macro	Description
U+00028	(<code>\lparen</code>	LEFT PARENTHESIS
U+0005B	[<code>\lbrack</code>	LEFT SQUARE BRACKET
U+0007B	{	<code>\lbrace</code>	LEFT CURLY BRACKET
U+000AB	«	<code>\guillemotleft</code>	(GUILLEMET), LEFT
U+02018	‘	<code>\lq</code>	SINGLE QUOTATION MARK, LEFT
U+0201A	,	<code>\quotsinglbase</code>	RISING SINGLE QUOTE, LEFT (LOW)
U+0201E	„	<code>\quotdblbase</code>	RISING DOUBLE QUOTE, LEFT (LOW)
U+02039	<	<code>\guilsinglleft</code>	(GUILLEMET), LEFT
U+0221A	√	<code>\sqrt</code>	RADICAL
U+02308	⌈	<code>\lceil</code>	LEFT CEILING
U+0230A	⌋	<code>\lfloor</code>	LEFT FLOOR
U+0231C	⤵	<code>\ulcorner</code>	UPPER LEFT CORNER
U+0231E	⤴	<code>\llcorner</code>	LOWER LEFT CORNER
U+02772	⌈	<code>\lbrbrak</code>	ORNAMENT
U+027C5	⌈	<code>\lbag</code>	LEFT S-SHAPED BAG DELIMITER
U+027E6	⌈	<code>\lBrack</code>	BRACKET
U+027E8	⌈	<code>\langle</code>	MATHEMATICAL LEFT ANGLE BRACKET
U+027EA	⌈	<code>\lAngle</code>	BRACKET
U+027EC	⌈	<code>\lbrbrak</code>	SHELL BRACKET
U+02983	⌈	<code>\lBrace</code>	LEFT WHITE CURLY BRACKET
U+02985	⌈	<code>\lParen</code>	LEFT WHITE PARENTHESIS
U+02987	⌈	<code>\llparenthesis</code>	Z NOTATION LEFT IMAGE BRACKET
U+02989	⌈	<code>\llangle</code>	Z NOTATION LEFT BINDING BRACKET
U+0298B	⌈	<code>\lbrackubar</code>	LEFT SQUARE BRACKET WITH UNDERBAR

U+0298D		<code>\lbrackultick</code>	LEFT SQUARE BRACKET WITH TICK IN TOP CORNER
U+0298F		<code>\lbracklltick</code>	LEFT SQUARE BRACKET WITH TICK IN BOTTOM CORNER
U+02991		<code>\langedot</code>	LEFT ANGLE BRACKET WITH DOT
U+02993		<code>\lparenless</code>	LEFT ARC LESS-THAN BRACKET
U+02997		<code>\lblkbrbrak</code>	LEFT BLACK TORTOISE SHELL BRACKET
U+029D8		<code>\lvzigzag</code>	LEFT WIGGLY FENCE
U+029DA		<code>\Lvzigzag</code>	LEFT DOUBLE WIGGLY FENCE
U+029FC		<code>\lcurvyangle</code>	LEFT POINTING CURVED ANGLE BRACKET
U+03014		<code>\lbrbrak</code>	LEFT BROKEN BRACKET
U+03018		<code>\Lbrbrak</code>	LEFT WHITE TORTOISE SHELL BRACKET

And `\mathclose`:












USV	Ex.	Macro	Description
U+00029)	<code>\rparen</code>	RIGHT PARENTHESIS
U+0005D]	<code>\rbrack</code>	RIGHT SQUARE BRACKET
U+0007D	}	<code>\rbrace</code>	RIGHT CURLY BRACKET
U+000BB	»	<code>\guillemotright</code>	DOUBLE ANGLE QUOTATION MARK (GUILLEMET), RIGHT
U+02019	'	<code>\rq</code>	SINGLE QUOTATION MARK, RIGHT
U+0201B		<code>\quotsinglright</code>	RISING SINGLE QUOTE, RIGHT (HIGH)
U+0201F		<code>\quotdblright</code>	RISING DOUBLE QUOTE, RIGHT (HIGH)
U+0203A	›	<code>\guilsinglright</code>	SINGLE ANGLE QUOTATION MARK (GUILLEMET), RIGHT
U+02309	⌈	<code>\rceil</code>	RIGHT CEILING
U+0230B	⌋	<code>\rfloor</code>	RIGHT FLOOR
U+0231D		<code>\urcorner</code>	UPPER RIGHT CORNER
U+0231F		<code>\lrcorner</code>	LOWER RIGHT CORNER
U+02773		<code>\rbrbrak</code>	LIGHT RIGHT TORTOISE SHELL BRACKET
U+027C6		<code>\rbag</code>	ORNAMENT
U+027E7		<code>\rBrack</code>	RIGHT S-SHAPED BAG DELIMITER
U+027E9		<code>\rangle</code>	MATHEMATICAL RIGHT WHITE SQUARE BRACKET
U+027EB		<code>\rAngle</code>	MATHEMATICAL RIGHT DOUBLE ANGLE BRACKET
U+027ED		<code>\Rbrbrak</code>	MATHEMATICAL RIGHT WHITE TORTOISE SHELL BRACKET
U+02984		<code>\Rbrace</code>	SHELL BRACKET
U+02986		<code>\Rbrack</code>	RIGHT WHITE CURLY BRACKET
U+02988		<code>\RParen</code>	RIGHT WHITE PARENTHESIS
U+0298A		<code>\rrparenthesis</code>	RIGHT WHITE TORTOISE SHELL BRACKET
U+0298C		<code>\rrangle</code>	RIGHT WHITE CURLY BRACKET
U+0298E		<code>\rbrackubar</code>	RIGHT WHITE PARENTHESIS
U+0298F		<code>\rbrackubar</code>	Z NOTATION RIGHT IMAGE BRACKET
U+02990		<code>\rbrackubar</code>	Z NOTATION RIGHT BINDING BRACKET
U+02991		<code>\rbrackubar</code>	RIGHT SQUARE BRACKET WITH UNDERBAR

U+0298E	⌞	\rbracklrtick	RIGHT SQUARE BRACKET WITH TICK IN BOTTOM CORNER
U+02990	⌟	\rbrackurtick	RIGHT SQUARE BRACKET WITH TICK IN TOP CORNER
U+02992	⌞	\rangledot	RIGHT ANGLE BRACKET WITH DOT
U+02994	⌞	\rparengtr	RIGHT ARC GREATER-THAN BRACKET
U+02998	⌞	\rblbrbrak	RIGHT BLACK TORTOISE SHELL BRACKET
U+029D9	⌞	\rvzigzag	RIGHT WIGGLY FENCE
U+029DB	⌞	\Rvzigzag	RIGHT DOUBLE WIGGLY FENCE
U+029FD	⌞	\rcurvyangle	RIGHT POINTING CURVED ANGLE BRACKET
U+03015	⌞	\rbrbrak	RIGHT BROKEN BRACKET
U+03019	⌞	\Rbrbrak	RIGHT WHITE TORTOISE SHELL BRACKET

6.8 Maths accents

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

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

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.

```

482 \newcommand\um@zf@feature[2]{%
483   \define@key[zf]{options}{#1}[]{}%
484   \if@um@fontspec@feature
485     #2
486   \else
487     \PackageError{fontspec/unicode-math}
488       {The ‘#1’ font feature can only be used for maths fonts}
489       {The feature you tried to use can only be in commands
490        like \protect\setmathfont}%
491   \fi}}

```

7.1 OpenType maths font features

```

492 \um@zf@feature{ScriptStyle}{%
493   \zf@update@ff{+ssty=0}}
494 \um@zf@feature{ScriptScriptStyle}{%
495   \zf@update@ff{+ssty=1}}

```

7.2 Script and scriptscript font options

```

496 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
497 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
498 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
499 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

7.3 Range processing

```

500 \define@choicekey+[um]{options}{Range}[\@tempa\@tempb]{ALL}{%
501   \ifcase\@tempb\relax
502     \@um@inittrue
503   \fi}{% else:
504     \xdef\um@char@range{\zap@space#1 \@empty}}

```

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

`\um@parse@term` #1 : unicode character slot
 #2 : control sequence (character macro)
 #3 : control sequence (math type)
 #4 : code to execute

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

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

```

505 \newcommand\um@parse@term[4]{%
506   \@for\@ii:=\um@char@range\do{%
507     \unless\ifx\@ii\@empty
508       \@tempswafalse

```

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

```

509     \expandafter\um@firstchar\expandafter{\@ii}%
510     \ifx\@tempa\um@backslash
511       \expandafter\ifx\@ii#2\relax
512         \@tempswatrue
513       \else
514         \expandafter\ifx\@ii#3\relax
515         \@tempswatrue
516       \fi
517     \fi

```

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


```

518     \else
519         \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
520     \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.

521     \if@tempswa
522         \ifx\um@char@num@range\empty
523             \g@addto@macro\um@char@num@range{#1}%
524         \else
525             \g@addto@macro\um@char@num@range{,#1}%
526         \fi
527         #4%
528     \fi
529 \fi}}
530 \def\um@firstof#1#2\@nil{#1}
531 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
532 \def\um@firstchar#1{\edef\tempa{\expandafter\um@firstof\string#1\@nil}}

```

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

```

533 \def\um@parse@range#1-#2-#3\@nil#4\@nil{%
534     \def\@tempa{#1}%
535     \def\@tempb{#2}%

```

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

```

536     \expandafter\ifx\expandafter\@marker\@tempb\relax
537         \ifnum#4=#1\relax
538             \@tempswatrue
539         \fi
540     \else

```

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

```

541 \ifx\@empty\@tempb
542 \ifnum#4>\numexpr#1-1\relax
543 \@tempwattrue
544 \fi
545 \else

```

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

```

546 \ifx\@empty\@tempa
547 \ifnum#4<\numexpr#2+1\relax
548 \@tempwattrue
549 \fi

```

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

```

550 \else
551 \ifnum#4>\numexpr#1-1\relax
552 \ifnum#4<\numexpr#2+1\relax
553 \@tempwattrue
554 \fi\fi\fi\fi\fi}

```

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

```

555 \newcommand\um@setmathcode[3][1]{%
556 \@for\um@inp:=#2\do{%
557 \um@FOR\@tempcnta=[1:#1]\do{%
558 \SetMathCode{\numexpr\um@inp+\@tempcnta-1\relax}
559 {\mathalpha}{\um@symfont}{\numexpr#3+\@tempcnta-1\relax}}}}

```

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

```

560 \newcommand\um@setmathalph[4][1]{%
561 \@for\um@inp:=#3\do{%
562 \um@FOR\@tempcnta=[1:#1]\do{%
563 \edef\@tempa{%
564 \noexpand\um@setsinglemathalph
565 {\noexpand#2}
566 {\number\numexpr \um@inp+\@tempcnta-1 \relax}
567 {\number\numexpr #4+\@tempcnta-1 \relax}}\@tempa}}}

```

BCDBCD *ABCDEF*

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

`\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.

```
568 \AtBeginDocument{\um@resolve@greek}%
569 \newcommand\um@resolve@greek{%
570   \def\Alpha{\itAlpha}%
571   \def\Beta{\itBeta}%
572   \def\Gamma{\itGamma}%
573   \def\Delta{\itDelta}%
574   \def\Epsilon{\itEpsilon}%
575   \def\Zeta{\itZeta}%
576   \def\Eta{\itEta}%
577   \def\Theta{\itTheta}%
578   \def\Iota{\itIota}%
579   \def\Kappa{\itKappa}%
580   \def\Lambda{\itLambda}%
581   \def\Mu{\itMu}%
582   \def\Nu{\itNu}%
583   \def\Xi{\itXi}%
584   \def\Omicron{\itOmicron}%
585   \def\Pi{\itPi}%
586   \def\Rho{\itRho}%
587   \def\varTheta{\itvarTheta}%
588   \def\Sigma{\itSigma}%
589   \def\Tau{\itTau}%
590   \def\Upsilon{\itUpsilon}%
591   \def\Phi{\itPhi}%
592   \def\Chi{\itChi}%
593   \def\Psi{\itPsi}%
594   \def\Omega{\itOmega}%
```

Lowercase:

```
595 \def\alpha{\italpha}%
596 \def\beta{\itbeta}%
597 \def\gamma{\itgamma}%
598 \def\delta{\itdelta}%
599 \def\varepsilon{\itvarepsilon}%
600 \def\zeta{\itzeta}%
601 \def\eta{\iteta}%
602 \def\theta{\ittheta}%
```

```

603 \def\iota{\itiota}%
604 \def\kappa{\itkappa}%
605 \def\lambda{\itlambda}%
606 \def\mu{\itmu}%
607 \def\nu{\itnu}%
608 \def\xi{\itxi}%
609 \def\omicron{\itomicron}%
610 \def\pi{\itpi}%
611 \def\rho{\itrho}%
612 \def\varsigma{\itvarsigma}%
613 \def\sigma{\itsigma}%
614 \def\tau{\ittau}%
615 \def\upsilon{\itupsilon}%
616 \def\phi{\itphi}%
617 \def\chi{\itchi}%
618 \def\psi{\itpsi}%
619 \def\omega{\itomega}%
620 \def\varepsilon{\itvarepsilon}%
621 \def\vartheta{\itvartheta}%
622 \def\varkappa{\itvarkappa}%
623 \def\varphi{\itvarphi}%
624 \def\varrho{\itvarrho}%
625 \def\varrho{\itvarsigma}%
626 \def\varpi{\itvarpi}}

```

\um@def@numbers

```

627 \newcommand\um@def@numbers{%
628   \um@setmathcode[10]{\um@usv@num}{\um@usv@num}}

```

\um@def@upLatin

```

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

```

\um@def@itLatin

```

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

```

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

```

633 \newcommand\um@def@itlatin{%
634   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}%
635   \um@setmathcode{\`h}{\um@usv@ith}%
636   \um@setmathcode{\um@usv@ith}{\um@usv@ith}%
637 }

```

\um@def@uplatin

```

638 \newcommand\um@def@uplatin{%
639   \um@setmathcode[26]{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}%
640   \um@setmathcode{\um@usv@ith}{\`h}%
641 }

```

\um@def@upGreek

```

642 \newcommand\um@def@upGreek{%
643   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}%
644   \um@setmathcode{\um@usv@varTheta}{\um@usv@varTheta}%
645   \um@setmathcode{"1D6F3}{\um@usv@varTheta}%
646 }

```

\um@def@itGreek

```

647 \newcommand\um@def@itGreek{%
648   \um@setmathcode[25]{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}%
649   \um@setmathcode{\um@usv@varTheta}{\um@usv@itvarTheta}%
650 }

```

\um@def@upgreek

```

651 \newcommand\um@def@upgreek{%
652   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}%
653   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
654   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
655   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
656   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
657   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
658   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
659 }

```

\um@def@itgreek

```

660 \newcommand\um@def@itgreek{%
661   \um@setmathcode[25]{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}%
662   \um@setmathcode{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}%
663   \um@setmathcode{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}%
664   \um@setmathcode{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}%
665   \um@setmathcode{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}%
666   \um@setmathcode{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}%
667   \um@setmathcode{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}%
668 }

```

File II

Maths alphabets mapping definitions

```

1 \newcommand\um@setup@alphabets{%

```

: TODO : nested alphabets?

7.3.1 Upright: `\mathup`

Can't call it `\mathrm` any more because it contains Greek as well!

ABCDEFGHIJKLMNOPQRSTUVWXYZ	<code>\mathup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
abcdefghijklmnopqrstuvwxyz	<code>\mathup{abcdefghijklmnopqrstuvwxyz}\$ \</code>
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ	<code>\mathup{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}\$\quad\mathup{ }\$ \</code>
αβγδεζηθικλμνξοπρστυφχψω εθκφρω	<code>\mathup{αβγδεζηθικλμνξοπρστυφχψω}\$\quad\mathup{εθκφρω}\$ \</code>

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

```
2 \um@setmathalph[26]{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}%
3 \um@setmathalph[26]{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}%
4 \um@setmathalph[25]{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}%
5 \um@setmathalph[25]{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}%
6 \um@setmathalph{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
7 \um@setmathalph{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
8 \um@setmathalph{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}%
9 \um@setmathalph{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
10 \um@setmathalph{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
11 \um@setmathalph{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
12 \um@setmathalph{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
13 \um@setmathalph{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
14 \um@setmathalph{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
15 \um@setmathalph[26]{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}%
16 \um@setmathalph[26]{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}%
17 \um@setmathalph[25]{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}%
18 \um@setmathalph[25]{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}%
19 \um@setmathalph{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
20 \um@setmathalph{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
21 \um@setmathalph{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}%
22 \um@setmathalph{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
23 \um@setmathalph{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
24 \um@setmathalph{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
25 \um@setmathalph{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
26 \um@setmathalph{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
27 \um@setmathalph{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
```

7.3.2 Italic: `\mathit`

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

Roman:

```

28 \um@setmathalph[26]{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}%
29 \um@setmathalph[26]{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}%
30 \um@setmathalph{\mathit}{\`h}{\um@usv@ith}%
31 \um@setmathalph{\mathit}{\um@usv@ith}{\um@usv@ith}%

```

Greek:

```

32 \um@setmathalph[25]{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}%
33 \um@setmathalph[25]{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}%
34 \um@setmathalph{\mathit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}%
35 \um@setmathalph{\mathit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}%
36 \um@setmathalph{\mathit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarTheta}%
37 \um@setmathalph{\mathit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}%
38 \um@setmathalph{\mathit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}%
39 \um@setmathalph{\mathit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}%
40 \um@setmathalph{\mathit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}%
41 \um@setmathalph{\mathit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}%
42 \um@setmathalph{\mathit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}%

```

7.3.3 Blackboard or double-struck: \mathbb

0123456789	<code>\$\mathbb{0123456789}\$ \</code>
ABCDEFGHIJKLMN O PQRSTU VWXYZ	<code>\$\mathbb{ABCDEFGHIJKLMN O PQRSTU VWXYZ}\$ \</code>
abcdefghijklmnopqrstuvwxyz	<code>\$\mathbb{abcdefghijklmnopqrstuvwxyz}\$ \</code>

Numbers:

```

43 \um@setmathalph[10]{\mathbb}{\um@usv@num}{\um@usv@bbnum}%

```

Letters:

```

44 \um@setmathalph[26]{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bbLatin}%
45 \um@setmathalph{\mathbb}{\`C,"1D60A"}{"2102}%
46 \um@setmathalph{\mathbb}{\`H,"1D60F"}{"210D}%
47 \um@setmathalph{\mathbb}{\`N,"1D60F"}{"2115}%
48 \um@setmathalph{\mathbb}{\`P,"1D617"}{"2119}%
49 \um@setmathalph{\mathbb}{\`Q,"1D618"}{"211A}%
50 \um@setmathalph{\mathbb}{\`R,"1D619"}{"211D}%
51 \um@setmathalph{\mathbb}{\`Z,"1D621"}{"2124}%

```

Roman lowercase:

```

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

```

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

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

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

```

53 \um@setmathalph[26]{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}%
54 \um@setmathalph{\mathscr}{`\B,"1D435}{`"212C}%
55 \um@setmathalph{\mathscr}{`\E,"1D438}{`"2130}%
56 \um@setmathalph{\mathscr}{`\F,"1D439}{`"2131}%
57 \um@setmathalph{\mathscr}{`\H,"1D43B}{`"210B}%
58 \um@setmathalph{\mathscr}{`\I,"1D43C}{`"2110}%
59 \um@setmathalph{\mathscr}{`\L,"1D43F}{`"2112}%
60 \um@setmathalph{\mathscr}{`\M,"1D440}{`"2133}%
61 \um@setmathalph{\mathscr}{`\R,"1D445}{`"211B}%
62 \um@setmathalph[26]{\mathscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@scrlatin}%
63 \um@setmathalph{\mathscr}{`\e,"1D452}{`"212F}%
64 \um@setmathalph{\mathscr}{`\g,"1D454}{`"210A}%
65 \um@setmathalph{\mathscr}{`\o,"1D45C}{`"2134}%

```

7.3.5 Fraktur or fraktur or blackletter: `\mathfrak`

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

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

```

66 \um@setmathalph[26]{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}%
67 \um@setmathalph{\mathfrak}{`\C,"1D436}{`"212D}%
68 \um@setmathalph{\mathfrak}{`\H,"1D43B}{`"210C}%
69 \um@setmathalph{\mathfrak}{`\I,"1D43C}{`"2111}%
70 \um@setmathalph{\mathfrak}{`\R,"1D445}{`"211C}%
71 \um@setmathalph{\mathfrak}{`\Z,"1D44D}{`"2128}%
72 \um@setmathalph[26]{\mathfrak}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@fraklatin}%

```

7.3.6 Sans serif: `\mathsf`

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

```

73 \um@setmathalph[10]{\mathsf}{\um@usv@num}{\um@usv@sfnun}%

```



```

74 \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sflatin}%
75 \um@setmathalph[26]{\mathsf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sflatin}%

```

7.3.7 Sans serif italic: \mathsf

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

```

76 \um@setmathalph[10]{\mathsf}{\um@usv@num}{\um@usv@sfnun}%
77 \um@setmathalph[26]{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLatin}%
78 \um@setmathalph[26]{\mathsf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfitlatin}%

```

7.3.8 Typewriter or monospaced: \mathtt

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

```

79 \um@setmathalph[10]{\mathtt}{\um@usv@num}{\um@usv@ttnum}%
80 \um@setmathalph[26]{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}%
81 \um@setmathalph[26]{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}%

```

7.4 Bold alphabets' character mappings

7.4.1 Bold: \mathbf

0123456789	$\mathbf{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathbf{abcdefghijklmnopqrstuvwxyz}$
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	$\mathbf{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$
θ	$\mathbf{\theta}$
αβγδεζηθικλμνξοπρστυφχψω	$\mathbf{\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\rho\sigma\tau\upsilon\phi\chi\psi\omega}$
€θκφρϖ	$\mathbf{\epsilon\theta\kappa\phi\rho\varpi}$

```

82 \um@setmathalph[10]{\mathbf}{\um@usv@num}{\um@usv@bfnum}%
83 \um@setmathalph{\mathbf}{\um@usv@Digamma}{\um@usv@1D7CA}%
84 \um@setmathalph{\mathbf}{\um@usv@digamma}{\um@usv@1D7CB}%
85 \if@um@bfliteral
86 \um@setmathalph[26]{\mathbf}{\um@usv@upLatin}{\um@usv@bfLatin}%

```

```

87 \um@setmathalph[26]{\mathbf}{\um@usv@itLatin}{\um@usv@bfitLatin}%
88 \um@setmathalph[26]{\mathbf}{\um@usv@uplatin}{\um@usv@bflatin}%
89 \um@setmathalph[26]{\mathbf}{\um@usv@itlatin}{\um@usv@bfitlatin}%
90 \um@setmathalph[25]{\mathbf}{\um@usv@upGreek}{\um@usv@bfGreek}%
91 \um@setmathalph[25]{\mathbf}{\um@usv@itGreek}{\um@usv@bfitGreek}%
92 \um@setmathalph[25]{\mathbf}{\um@usv@upgreek}{\um@usv@bfgreek}%
93 \um@setmathalph[25]{\mathbf}{\um@usv@itgreek}{\um@usv@bfitgreek}%
94 \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfith}%
95 \um@setmathalph{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}%
96 \um@setmathalph{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}%
97 \um@setmathalph{\mathbf}{\um@usv@Digamma}{\um@usv@bfDigamma}%
98 \um@setmathalph{\mathbf}{\um@usv@partial}{\um@usv@bfpartial}%
99 \um@setmathalph{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}%
100 \um@setmathalph{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}%
101 \um@setmathalph{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}%
102 \um@setmathalph{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}%
103 \um@setmathalph{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}%
104 \um@setmathalph{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}%
105 \um@setmathalph{\mathbf}{\um@usv@digamma}{\um@usv@bfdigamma}%
106 \um@setmathalph{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}%
107 \um@setmathalph{\mathbf}{\um@usv@itNabla}{\um@usv@bfitNabla}%
108 \um@setmathalph{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}%
109 \um@setmathalph{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}%
110 \um@setmathalph{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}%
111 \um@setmathalph{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}%
112 \um@setmathalph{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}%
113 \um@setmathalph{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}%
114 \um@setmathalph{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}%
115 \else
116 \ifum@bfupLatin
117 \um@setmathalph[26]{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bflatin}%
118 \else
119 \um@setmathalph[26]{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}%
120 \fi
121 \ifum@bfuplatin
122 \um@setmathalph[26]{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bflatin}%
123 \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfuph}%
124 \else
125 \um@setmathalph[26]{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}%
126 \um@setmathalph{\mathbf}{\um@usv@ith}{\um@usv@bfith}%
127 \fi
128 \ifum@bfupGreek
129 \um@setmathalph[25]{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfGreek}%
130 \um@setmathalph{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvarTheta}%
131 \else
132 \um@setmathalph[25]{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}%

```


7.4.3 Bold Italic: `\mathbf{fup}`

```

175 \um@setmathalph[10]{\mathbfup}{\um@usv@num}{\um@usv@bnum}%
176 \um@setmathalph[26]{\mathbfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bLatin}%
177 \um@setmathalph[26]{\mathbfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bLatin}%
178 \um@setmathalph[25]{\mathbfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bGreek}%
179 \um@setmathalph[25]{\mathbfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bGreek}%
180 \um@setmathalph[26]{\mathbfup}{\um@usv@bLatin}{\um@usv@bLatin}%
181 \um@setmathalph[26]{\mathbfup}{\um@usv@bLatin}{\um@usv@bLatin}%
182 \um@setmathalph[25]{\mathbfup}{\um@usv@bGreek}{\um@usv@bGreek}%
183 \um@setmathalph[25]{\mathbfup}{\um@usv@bGreek}{\um@usv@bGreek}%
184 \um@setmathalph{\mathbfup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bvarTheta}%
185 \um@setmathalph{\mathbfup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bNabla}%
186 \um@setmathalph{\mathbfup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bpartial}%
187 \um@setmathalph{\mathbfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bvarepsilon}%
188 \um@setmathalph{\mathbfup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bvartheta}%
189 \um@setmathalph{\mathbfup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bvarkappa}%
190 \um@setmathalph{\mathbfup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bvarphi}%
191 \um@setmathalph{\mathbfup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bvarrho}%
192 \um@setmathalph{\mathbfup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bvarpi}%

```

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

193 \um@setmathalph{10}{\mathbbfrak}{\um@usv@num}{\um@usv@bfnum}%
194 \um@setmathalph{26}{\mathbbfrak}{\um@usv@upLatin, \um@usv@itLatin, \um@usv@frakLatin}{\um@usv@bffr

195 \um@setmathalph[26]{\mathbfrac}{\um@usv@uplatin,\um@usv@itlatin,\um@usv@fraklatin}{\um@usv@bfrac

*ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz*

```

196 \um@setmathalph{10}{\mathbfscr}{\um@usv@num}{\um@usv@bfnun}%
197 \um@setmathalph{26}{\mathbfscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfscrLatin}%
198 \um@setmathalph{26}{\mathbfscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfscrLatin}%

```

[illegible]

```

199 \um@setmathalph[10]{\mathbfsf}{\um@usv@num}{\um@usv@bfnun}%
200 \um@setmathalph[26]{\mathbfsf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfLatin}%
201 \um@setmathalph[26]{\mathbfsf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfLatin}%
202 \um@setmathalph[25]{\mathbfsf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfGreek}%
203 \um@setmathalph[25]{\mathbfsf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfGreek}%

```

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

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

206 \um@setmathalph\mathbfsf\um@usv@partial}{"1D789}%
207 \um@setmathalph\mathbfsf\um@usv@varepsilon}{"1D78A}%
208 \um@setmathalph\mathbfsf\um@usv@vartheta}{"1D78B}%
209 \um@setmathalph\mathbfsf\um@usv@varkappa}{"1D78C}%
210 \um@setmathalph\mathbfsf\um@usv@varphi}{"1D78D}%
211 \um@setmathalph\mathbfsf\um@usv@varrho}{"1D78E}%
212 \um@setmathalph\mathbfsf\um@usv@varpi}{"1D78F}%

7.4.7 Bold italic sans serif: `\mathbfsfit`

[illegible]

```

213 \um@setmathalph[10]{\mathbfsfit}{\um@usv@num}{\um@usv@bfnun}%
214 \um@setmathalph[26]{\mathbfsfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfitLatin}%
215 \um@setmathalph[26]{\mathbfsfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfitLatin}%
216 \um@setmathalph[25]{\mathbfsfit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfitGreek}%
217 \um@setmathalph[25]{\mathbfsfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfitgreek}%

```

Other symbols:

218 \um@setmathalph{\mathbfsfit}{\um@usv@varTheta}{"1D7A1}%
219 \um@setmathalph{\mathbfsfit}{\um@usv@Nabla}{"1D7A9}%
220 \um@setmathalph{\mathbfsfit}{\um@usv@partial}{"1D7C3}%
221 \um@setmathalph{\mathbfsfit}{\um@usv@varepsilon}{"1D7C4}%
222 \um@setmathalph{\mathbfsfit}{\um@usv@vartheta}{"1D7C5}%
223 \um@setmathalph{\mathbfsfit}{\um@usv@varkappa}{"1D7C6}%
224 \um@setmathalph{\mathbfsfit}{\um@usv@varphi}{"1D7C7}%
225 \um@setmathalph{\mathbfsfit}{\um@usv@varrho}{"1D7C8}%
226 \um@setmathalph{\mathbfsfit}{\um@usv@varpi}{"1D7C9}%

That it.

227 }

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

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

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

```

228 \begingroup
229   \catcode`\^=12\relax
230   \gdef\um@scancharlet#1="#2\@nil{%
231     \lowercase{\scantokens{\global\let#1=^^^#2}}}
232   \catcode`\^=12\relax
233   \gdef\um@scanactivedef"#1\@nil#2{%
234     \lowercase{\scantokens{\global\def^^^^#1#2}}}%

```

```

235 \endgroup
236 \let\unicodemathgobble\@gobble

    Now give \UnicodeMathSymbol a definition in terms of \um@scancharlet and
    we're good to go.
237 \begingroup
238   \def\UnicodeMathSymbol#1#2#3#4{%
239     \um@scancharlet#2=#1\@nil}
240   \input unicode-math.tex\relax
241   \input unicode-math-add.tex\relax
242 \endgroup

    We need to change LATEX's idea of the font used to typeset things like \sin
    and \cos:
243 \def\operator@font{\um@mathup}
244 \AtBeginDocument{\setmathfont[Range=ALL]{Cambria Math}}

```

File III

stix table data extraction

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

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

```

1 #!/bin/sh
2
3 cat stix-tbl.asc |
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):

Print the actual entry corresponding to the unicode character:

Now replace the STIX class abbreviations with their T_EX macro names.

```

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

```

A.1 Overview

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

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

48

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

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

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

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

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

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

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

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

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

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

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

Accents are not included yet.

Summary For symbols, something like:

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

For characters, something like:

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

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

File IV

Some manner of unit testing

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

B The regular weight alphabets

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

```

1 <*test>
2 \documentclass{article}
3 \usepackage[a6paper]{geometry}
4 \usepackage{fontspec}
5 \setmainfont{FPL Neu}
6 \usepackage{unicode-math}
7 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
8 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
9 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
10 \def\upgreek{αβγδεζηθικλμνξοπρστυφχψω}
11 \def\itLatin{abcdefghijklmnopqrstuvwxyz}
12 \def\itLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
13 \def\itGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}
14 \def\itgreek{αβγδεζηθικλμνξοπρστυφχψω}
15 \def\testmath#1{%
16     \makebox[\linewidth][l]{%
17         \makebox[0pt][l]{${\csname up#1\endcsname}$}%
18         \makebox[0pt][l]{${\csname it#1\endcsname}$}%
19     }
20 \begin{document}
21 \setmathfont[Colour=2255FF99]{Cambria Math}
22 \parindent=0pt
23 \voffset=-1in
24 \hoffset=-1in
25 \setbox0=\vbox{%
26     \testmath{Latin}\\
27     \testmath{Latin}\\
28     \testmath{Greek}\\
29     \testmath{greek}}

```

```

28 \dimen0=\ht0
29 \advance\dimen0\dp0
30 \edef\papersize{papersize=\the\wd0,\the\dimen0}
31 \setbox255=\vbox{\special{\papersize}\box0}
32 \shipout\box255
33 \end{document}

```

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

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

```

34 \documentclass{article}
35 \usepackage[a6paper]{geometry}
36 \usepackage{fontspec}
37 \setmainfont{FPL Neu}
38 \usepackage{unicode-math}
39 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
40 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
41 \def\upGreek{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΨΤΥΦΧΨΩ}
42 \def\upgreek{0000000000000000π0000000000000000}
43 \def\itLatin{00000000000000000000000000000000}
44 \def\itlatin{00000000000000000000000000000000}
45 \def\itGreek{00000000000000000000000000000000}
46 \def\itgreek{00000000000000000000000000000000}
47 \def\bfulatin{00000000000000000000000000000000}
48 \def\bfulatin{00000000000000000000000000000000}
49 \def\bfulgreek{00000000000000000000000000000000}
50 \def\bfulgreek{00000000000000000000000000000000}
51 \def\bfitLatin{00000000000000000000000000000000}
52 \def\bfitlatin{00000000000000000000000000000000}
53 \def\bfitGreek{00000000000000000000000000000000}
54 \def\bfitgreek{00000000000000000000000000000000}
55 \providecommand\mathalphabet{\mathbf}
56 \def\testmath#1{%
57   \makebox[\linewidth][l]{%
58     \makebox[0pt][l]{\mathalphabet{\csname up#1\endcsname}}}%
59     \makebox[0pt][l]{\mathalphabet{\csname it#1\endcsname}}}%
60     \makebox[0pt][l]{\csname bful#1\endcsname}}%
61     \makebox[0pt][l]{\csname bfit#1\endcsname}}%

```

```

62     }}
63 \begin{document}
64 \setmathfont[Colour=2255FF55]{Cambria Math}
65 \parindent=0pt
66 \voffset=-1in
67 \hoffset=-1in
68 \setbox0=\vbox{%
69 \testmath{Latin}\\
70 \testmath{latin}\\
71 \testmath{Greek}\\
72 \testmath{greek}}
73 \dimen0=\ht0
74 \advance\dimen0\dp0
75 \edef\papersize{papersize=\the\wd0,\the\dimen0}
76 \setbox255=\vbox{\special{\papersize}\box0}
77 \shipout\box255
78 \end{document}
</testbf>

```

Change History

v0.01	
General: Tidied up awk code	48
<code>\addnolimits</code> : Implemented for <code>\nolimits</code> processing	26
<code>\um@nolimits</code> : Implemented for <code>\nolimits</code> processing	26
<code>\um@radicals</code> : Implemented for more general radicals processing.	27
v0.1	
General: Ignore <code>\text..</code> STIX commands.	47
v0.2	
<code>\removenolimits</code> : Implemented for <code>\nolimits</code> processing	26
<code>\um@setmathalph</code> : Beginning to sort out the input encoding.	35
<code>\um@setmathcode</code> : Beginning to sort out the input encoding.	34
v0.3	
General: Ignore <code>\ipa..</code> and <code>\tone..</code> accents.	47
<code>\um@mathsymbol</code> : Guttled and simplified. TODO: re-add robust checking.	13
<code>\um@set@mathsymbol</code> : Added <code>\mathaccentsupport</code> .	15

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
\@DeclareMathDelimiter	79, <u>98</u>
\@cclvi	199
\@eha	61, 71, 139, 144, 147, 179, 182, 221, 245, 285
\@elt	462–466, 470, 473, 475
\@empty	245, 246, 269, 284, 289–305, 310, 504, 507, 522, 541, 546
\@firstofone	90
\@firstoftwo	94
\@font@info	52, 129, 204, 270
\@font@warning	264
\@for	436, 446, 448, 506, 556, 561
\@gobble	55, 132, 211, 213, 214, 231, 236, 275, 276, 459
\@gobblefour	266, 268, 272, 274, 284, 288
\@ifundefined	200
\@ii	436, 443, 446, 447, 506, 507, 509, 511, 514, 519
\@jj	448, 455
\@latex@error	61, 71, 139, 144, 147, 179, 182, 221, 245, 284, 288
\@marker	519, 536
\@ne	26, 231
\@nil	208, 230, 233, 239, 254, 262, 519, 530–533
\@onlypreamble	84, 97, 150, 185, 189, 192
\@preamblecmds	198
\@spaces	275
\@tempa	128, 165, 184, 190, 258, 267, 500, 510, 532, 534, 546, 563, 567
\@tempb	128, 129, 165, 166, 184, 185, 190, 191, 500, 501, 535, 536, 541
\@tempcnta	557–559, 562, 566, 567
\@tempswafalse	194, 235, 508
\@tempswatrue	197, 238, 512, 515, 538, 543, 548, 553
\@um@bfliteraltrue	182
\@um@bfupGreekfalse	134, 167
\@um@bfupGreektrue	145, 156, 172, 177
\@um@bfupLatinfalse	136, 169
\@um@bfupLatintrue	147, 158, 174, 179
\@um@bfupgreekfalse	135, 146, 168, 173
\@um@bfupgreektrue	157, 178
\@um@bfuplatinfalse	137, 170
\@um@bfuplatintrue	148, 159, 175, 180
\@um@fontspec@featuretrue	248
\@um@initfalse	247
\@um@inittrue	502
\@um@literaltrue	163
\@um@upGreekfalse	130
\@um@upGreektrue	141, 152
\@um@upLatinfalse	132, 143
\@um@upLatintrue	154
\@um@upNablafalse	138, 188
\@um@upNablatrie	149, 160, 186
\@um@upgreekfalse	131, 142
\@um@upgreektrue	153
\@um@uplatinfalse	133, 144, 155
\@um@uppartialfalse	139, 194
\@um@uppartialtrue	150, 161, 192
\@xDeclareMathDelimiter	96, <u>151</u>
\@xxDeclareMathDelimiter	81, <u>85</u>
\\	10–13, 17, 23–33, 69–71
\^	24, 229, 232
Numbers	
\0	34
\	17–20, 23–32
A	
\A	35
\a	36
\active	206
\addnolimits	<u>467</u>
\addto@hook	257, 261, 278

<code>\advance</code>		<code>\DeclareMathSymbol</code>	34, 93
26, 29, 44, 74, 111, 119, 164, 172, 230		<code>\DeclareMathVersion</code>	251
<code>\alloc@</code>	199	<code>\DeclareSymbolFont</code>	193, 278
<code>\Alpha</code>	570	<code>\def</code>	6–14, 20–23, 34–126,
<code>\alpha</code>	595	151, 186, 190, 193, 196, 198, 199,	
<code>\AtBeginDocument</code>	244, 568	201, 203, 205, 213, 223, 227, 234,	
<code>\awint</code>	464	237, 238, 243, 248, 250, 252–256,	
		259, 270, 414, 461, 468, 471,	
		473, 479, 481, 530, 532–535, 570–626	
B		<code>\def@cn</code>	19
<code>\B</code>	54	<code>\define@choicekey</code> 127, 165, 184, 190, 500	
<code>\begin</code>	18, 63	<code>\define@cmdkey</code>	496–499
<code>\begingroup</code>	38, 86,	<code>\define@key</code>	483
105, 158, 205, 228, 237, 253, 415, 472		<code>\delcode</code>	191
<code>\Beta</code>	571	<code>\delimiter</code>	122, 187
<code>\beta</code>	596	<code>\Delta</code>	573
<code>\bfitGreek</code>	53	<code>\delta</code>	598
<code>\bfitgreek</code>	54	<code>\dimen</code>	28–30, 73–75
<code>\bfitLatin</code>	51	<code>\divide</code>	41, 108, 116, 161, 169
<code>\bfitlatin</code>	52	<code>\do</code> 23, 436, 446, 448, 506, 556, 557, 561, 562	
<code>\bfupGreek</code>	49	<code>\documentclass</code>	1, 34
<code>\bfupgreek</code>	50	<code>\dp</code>	29, 74
<code>\bfupLatin</code>	47		
<code>\bfuplatin</code>	48	E	
<code>\box</code>	31, 32, 76, 77	<code>\E</code>	55
		<code>\e</code>	63
C		<code>\edef</code>	30,
<code>\C</code>	45, 67	46, 75, 112, 120, 122, 165, 173, 195,	
<code>\catcode</code>	206, 229, 232	225, 228, 236, 258, 275, 531, 532, 563	
<code>\cdp@elt</code>	196, 237	<code>\else</code>	53, 60, 64, 70, 80, 91, 115,
<code>\cdp@list</code>	198, 239	118, 124, 130, 131, 138, 143, 146,	
<code>\chardef</code>	199	178, 181, 209, 217, 220, 221, 226,	
<code>\Chi</code>	592	231, 234, 244, 263, 273, 277, 283,	
<code>\chi</code>	617	287, 307, 326–330, 333, 338, 351,	
<code>\cirfnint</code>	464	353, 355, 359, 371, 385, 396, 403,	
<code>\count</code> . 39–42, 44, 51, 59, 67, 106–109,		474, 486, 513, 518, 524, 540, 545, 550	
111, 112, 114–117, 119, 120,		<code>\encodingdefault</code>	279
159–162, 164, 165, 167–170, 172, 173		<code>\end</code>	33, 78
<code>\count@</code>	42–44,	<code>\endcsname</code> 16, 17, 20, 35, 50, 55,	
109–111, 117–119, 162–164, 170–172		58–61, 66, 99, 102, 126, 127, 132,	
<code>\csname</code>	16, 17, 20, 35,	135, 136, 152, 155, 175, 201, 202,	
50, 55, 58–61, 66, 99, 102, 126, 127,		207, 208, 211, 213, 214, 216, 226,	
132, 135, 136, 152, 155, 175, 201,		229, 231, 242, 243, 249, 414, 416, 459	
202, 207, 208, 211, 213–216, 226,		<code>\endgroup</code>	69, 89, 92,
229, 230, 242, 243, 249, 414, 416, 459		142, 177, 209, 235, 242, 282, 418, 478	
D		<code>\Epsilon</code>	574
<code>\DeclareMathDelimiter</code>	77		

\backslash iota	603	\backslash itUpsilon	590
\backslash itAlpha	570	\backslash itupsilon	615
\backslash italpha	595	\backslash itvarepsilon	599, 620
\backslash itBeta	571	\backslash itvarkappa	622
\backslash itbeta	596	\backslash itvarphi	623
\backslash itChi	592	\backslash itvarpi	626
\backslash itchi	617	\backslash itvarrho	624
\backslash itDelta	573	\backslash itvarsigma	612, 625
\backslash itdelta	598	\backslash itvarTheta	587
\backslash itEpsilon	574	\backslash itvartheta	621
\backslash itEta	576	\backslash itXi	583
\backslash iteta	601	\backslash itxi	608
\backslash itGamma	572	\backslash itZeta	575
\backslash itgamma	597	\backslash itzeta	600
\backslash itGreek	12, 45		
\backslash itgreek	13, 46	K	
\backslash itIota	578	\backslash Kappa	579
\backslash itiota	603	\backslash kappa	604
\backslash itKappa	579		
\backslash itkappa	604	L	
\backslash itLambda	580	\backslash L	59
\backslash itlambda	605	\backslash Lambda	580
\backslash itLatin	10, 43	\backslash lambda	605
\backslash itlatin	11, 44	\backslash left	480
\backslash itMu	581	\backslash left@primitive	480, 481
\backslash itmu	606	\backslash let	87, 200, 231, 236, 244–246, 272, 276, 290–306, 308, 434, 480
\backslash itNu	582	\backslash linewidth	15, 57
\backslash itnu	607	\backslash long	23
\backslash itOmega	594	\backslash lowercase	231, 234
\backslash itomega	619	\backslash lowint	466
\backslash itOmicron	584		
\backslash itomicron	609	M	
\backslash itPhi	591	\backslash M	60
\backslash itphi	616	\backslash makebox	15–17, 57–61
\backslash itPi	585	\backslash mathaccent	232
\backslash itpi	610	\backslash mathalpha	87, 460, 559
\backslash itPsi	593	\backslash mathalphabet	55, 58, 59
\backslash itpsi	618	\backslash mathbb	43–52
\backslash itRho	586	\backslash mathbf	55, 82–84, 86–114, 117, 119, 122, 123, 125, 126, 129, 130, 132, 133, 136–142, 144–150, 153–156
\backslash itrho	611	\backslash mathbffrak	193–195
\backslash itSigma	588	\backslash mathbfit	157–174
\backslash itsigma	613	\backslash mathbfscr	196–198
\backslash itTau	589	\backslash mathbfsf	199–212
\backslash ittau	614	\backslash mathbfsfrit	213–226
\backslash itTheta	577		
\backslash ittheta	602		

\backslash mathbfup 175–192
 \backslash mathcal 434
 \backslash mathchar 46
 \backslash mathchar@type 74, 76, 88, 187,
 212, 223, 225, 228, 230, 233, 235, 242
 \backslash mathchardef 74
 \backslash mathclose 227
 \backslash mathcode 76, 207
 \backslash mathfrak 66–72
 \backslash mathgroup 199
 \backslash mathit 28–42
 \backslash mathop 204
 \backslash mathopen 222, 481
 \backslash mathord 87
 \backslash mathscr 53–65, 434
 \backslash mathsf 73–75
 \backslash mathsfitt 76–78
 \backslash mathtt 79–81
 \backslash mathup 2–27
 \backslash mddefault 279
 \backslash meaning 46, 122
 \backslash MessageBreak 265, 274
 \backslash Mu 581
 \backslash mu 606
 \backslash multiply 43, 110, 118, 163, 171

N

\backslash N 47
 \backslash new@mathgroup 199–201
 \backslash new@symbolfont 202, 223
 \backslash newcommand 1, 31–33, 241,
 243, 407, 409, 413, 435, 445, 457,
 467, 482, 505, 555, 560, 569, 627,
 629, 631, 633, 638, 642, 647, 651, 660
 \backslash newcounter 4
 \backslash newfam 200
 \backslash newif 5–18
 \backslash noexpand 45, 46, 78, 122, 206, 208, 210,
 225, 226, 228, 229, 258, 475, 564, 565
 \backslash nolimits 216
 \backslash npolint 465
 \backslash Nu 582
 \backslash nu 607
 \backslash number 566, 567
 \backslash numexpr
 542, 547, 551, 552, 558, 559, 566, 567

O

\backslash o 65
 \backslash oiint 462
 \backslash oint 462
 \backslash ointctrclockwise 463
 \backslash Omega 594
 \backslash omega 619
 \backslash Omicron 584
 \backslash omicron 609
 \backslash operator@font 243
 \backslash or .. 140, 151, 162, 171, 176, 181, 187, 193

P

\backslash P 48
 \backslash PackageError 31, 487
 \backslash PackageInfo 33
 \backslash PackageWarning 32
 \backslash papersize 30, 31, 75, 76
 \backslash parindent 20, 65
 \backslash Phi 591
 \backslash phi 616
 \backslash Pi 585
 \backslash pi 610
 \backslash pointint 465
 \backslash ProcessOptionsX 197
 \backslash protect 490
 \backslash providecommand 19, 55
 \backslash ProvidesPackage 1
 \backslash Psi 593
 \backslash psi 618

Q

\backslash Q 49

R

\backslash R 50, 61, 70
 \backslash relax ... 24, 27, 39, 45, 56, 74, 76, 78,
 106, 114, 129, 133, 159, 166, 167,
 185, 191, 204, 207, 212, 220, 222,
 224, 225, 227, 229, 230, 232, 233,
 235, 240–242, 244, 281, 282, 285,
 286, 447, 501, 511, 514, 536, 537,
 542, 547, 551, 552, 558, 559, 566, 567
 \backslash removenolimits 471
 \backslash RequirePackage 3
 \backslash reserved@a 46, 47, 122, 123, 254, 263, 265
 \backslash reserved@b .. 195, 197, 236, 238, 262–264

\reserved@c	112,	\theum@fam	275
128, 137, 165, 176, 196, 197, 237, 238		\toks@	224,
\reserved@d	120, 128, 137, 173, 176	225, 227, 228, 255, 257, 261, 278, 281	
\Rho	586	\tw@	40, 44, 51, 59, 67, 107, 111, 112, 115,
\rho	611	119, 120, 160, 164, 165, 168, 172, 173	
\rppolint	464		
\rule	257, 268		
		U	
S		\um@addto@mathmap	438, 450, 457
\scantokens	231, 234	\um@backslash	510, 531
\scpolint	465	\um@Break	21, 28
\set@mathdelimiter	174, 190	\um@char@num@range	246, 446, 522, 523, 525
\set@mathchar	65, 75	\um@char@range	
\set@mathdelimiter	125, 134, 186	... 245, 269, 284, 289, 310, 504, 506	
\set@mathsymbol	49, 57, 73	\um@def@itGreek	329, 647
\setbox	23, 31, 68, 76	\um@def@itgreek	330, 660
\setkeys	256	\um@def@itLatin	327, 631
\setmainfont	4, 37	\um@def@itlatin	328, 633
\SetMathCode	241, 460, 558	\um@def@numbers	311, 627
\setmathfont	19, 64, 243, 244, 490	\um@def@upGreek	329, 642
\SetSymbolFont	234	\um@def@upgreek	330, 651
\SetSymbolFont@	215, 241, 248	\um@def@upLatin	327, 629
\sf@size	261, 264	\um@def@uplatin	328, 638
\shipout	32, 77	\um@firstchar	509, 532
\Sigma	588	\um@firstof	530-532
\sigma	613	\um@FOR	23, 557, 562
\sixt@@n	41, 43,	\um@inp	556, 558, 561, 566
108, 110, 116, 118, 161, 163, 169, 171		\um@Loop	21, 25
\special	31, 76	\um@mathbb	290
\sqint	465	\um@mathbf	291, 299
\sqrt	479	\um@mathbffrak	302
\stepcounter	274	\um@mathbfit	301
\string	46, 52, 55, 61,	\um@mathbfsc	303
122, 129, 132, 139, 208, 211, 213,		\um@mathbfsf	304
214, 231, 254, 262, 266, 268, 272,		\um@mathbfsfit	305
274-276, 284, 288, 411, 459, 531, 532		\um@mathbfup	300
\sumint	463	\um@mathfrak	292
		\um@mathit	294
T		\um@mathmap@noparse	306, 435
\Tau	589	\um@mathmap@parse	308, 445
\tau	614	\um@mathscr	295
\testmath	14, 24-27, 56, 69-72	\um@mathsf	296
\tf@size	260, 261	\um@mathsf	297
\the	30, 75, 225, 228, 281	\um@mathsymbol	201, 408, 412
\Theta	577	\um@mathsymbol@noparse	272, 407
\theta	602	\um@mathsymbol@parse	276, 409

<code>\um@mathtt</code>	298	<code>\um@usv@bfgreek</code>	62,
<code>\um@mathup</code>	243, 293		92, 136, 165, 179, 183, 348, 364, 372
<code>\um@mk@alph</code>	<u>413</u>	<code>\um@usv@bfitGreek</code>	
<code>\um@mk@math</code>	413, 419–433		. 65, 91, 132, 160, 164, 349, 357, 360
<code>\um@mversion</code>	250, 251	<code>\um@usv@bfitgreek</code>	
<code>\um@nolimits</code>	215, <u>461</u> , 469, 470, 477		. 66, 93, 144, 161, 165, 350, 364, 372
<code>\um@PackageError</code>	31	<code>\um@usv@bfith</code>	94, 107, 126
<code>\um@PackageInfo</code>	33, 271, 411	<code>\um@usv@bfitLatin</code>	
<code>\um@PackageWarning</code>	32		. 63, 87, 119, 158, 162, 345, 352, 353
<code>\um@parse@range</code>	519, <u>533</u>	<code>\um@usv@bfitlatin</code>	
<code>\um@parse@term</code>	410, <u>505</u>		. 64, 89, 125, 159, 163, 346, 354, 355
<code>\um@Pool</code>	21, 22, 30	<code>\um@usv@bfitNabla</code>	
<code>\um@radicals</code>	218, <u>479</u>	 107, 118, 167, 382, 386, 387
<code>\um@resolve@greek</code>	<u>568</u>	<code>\um@usv@bfitpartial</code>	
<code>\um@scanactivedef</code>	208, <u>228</u>	 108, 124, 168, 393, 397, 398
<code>\um@scancharlet</code>	<u>228</u> , 239	<code>\um@usv@bfitvarepsilon</code>	
<code>\um@ScriptFeatures</code>	252, 263	 109, 145, 169, 365, 373
<code>\um@ScriptFont</code>	254, 262	<code>\um@usv@bfitvarkappa</code>	
<code>\um@ScriptScriptFeatures</code> ...	253, 266	 111, 147, 171, 367, 375
<code>\um@ScriptScriptFont</code>	255, 265	<code>\um@usv@bfitvarphi</code> 112, 148, 172, 368, 376	
<code>\um@set@mathsymbol</code>	202, <u>203</u>	<code>\um@usv@bfitvarpi</code> 114, 150, 174, 370, 378	
<code>\um@setmathalph</code>		<code>\um@usv@bfitvarrho</code> 113, 149, 173, 369, 377	
 2–84, 86–114, 117, 119,	<code>\um@usv@bfitvarTheta</code>	
	122, 123, 125, 126, 129, 130, 132,	 106, 108, 133, 166, 358, 361
	133, 136–142, 144–150, 153–226, <u>560</u>	<code>\um@usv@bfitvartheta</code>	
<code>\um@setmathcode</code> 110, 146, 170, 366, 374
	... 313–325, 332, 334, 337, 339,	<code>\um@usv@bflatin</code>	59,
	343–350, 352, 354, 357, 358, 360,		86, 117, 162, 176, 180, 343, 352, 353
	361, 364–370, 372–378, 381–384,	<code>\um@usv@bflatin</code>	60,
	386–389, 392–395, 397–400, <u>555</u> ,		88, 122, 163, 177, 181, 344, 354, 355
	628, 630, 632, 634–636, 639, 640,	<code>\um@usv@bfNabla</code> 96, 117, 185, 381, 382, 386	
	643–645, 648, 649, 652–658, 661–667	<code>\um@usv@bfnum</code>	
<code>\um@setsinglemathalph</code> ...	306, 308, 564		. 58, 82, 157, 175, 193, 196, 199, 213
<code>\um@setup@alphabets</code>	1, 405	<code>\um@usv@bfpartial</code>	
<code>\um@symfont</code> 98, 123, 186, 392, 393, 397
	270, 275, 278, 408, 412, 441, 453, 559	<code>\um@usv@bfscrLatin</code>	69, 197
<code>\um@usv@bbLatin</code>	44	<code>\um@usv@bfscrLatin</code>	70, 198
<code>\um@usv@bblatin</code>	45, 52	<code>\um@usv@bfsfGreek</code>	74, 202
<code>\um@usv@bbnum</code>	43	<code>\um@usv@bfsfgreek</code>	75, 203
<code>\um@usv@bfDigamma</code>	90, 97	<code>\um@usv@bfsfitGreek</code>	78, 216
<code>\um@usv@bfDigamma</code>	97, 105	<code>\um@usv@bfsfitgreek</code>	79, 217
<code>\um@usv@bffrakLatin</code>	67, 194	<code>\um@usv@bfsfitLatin</code>	76, 214
<code>\um@usv@bffrakLatin</code>	68, 195	<code>\um@usv@bfsfitlatin</code>	77, 215
<code>\um@usv@bfGreek</code>	61,	<code>\um@usv@bfsfitNabla</code> ..	120, 384, 388, 389
	90, 129, 164, 178, 182, 347, 357, 360	<code>\um@usv@bfsfitpartial</code> 126, 395, 399, 400	
		<code>\um@usv@bfsfLatin</code>	72, 200

$\backslash\mathrm{um@usv@bfsflatin}$	73, 201	$\backslash\mathrm{um@usv@itvarrho}$	13, 26, 41, 104, 113, 141, 149, 173, 191, 657, 666
$\backslash\mathrm{um@usv@bfsfNabla}$	119, 383, 384	$\backslash\mathrm{um@usv@itvarTheta}$	8, 21, 36, 99, 106, 130, 133, 166, 184, 649
$\backslash\mathrm{um@usv@bfsfnum}$	71	$\backslash\mathrm{um@usv@itvartheta}$	10, 23, 38, 101, 110, 138, 146, 170, 188, 654, 663
$\backslash\mathrm{um@usv@bfsfpartial}$. .	125, 394, 395, 399	$\backslash\mathrm{um@usv@Nabla}$	6, 19, 34, 96, 115, 153, 167, 185, 205, 219, 322, 332, 334
$\backslash\mathrm{um@usv@bfuph}$	106, 123	$\backslash\mathrm{um@usv@num}$	34, 43, 73, 76, 79, 82, 157, 175, 193, 196, 199, 213, 628
$\backslash\mathrm{um@usv@bfvarepsilon}$	91, 99, 137, 187, 365, 373	$\backslash\mathrm{um@usv@partial}$. .	7, 20, 35, 98, 121, 154, 168, 186, 206, 220, 324, 337, 339
$\backslash\mathrm{um@usv@bfvarkappa}$	93, 101, 139, 189, 367, 375	$\backslash\mathrm{um@usv@scrLatin}$	46, 53
$\backslash\mathrm{um@usv@bfvarphi}$	94, 102, 140, 190, 368, 376	$\backslash\mathrm{um@usv@scrLatin}$	47, 62
$\backslash\mathrm{um@usv@bfvarpi}$ 96, 104, 142, 192, 370, 378		$\backslash\mathrm{um@usv@sfitLatin}$	53, 77
$\backslash\mathrm{um@usv@bfvarrho}$	95, 103, 141, 191, 369, 377	$\backslash\mathrm{um@usv@sfitlatin}$	54, 78
$\backslash\mathrm{um@usv@bfvarTheta}$	89, 95, 130, 184, 358, 361	$\backslash\mathrm{um@usv@sflatin}$	51, 74
$\backslash\mathrm{um@usv@bfvartheta}$	92, 100, 138, 188, 366, 374	$\backslash\mathrm{um@usv@sflatin}$	52, 75
$\backslash\mathrm{um@usv@Digamma}$	81, 83, 97	$\backslash\mathrm{um@usv@sfnum}$	50, 73, 76
$\backslash\mathrm{um@usv@digamma}$	84, 88, 105	$\backslash\mathrm{um@usv@ttLatin}$	56, 80
$\backslash\mathrm{um@usv@frakLatin}$	48, 66, 194	$\backslash\mathrm{um@usv@ttlatin}$	57, 81
$\backslash\mathrm{um@usv@fraklatin}$	49, 72, 195	$\backslash\mathrm{um@usv@ttnum}$	55, 79
$\backslash\mathrm{um@usv@itGreek}$ 4, 17, 32, 41, 91, 129, 132, 160, 178, 202, 216, 320, 643, 648		$\backslash\mathrm{um@usv@upGreek}$ 4, 17, 32, 39, 90, 129, 132, 160, 178, 202, 216, 318, 643, 648	
$\backslash\mathrm{um@usv@itgreek}$. . . 5, 18, 33, 42, 93, 136, 144, 161, 179, 203, 217, 652, 661		$\backslash\mathrm{um@usv@upgreek}$ 5, 18, 33, 40, 92, 136, 144, 161, 179, 203, 217, 321, 652, 661	
$\backslash\mathrm{um@usv@ith}$	30, 31, 94, 98, 123, 126, 316, 635, 636, 640	$\backslash\mathrm{um@usv@upLatin}$. . . 2, 15, 28, 35, 44, 53, 66, 74, 77, 80, 86, 117, 119, 158, 176, 194, 197, 200, 214, 313, 630, 632	
$\backslash\mathrm{um@usv@itLatin}$. . . 2, 15, 28, 37, 44, 53, 66, 74, 77, 80, 87, 117, 119, 158, 176, 194, 197, 200, 214, 314, 630, 632		$\backslash\mathrm{um@usv@uplatin}$. . . 3, 16, 29, 36, 52, 62, 72, 75, 78, 81, 88, 122, 125, 159, 177, 195, 198, 201, 215, 317, 634, 639	
$\backslash\mathrm{um@usv@itlatin}$. . . 3, 16, 29, 38, 52, 62, 72, 75, 78, 81, 89, 122, 125, 159, 177, 195, 198, 201, 215, 315, 634, 639		$\backslash\mathrm{um@usv@varepsilon}$. 9, 22, 37, 82, 99, 137, 145, 169, 187, 207, 221, 653, 662	
$\backslash\mathrm{um@usv@itNabla}$	6, 19, 34, 107, 116, 155, 167, 185, 323, 332, 334	$\backslash\mathrm{um@usv@varkappa}$ 11, 24, 39, 84, 101, 139, 147, 171, 189, 209, 223, 655, 664	
$\backslash\mathrm{um@usv@itpartial}$	7, 20, 35, 108, 122, 156, 168, 186, 325, 337, 339	$\backslash\mathrm{um@usv@varphi}$. . . 12, 25, 40, 85, 102, 140, 148, 172, 190, 210, 224, 656, 665	
$\backslash\mathrm{um@usv@itvarepsilon}$ 9, 22, 37, 100, 109, 137, 145, 169, 187, 653, 662		$\backslash\mathrm{um@usv@varpi}$. . . 14, 27, 42, 87, 104, 142, 150, 174, 192, 212, 226, 658, 667	
$\backslash\mathrm{um@usv@itvarkappa}$	11, 24, 39, 102, 111, 139, 147, 171, 189, 655, 664	$\backslash\mathrm{um@usv@varrho}$. . . 13, 26, 41, 86, 103, 141, 149, 173, 191, 211, 225, 657, 666	
$\backslash\mathrm{um@usv@itvarphi}$	12, 25, 40, 103, 112, 140, 148, 172, 190, 656, 665	$\backslash\mathrm{um@usv@varTheta}$	8, 21, 36, 80, 95, 130, 133, 166, 184, 204, 218, 319, 644, 645, 649
$\backslash\mathrm{um@usv@itvarpi}$	14, 27, 42, 105, 114, 142, 150, 174, 192, 658, 667		

<code>\um@usv@vartheta</code>	10, 23, 38, 83, 100, 138, 146, 170, 188, 208, 222, 654, 663
<code>\um@zf@feature</code>	482, 492, 494
<code>\unicodemathgobble</code>	208, 236
<code>\UnicodeMathSymbol</code>	238, 272, 276
<code>\unless</code>	284, 507
<code>\updefault</code>	279
<code>\upGreek</code>	8, 41
<code>\upgreek</code>	9, 42
<code>\upint</code>	466
<code>\upLatin</code>	7, 39
<code>\uplatin</code>	6, 40
<code>\Upsilon</code>	590
<code>\upsilon</code>	615
<code>\usepackage</code>	2, 3, 5, 35, 36, 38
V	
<code>\varepsilon</code>	599, 620
<code>\varkappa</code>	622
<code>\varointclockwise</code>	463
<code>\varphi</code>	623
<code>\varpi</code>	626
<code>\varrho</code>	624, 625
<code>\varsigma</code>	612
<code>\varTheta</code>	587
<code>\vartheta</code>	621
<code>\vbox</code>	23, 31, 68, 76
<code>\version@elt</code>	213, 227
<code>\version@list</code>	218, 233, 249
<code>\voffset</code>	21, 66
W	
<code>\wd</code>	30, 75
X	
<code>\xdef</code>	187, 212, 233, 281, 477, 504
<code>\XeTeXdelcode</code>	224, 229
<code>\XeTeXdelimeter</code>	223, 228
<code>\XeTeXmathaccent</code>	233
<code>\XeTeXmathchardef</code>	210
<code>\XeTeXmathcode</code>	225, 230, 235, 242
<code>\XeTeXradical</code>	220
<code>\Xi</code>	583
<code>\xi</code>	608
<code>\XKV@rm</code>	267
Z	
<code>\Z</code>	51, 71
<code>\z@</code>	39–42, 51, 59, 67, 106–109, 112, 114–117, 120, 159–162, 165, 167–170, 173
<code>\zap@space</code>	504
<code>\Zeta</code>	575
<code>\zeta</code>	600
<code>\zf@family</code>	279
<code>\zf@fontspec</code>	258
<code>\zf@update@ff</code>	493, 495