

# Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2009/09/17 v0.4

## Abstract

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

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>	<b>4.3</b>	Other things	19
<b>2</b>	<b>Unicode maths font setup</b>	<b>2</b>	<b>5</b>	<b>Fundamentals</b>	<b>20</b>
2.1	Using multiple fonts	3	5.1	Enlarging the number of maths families	20
2.2	Script and scriptscript fonts/features	3	5.2	\DeclareMathSymbol for unicode ranges	20
<b>3</b>	<b>Maths input</b>	<b>3</b>	5.3	The main \setmathfont macro	22
3.1	Math ‘style’	4	5.4	(Big) operators	29
3.2	Bold style	5	5.5	Radicals	33
3.3	Sans serif style	6	5.6	Delimiters	33
3.4	All (the rest) of the math- ematical alphabets	7	5.7	Maths accents	36
3.5	Miscellanea	7	<b>6</b>	<b>Font features</b>	<b>37</b>
<b>I</b>	<b>The unicode-math pack- age</b>	<b>11</b>	6.1	OpenType maths font features	37
<b>4</b>	<b>Things we need</b>	<b>11</b>	6.2	Script and scriptscript font options	38
4.1	Package options	14	6.3	Range processing	38
4.2	Overcoming \@on- lypreamble	18	6.4	Resolving Greek symbol name control sequences	42

6.5	Setting up the mappings	44	<b>the NFSS</b>	<b>69</b>
			A.1 Overview	69
<b>7</b>	<b>Maths alphabets mapping definitions</b>	<b>45</b>	<b>III X<sub>Y</sub>TeX math font dimensions</b>	<b>70</b>
7.1	Non-bold math alphabets	48		
7.2	Bold math alphabets	52		
7.3	Definitions of the math symbols	58		
<b>8</b>	<b>Epilogue</b>	<b>59</b>	<b>IV Some manner of unit testing</b>	<b>75</b>
<b>II</b>	<b>stix table data extraction</b>	<b>67</b>	<b>B The regular weight alphabets</b>	<b>75</b>
<b>A</b>	<b>Documenting maths support in</b>		<b>C The bold alphabets</b>	<b>76</b>

## 1 Introduction

This document describes the unicode-math package, which is an *experimental* implementation of a macro to unicode glyph encoding for mathematical characters. Its intended use is for X<sub>Y</sub>TeX, although it is conjectured that some effect could be spent to create a cross-format package that would also work with LuaTeX.

Users who desire to specify maths alphabets only from various fonts may wish to use Andrew Moschou’s mathspec package instead.

## 2 Unicode maths font setup

In the ideal case, a single unicode font will contain all maths glyphs we need. The file unicode-math-table.tex (based on 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⟩}
```

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

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

Finally, maths versions must also be provided for. While I guess version selection in L<sup>A</sup>TeX will remain the same, the specification for choosing the version fonts will probably be an optional argument:

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

This has not been implemented yet.

Instances above of

`[<font features>]{<font name>}`

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

## 2.1 Using multiple fonts

There will probably be few cases where a single unicode maths font suffices (simply due to glyph coverage). The upcoming `stix` font comes to mind as a possible exception. It will therefore be necessary to delegate specific unicode ranges of glyphs to separate fonts:

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

where *<unicode range>* is a comma-separated list of unicode slots and ranges such as `{27D0-27EB,27FF,295B-297F}`. You may also use the macro for accessing the glyph, such as `\`, or whole collection of symbols with the same math type, such as `\mathopen`. (Only numerical slots, however, can be used in proper ranges.) This interface still requires some thought.

Not yet implemented: preset names ranges could be used in the range spec., such as `MiscMathSymbolsA`, with such ranges based on unicode chunks. The amount of optimisation required here to achieve acceptable performance has yet to be determined. Techniques such as saving out unicode subsets based on *<unicode range>* data to be `\input` in the next  $\text{\LaTeX}$  run are a possibility, but at this stage, performance without such measures seems acceptable.

## 2.2 Script and scriptscript fonts/features

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

Not yet implemented: Both of these options must be taken into account. I hope this will be mostly automatic from the users' points of view. The `+ssty` feature can be detected and applied automatically, and appropriate optical size information embedded in the fonts will ensure this latter case. Fine tuning should be possible automatically with fontspec options. We might have to wait until `MnMath`, for example, before we really know.

## 3 Maths input

$\text{\XeTeX}$ 's unicode support allows maths input through two methods. Like classical  $\text{\TeX}$ , macros such as `\alpha`, `\sum`, `\pm`, `\leq`, and so on, provide verbose access

to the entire repertoire of characters defined by unicode. The literal characters themselves may be used instead, for more readable input files.

### 3.1 Math ‘style’

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

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

The philosophy behind the interface to the mathematical alphabet symbols lies in  $\text{\LaTeX}$ ’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. Figure 1 on the following page shows every character under the effect of this package option.

### 3.2 Bold style

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

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

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*  
*abcdefghijklmnopqrstuvwxyz*  
*ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ*  
*αβγδεεζηθθικκλμνξοπωρρςστυφφχψω*

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*  
*abcdefghijklmnopqrstuvwxyz*  
*ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ*  
*αβγδεεζηθθικκλμνξοπωρρςστυφφχψω*

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*  
*abcdefghijklmnopqrstuvwxyz*  
*ABΓΔΕΖΗΘΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ*  
*αβγδεεζηθθικκλμνξοπωρρςστυφφχψω*

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

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

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

Package option	Example	
	Latin	Greek
<code>bold-style=ISO</code>	<b>(<i>a</i>, <i>z</i>, <i>B</i>, <i>X</i>)</b>	<b>(<i>α</i>, <i>β</i>, <i>Γ</i>, <i>Ξ</i>)</b>
<code>bold-style=TeX</code>	<b>(<i>a</i>, <i>z</i>, <i>B</i>, <i>X</i>)</b>	<b>(<i>α</i>, <i>β</i>, <i>Γ</i>, <i>Ξ</i>)</b>
<code>bold-style=French</code>	<b>(<i>a</i>, <i>z</i>, <i>B</i>, <i>X</i>)</b>	<b>(<i>α</i>, <i>β</i>, <i>Γ</i>, <i>Ξ</i>)</b>

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

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

### 3.3 Sans serif style

Unicode contains upright and italic, medium and bold mathematical alphabet characters. These may be explicitly selected with the `\mathsfup`, `\mathsfit`, `\mathbfsfup`, and `\mathbfsfit` commands discussed in section §3.4.

How should the generic `\mathsf` behave? Unlike bold, sans serif is used much more sparingly in mathematics. I've seen recommendations to typeset tensors in sans serif italic or sans serif italic bold (e.g., examples in the `isomath` and `mattens` packages). But  $\text{\LaTeX}$ 's `\mathsf` is *upright* sans serif.

Therefore I reluctantly add the package options `[sans-style=TeX]` and `[sans-style=ISO]` to control the behaviour of `\mathsf`. The `TeX` style sets up the command to use the seemingly-useless upright sans serif, including Greek; the `ISO` style switches to using italic in both Latin and Greek alphabets. In other words,

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ**  
**αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ**

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

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ**  
**αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ**

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

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**  
**abcdefghijklmnopqrstuvwxyz**  
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΘΣΤΥΦΧΨΩ**  
**αβγδεζηθικλμνξοπρςστυφχψωεθκφρτ**

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

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

this option simply changes the meaning of `\mathsf` to either `\mathsfup` or `\mathsf-fit`, respectively. Please let me know if more granular control is necessary here.

There is also a `[sans-style=literal]` setting, set automatically with `[math-style=literal]`, which retains the uprightness of the input characters used when selecting the sans serif output.

### 3.3.1 What about bold sans serif?

While you might want your bold upright and your sans serif italic, I don't believe you'd also want your bold sans serif upright (or all vice versa, if that's even conceivable). Therefore, bold sans serif follows from the setting for sans serif; it is completely independent of the setting for bold.

In other words, `\mathbfsf` is `\mathbfsfup` or `\mathbfsffit` based on `[sans-style=TeX]` or `[sans-style=ISO]`, respectively. And `[sans-style=literal]` causes `\mathbfsf` to retain the same italic or upright shape as the input, and turns it bold sans serif.

Note well! There is no medium-weight sans serif Greek alphabet in unicode; therefore, `\mathsf{\alpha}` does not make sense (simply produces 'α') while `\mathbfsf{\alpha}` gives 'α'.

## 3.4 All (the rest) of the mathematical alphabets

Unicode contains separate codepoints for most if not all variations of alphabet shape one may wish to use in mathematical notation. The complete list is shown

Table 3: Mathematical alphabets defined in unicode. Black dots indicate an alphabet exists in the font specified; grey dots indicate shapes that should always be taken from the upright font even in the italic style.

Font				Alphabet		
Style	Shape	Series	Switch	Latin	Greek	Numerals
Serif	Upright	Normal	<code>\mathup</code>	•	•	•
		Bold	<code>\mathbfup</code>	•	•	•
	Italic	Normal	<code>\mathit</code>	•	•	•
		Bold	<code>\mathbfit</code>	•	•	•
Sans serif	Upright	Normal	<code>\mathsfup</code>	•		•
	Italic	Normal	<code>\mathsfit</code>	•		•
	Upright	Bold	<code>\mathsfbfup</code>	•	•	•
	Italic	Bold	<code>\mathsfbfit</code>	•	•	•
Typewriter	Upright	Normal	<code>\mathtt</code>	•		•
Double-struck	Upright	Normal	<code>\mathbb</code>	•		•
Script	Upright	Normal	<code>\mathscr</code>	•		
		Bold	<code>\matbfscr</code>	•		
Fraktur	Upright	Normal	<code>\mathfrak</code>	•		
		Bold	<code>\mathbffrac</code>	•		

in table 3. Some of these have been covered in the previous sections.

At present, the math font switching commands do not nest; therefore if you want sans serif bold, you must write `\mathsfbf{...}` rather than `\mathbf{\mathsf{...}}`. This may change in the future.

## 3.5 Miscellanea

### 3.5.1 Nabla

The symbol  $\nabla$  comes in the six forms shown in table 4. We want an individual option to specify whether we want upright or italic nabla by default (when either upright or italic nabla is used in the source).  $\text{\TeX}$  classically uses an upright nabla, but iso standards differ (I think). The package options `nabla=upright` and `nabla=italic` switch between the two choices. This is then inherited through `\mathbf`; `\mathit` and `\mathup` can be used to force one way or the other.

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



Table 4: The various forms of nabla.

Description		Glyph
Upright	Serif	$\nabla$
	Bold serif	<b><math>\nabla</math></b>
	Bold sans	<b><math>\nabla</math></b>
Italic	Serif	<i><math>\nabla</math></i>
	Bold serif	<b><i><math>\nabla</math></i></b>
	Bold sans	<b><i><math>\nabla</math></i></b>

Table 5: The various forms of the partial differential. Note that in the fonts used to display these glyphs, the first upright partial is incorrectly shown in an italic style.

Description		Glyph
Regular	Upright	$\partial$
	Italic	<i><math>\partial</math></i>
Bold	Upright	<b><math>\partial</math></b>
	Italic	<b><i><math>\partial</math></i></b>
Sans bold	Upright	<b><math>\partial</math></b>
	Italic	<b><i><math>\partial</math></i></b>

### 3.5.2 Partial

The same applies to the symbols U+2202: PARTIAL DIFFERENTIAL and U+1D715: MATH ITALIC PARTIAL DIFFERENTIAL.

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

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

See table 5 for the variations on the partial differential symbol.

### 3.5.3 Epsilon and phi: $\epsilon$ vs. $\varepsilon$ and $\phi$ vs. $\varphi$

$\TeX$  defines `\epsilon` to look like  $\varepsilon$  and `\varepsilon` to look like  $\epsilon$ . The Unicode glyph directly after delta and before zeta is ‘epsilon’ and looks like  $\epsilon$ ; there is a subsequent variant of epsilon that looks like  $\varepsilon$ . This creates a problem. People

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

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

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

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

U+3B5: GREEK SMALL LETTER EPSILON

U+3F5: GREEK LUNATE EPSILON SYMBOL

U+3C6: GREEK SMALL LETTER PHI

U+3D5: GREEK SMALL LETTER SCRIPT PHI

### 3.5.4 Primes

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

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

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

### 3.5.5 Unicode subscripts and superscripts

You may, if you wish, use unicode subscripts and superscripts in your source document. For basic expressions, the use of these characters can make the input more readable. Adjacent sub- or super-scripts will be concatenated into a single expression.

The range of subscripts and superscripts supported by this package are shown in figures 3 and 4. Please request more if you think it is appropriate.

A 0 1 2 3 4 5 6 7 8 9 + - = ( ) i n Z

Figure 3: The unicode superscripts supported as input characters. These are the literal glyphs from Charis SIL, not the output seen when used for maths input. The ‘A’ and ‘Z’ are to provide context for the size and location of the superscript glyphs.

A 0 1 2 3 4 5 6 7 8 9 + - = ( ) a e i o r u v x β γ ρ ϕ χ Z

Figure 4: The unicode subscripts supported as input characters. See note from figure 3.

### 3.5.6 Vertical bar ‘|’

### 3.5.7 Colon ‘:’

### 3.5.8 Normalising some input characters

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

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

(Not yet implemented.)

## File I

# The unicode-math package

This is the package.

```
1 \ProvidesPackage{unicode-math}
2 [2009/09/17 v0.4 Unicode maths in XeLaTeX]
```

## 4 Things we need

### Packages

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

```

5 \RequirePackage{fontspec}
   Start using LATEX3 — finally!
6 \ExplSyntaxOn

```

### Counters and conditionals

```

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

```

For math-style:

```

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

```

For bold-style:

```

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

```

For nabla:

```

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

```

### 4.0.9 Alphabet unicode positions

Before we begin, let's define the positions of the various unicode alphabets so that our code is a little more readable.<sup>2</sup>

```

23 \def\um@usv@num{'\0}
24 \def\um@usv@upLatin{'\A}
25 \def\um@usv@uplatin{'\a}
26 \def\um@usv@upGreek{"391}
27 \def\um@usv@upgreek{"3B1}
28 \def\um@usv@itLatin{"1D434}
29 \def\um@usv@itlatin{"1D44E}
30 \def\um@usv@itGreek{"1D6E2}
31 \def\um@usv@itgreek{"1D6FC}
32 \def\um@usv@bbnum{"1D7D8}
33 \def\um@usv@bbLatin{"1D538}
34 \def\um@usv@bblatin{"1D552}
35 \def\um@usv@scrLatin{"1D49C}

```

---

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

```

36 \def\um@usv@scrLatin{"1D4B6}
37 \def\um@usv@frakLatin{"1D504}
38 \def\um@usv@frakLatin{"1D51E}
39 \def\um@usv@sfnum{"1D7E2}
40 \def\um@usv@sfupLatin{"1D5A0}
41 \def\um@usv@sfLatin {"1D5A0}
42 \def\um@usv@sfuplatin{"1D5BA}
43 \def\um@usv@sfitLatin{"1D608}
44 \def\um@usv@sfitlatin{"1D622}
45 \def\um@usv@ttnum{"1D7F6}
46 \def\um@usv@ttLatin{"1D670}
47 \def\um@usv@ttlatin{"1D68A}

```

**Bold:**

```

48 \def\um@usv@bfnum{"1D7CE}
49 \def\um@usv@bfupLatin{"1D400}
50 \def\um@usv@bfLatin {"1D400}
51 \def\um@usv@bfuplatin{"1D41A}
52 \def\um@usv@bfupGreek{"1D6A8}
53 \def\um@usv@bfupgreek{"1D6C2}
54 \def\um@usv@bfitLatin{"1D468}
55 \def\um@usv@bfitlatin{"1D482}
56 \def\um@usv@bfitGreek{"1D71C}
57 \def\um@usv@bfitgreek{"1D736}
58 \def\um@usv@bffrakLatin{"1D56C}
59 \def\um@usv@bffrakLatin{"1D586}
60 \def\um@usv@bfscrLatin{"1D4D0}
61 \def\um@usv@bfscrLatin{"1D4EA}
62 \def\um@usv@bfsfnum{"1D7EC}
63 \def\um@usv@bfsfupLatin{"1D5D4}
64 \def\um@usv@bfsfLatin {"1D5D4}
65 \def\um@usv@bfsfuplatin{"1D5EE}
66 \def\um@usv@bfsfupGreek{"1D756}
67 \def\um@usv@bfsfupgreek{"1D770}
68 \def\um@usv@bfsfitLatin{"1D63C}
69 \def\um@usv@bfsfitlatin{"1D656}
70 \def\um@usv@bfsfitGreek{"1D790}
71 \def\um@usv@bfsfitgreek{"1D7AA}

```

**Greek variants:**

```

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

```

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

**Bold:**

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

**Italic Greek variants:**

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

**Bold:**

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

**Nabla:**

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

**Partial:**

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

## 4.1 Package options

xkeyval's package support is used here.

### math-style

```
119 \define@choicekey*{unicode-math.sty}
120   {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{
121   \ifcase\@tempb\relax
122     \@um@upGreekfalse
123     \@um@upgreekfalse
124     \@um@upLatinfalse
125     \@um@uplatinfalse
126     \@um@bfupGreekfalse
127     \@um@bfupgreekfalse
128     \@um@uppartialfalse
129     \@um@bfupLatinfalse
130     \@um@bfuplatinfalse
131     \@um@upNablafalse
132     \bool_set_false:N \g_um_upsans_bool
133     \bool_set_false:N \g_um_texgreek_bool
134   \or
135     \@um@upGreektrue
136     \@um@upgreekfalse
137     \@um@upLatinfalse
138     \@um@uplatinfalse
139     \@um@bfupGreektrue
140     \@um@bfupgreekfalse
141     \@um@uppartialfalse
142     \@um@bfupLatintrue
143     \@um@bfuplatintrue
144     \@um@upNabltrue
145     \bool_set_true:N \g_um_upsans_bool
146     \bool_set_true:N \g_um_texgreek_bool
147   \or
148     \@um@upGreektrue
149     \@um@upgreektrue
150     \@um@upLatintrue
151     \@um@uplatinfalse
152     \@um@bfupGreektrue
153     \@um@bfupgreektrue
154     \@um@uppartialtrue
155     \@um@bfupLatintrue
156     \@um@bfuplatintrue
157     \@um@upNabltrue
158     \bool_set_true:N \g_um_upsans_bool
159     \bool_set_false:N \g_um_texgreek_bool
```

```

160 \or
161   \um@literaltrue
162   \um@bfliteraltrue
163   \bool_set_true:N \g_um_sfliteral_bool
164   \bool_set_false:N \g_um_texgreek_bool
165 \fi
166 }

```

### **bold-style**

```

167 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,french,literal}{
168   \ifcase\@tempb\relax
169     \um@bfupGreekfalse
170     \um@bfupgreekfalse
171     \um@uppartialfalse
172     \um@bfupLatinfalse
173     \um@bfuplatinfalse
174   \or
175     \um@bfupGreektrue
176     \um@bfupgreekfalse
177     \um@uppartialfalse
178     \um@bfupLatintrue
179     \um@bfuplatintrue
180   \or
181     \um@bfupGreektrue
182     \um@bfupgreektrue
183     \um@uppartialtrue
184     \um@bfupLatintrue
185     \um@bfuplatintrue
186   \or
187     \um@bfliteraltrue
188 \fi
189 }

```

### **sans-style**

```

190 \bool_new:N \g_um_upsans_bool
191 \bool_new:N \g_um_sfliteral_bool
192 \define@choicekey*{unicode-math.sty}
193   {sans-style}[\@tempa\@tempb]{iso,tex,literal}{
194   \ifcase\@tempb\relax
195     \bool_set_false:N \g_um_upsans_bool
196   \or
197     \bool_set_true:N \g_um_upsans_bool
198   \or
199     \bool_set_true:N \g_um_sfliteral_bool
200 \fi

```



201 }

### Symbol obliqueness

```
202 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright,italic}{
203   \ifcase\@tempb\relax
204     \@um@upNablatrue
205   \or
206     \@um@upNablafalse
207   \fi
208 }
209 \cs_set:Nn \um_setup_nabla: {
210   \if@um@upNabla
211     \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@Nabla }
212     \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfNabla }
213     \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfNabla }
214   \else
215     \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@itNabla }
216     \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfitNabla }
217     \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfitNabla }
218   \fi
219 }
220 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{
221   \ifcase\@tempb\relax
222     \@um@uppartialtrue
223   \or
224     \@um@uppartialfalse
225   \fi
226 }
227 \cs_set:Nn \um_setup_partial: {
228   \if@um@uppartial
229     \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@partial }
230     \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfpartial }
231     \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfpartial }
232   \else
233     \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@itpartial }
234     \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfitpartial }
235     \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfitpartial }
236   \fi
237 }
```

### Epsilon and phi shapes

```
238 \define@choicekey*{unicode-math.sty}{vargreek-shape}[\@tempa\@tempb]{unicode,TeX}{
239   \ifcase\@tempb\relax
240     \bool_set_false:N \g_um_texgreek_bool
241   \or
```

```

242     \bool_set_true:N \g_um_texgreek_bool
243     \fi
244 }
245 \ExecuteOptionsX{math-style=TeX}
246 \ProcessOptionsX

```

## 4.2 Overcoming \@onlypreamble

The requirement of only setting up the maths fonts is now removed. The following list might be overly ambitious.

```

247 \tl_map_inline:nn {
248   \new@mathgroup
249   \cdp@list
250   \cdp@elt
251   \DeclareMathSizes
252   \@DeclareMathSizes
253   \newmathalphabet
254   \newmathalphabet@
255   \newmathalphabet@@
256   \DeclareMathVersion
257   \define@mathalphabet
258   \define@mathgroup
259   \addtoversion
260   \version@list
261   \version@elt
262   \alpha@list
263   \alpha@elt
264   \restore@mathversion
265   \init@restore@version
266   \dorestore@version
267   \process@table
268   \new@mathversion
269   \DeclareSymbolFont
270   \group@list
271   \group@elt
272   \new@symbolfont
273   \SetSymbolFont
274   \SetSymbolFont@
275   \get@cdp
276   \DeclareMathAlphabet
277   \new@mathalphabet
278   \SetMathAlphabet
279   \SetMathAlphabet@
280   \DeclareMathAccent
281   \set@mathaccent
282   \DeclareMathSymbol

```

```

283 \set@mathchar
284 \set@mathsymbol
285 \DeclareMathDelimiter
286 \@xxDeclareMathDelimiter
287 \@DeclareMathDelimiter
288 \xDeclareMathDelimiter
289 \set@mathdelimiter
290 \set@@mathdelimiter
291 \DeclareMathRadical
292 \mathchar@type
293 \DeclareSymbolFontAlphabet
294 \DeclareSymbolFontAlphabet@
295 }{
296 \tl_remove_in:Nn \@preamblecmds {\do#1}
297 }

```

### 4.3 Other things

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

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

---

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

---

```

298 \def\um@fontdimen@percent#1#2{
299   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax
300 }

```

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

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

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

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

```

301 \def\um@scaled@apply#1#2#3{
302   \ifx#1\scriptstyle
303     #2\um@fontdimen@percent{10}\um@font#3
304   \else
305     \ifx#1\scriptscriptstyle
306       #2\um@fontdimen@percent{11}\um@font#3
307     \else
308       #2#3%
309     \fi

```

```

310 \fi
311 }

```

## 5 Fundamentals

### 5.1 Enlarging the number of maths families

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

```

312 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cc1vi}
313 \let\newfam\new@mathgroup

```

This is sufficient for L<sup>A</sup>T<sub>E</sub>X's `\DeclareSymbolFont`-type commands to be able to define 256 named maths fonts. Now we need a new `\DeclareMathSymbol`.

### 5.2 `\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
314 \def \um@mathsymbol#1#2#3#4{
315   \expandafter\um@set@mathsymbol\csname sym#3\endcsname#1#2{#4}}

```

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

```

\um@set@mathsymbol #1 : Symbol font number
                   #2 : Symbol macro, e.g., \alpha
                   #3 : Type, e.g., \mathalpha
                   #4 : Slot, e.g., "221E

```

If the symbol definition is for a macro. There are a bunch of tests to perform to process the various characters.

```

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

```

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

```

317 \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.

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

```

318   \begingroup

```

```

319     \char_make_active:n {#4}
320     \global\mathcode#4="8000\relax
321     \um@scanactivedef #4 \@nil { \csname\string#2@op\endcsname }
322     \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`.

```

323     \expandafter\global\expandafter\XeTeXmathchardef
324     \csname\string#2@sym\endcsname
325     =" \mathchar@type#3 #1 #4\relax

```

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

```

326     \cs_gset:cpn { \string#2 @op } {
327         \csname\string#2@sym\endcsname
328         \expandafter\in@\expandafter#2\expandafter{\um@nolimits}
329         \ifin@
330             \expandafter\nolimits
331         \fi
332     }

```

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

```

333     \else

```

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

```

334     \expandafter\in@\expandafter#2\expandafter{\um@radicals,}
335     \ifin@
336     \cs_gset:cpn {\cs_to_str:N #2 sign} { \XeTeXradical #1 #4 \relax }
337     \else

```

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

```

338     \ifx\mathopen#3\relax
339         \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
340         \global\XeTeXdelcode#4=#1 #4\relax
341         \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
342     \else
343         \ifx\mathclose#3\relax
344             \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
345             \global\XeTeXdelcode#4=#1 #4\relax
346             \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
347         \else

```

## Accents

```

348         \ifx\mathaccent#3\relax
349         \cs_gset:Npx #2 {\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}
350         \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.

```

351         \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
352     \fi
353 \fi
354 \fi
355 \fi
356 \fi
357 }

```

`\um_set_mathcode:nnnn` [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 inside math alphabet switches.

```

358 \cs_set:Nn \um_set_mathcode:nnnn {
359     \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax
360 }

```

## 5.3 The main `\setmathfont` macro

Here's the simplest usage:

---

$Ax \triangleq \nabla \times \mathcal{Z}$	<pre> \setmathfont{Asana Math} \$Ax \eqdef \nabla \times \mathscr{Z} </pre>
---	---

---

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

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

---

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

```

361 \DeclareDocumentCommand \setmathfont { O{} m } {

```

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

```
362 \let\glb@currsizel\relax
```

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

```
363 \let\um@char@range\@empty
364 \let\um@char@num@range\@empty
```

- Tell fontspec that maths font features are actually allowed.

```
365 \um@fontspec@featuretrue
```

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

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

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

```
367 \def\um@mversion{normal}
368 \DeclareMathVersion{\um@mversion}
```

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

```
369 \tl_set:Nn \l_um_script_features_tl {ScriptStyle}
370 \tl_set:Nn \l_um_sscript_features_tl {ScriptScriptStyle}
371 \tl_set:Nn \l_um_script_font_tl {#2}
372 \tl_set:Nn \l_um_sscript_font_tl {#2}
```

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

```
373 \setkeys*[um]{options}{#1}
374 \edef\@tempa{\noexpand\zf@fontspec{
375   Script = Math,
376   SizeFeatures = {
377     {Size = \tf@size-} ,
378     {Size = \sf@size-\tf@size ,
379       Font = \l_um_script_font_tl ,
380       \l_um_script_features_tl
381     } ,
382     {Size = -\sf@size ,
383       Font = \l_um_sscript_font_tl ,
```

```

384         \l_um_sscript_features_tl
385     },
386     \XKV@rm
387     }{#2}
388 }
389 \@tempa

```

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

Check for the correct number of `\fontdimens`:

```

391 \font\um@font="#2"\relax
392 \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt\relax
393     \um@ot@math@true
394 \else
395     \PackageWarningNoLine{unicode-math}{
396         The~ font~ '#2' ~is~ not~ a~ valid~ OpenType~ maths~ font.~
397         Some~ maths~ features~ will~ not~ be~ available~ or~ behave~
398         in~ a~ substandard~ manner
399     }
400 \fi

```

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

- Math symbols are defined with `\UnicodeMathSymbol`; see section §5.3.1 for the individual definitions

```

401 \ifx\um@char@range\@empty
402     \tl_set:Nn \um_symfont_tl {um@allsym}
403     \PackageInfo{unicode-math}{Defining~ the~ default~ maths~ font~ as~ '#2'}
404     \cs_set_eq:NN \UnicodeMathSymbol \um_process_symbol_noparse:nnnn
405     \cs_set_eq:NN \um_mathmap:Nnn \um_mathmap_noparse:Nnn
406     \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_noparse:nnn
407     \cs_set_eq:NN \um_maybe_init_alphabet:n \um_init_alphabet:n
408 \else
409     \stepcounter{um@fam}
410     \tl_set:Nx \um_symfont_tl {um@fam\theum@fam}
411     \cs_set_eq:NN \UnicodeMathSymbol \um_process_symbol_parse:nnnn
412     \cs_set_eq:NN \um_mathmap:Nnn \um_mathmap_parse:Nnn
413     \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_parse:nnn
414     \cs_set_eq:NN \um_maybe_init_alphabet:n \use_none:n
415 \fi

```

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

```

416 \DeclareSymbolFont{\um_symfont_tl}
417     {\encodingdefault}{\zf@family}{\mddefault}{\updefault}

```



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

```
418 \input{unicode-math-table.tex}
```

Finally,

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

```
419 \um_setup_shapes:
420 \um_remap_symbols:
421 \um_setup_mathactives:
422 \um_setup_alphanum:
423 \um_setup_alphabets:
```

End of the `\setmathfont` macro.

```
424 }

425 \cs_new:Nn \um_setup_shapes: {
426   \um_setup_nabla:
427   \um_setup_partial:
428 }
```

### 5.3.1 Functions for setting up symbols with mathcodes

`\um_process_symbol_noparse:nnnn` If the Range font feature has been used, then only a subset of the unicode glyphs are to be defined. See section §6.3 for the code that enables this.

```
429 \cs_set:Nn \um_process_symbol_noparse:nnnn {
430   \um@mathsymbol{#2}{#3}{\um_symfont_tl}{#1}
431 }

432 \cs_set:Nn \um_process_symbol_parse:nnnn {
433   \um@parse@term{#1}{#2}{#3}{
434     \um_process_symbol_noparse:nnnn{#1}{#2}{#3}{#4}
435   }
436 }
```

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

```
\um_remap_symbol_noparse:nnn
\um_remap_symbol_parse:nnn
437 \cs_new:Nn \um_remap_symbols: {
438   \um_remap_symbol:nnn{"2D}{\mathbin}{"02212}% hyphen to minus
```

```

439 \if@um@literal
440   \um_remap_symbol:nnn {\um@usv@Nabla}{\mathord}{\um@usv@Nabla}
441   \um_remap_symbol:nnn {\um@usv@itNabla}{\mathord}{\um@usv@itNabla}
442   \um_remap_symbol:nnn {\um@usv@partial}{\mathord}{\um@usv@partial}
443   \um_remap_symbol:nnn {\um@usv@itpartial}{\mathord}{\um@usv@itpartial}
444 \else
445   \um_remap_symbol:nnn {\um@usv@Nabla,\um@usv@itNabla}{\mathord}{\um@Nabla_up_or_it_usv}
446   \um_remap_symbol:nnn {\um@usv@partial,\um@usv@itpartial}{\mathord}{\um@partial_up_or_it_usv}
447 \fi

```

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

```

448 \if@um@bfliteral
449   \um_remap_symbol:nnn {\um@usv@bfNabla}{\mathord}{\um@usv@bfNabla}
450   \um_remap_symbol:nnn {\um@usv@bfitNabla}{\mathord}{\um@usv@bfitNabla}
451   \um_remap_symbol:nnn {\um@usv@bfsfNabla}{\mathord}{\um@usv@bfsfNabla}
452   \um_remap_symbol:nnn {\um@usv@bfsfitNabla}{\mathord}{\um@usv@bfsfitNabla}
453   \um_remap_symbol:nnn {\um@usv@bfpartial}{\mathord}{\um@usv@bfpartial}
454   \um_remap_symbol:nnn {\um@usv@bfitpartial}{\mathord}{\um@usv@bfitpartial}
455   \um_remap_symbol:nnn {\um@usv@bfsfpartial}{\mathord}{\um@usv@bfsfpartial}
456   \um_remap_symbol:nnn {\um@usv@bfsfitpartial}{\mathord}{\um@usv@bfsfitpartial}
457 \else
458   \um_remap_symbol:nnn {\um@usv@bfNabla,\um@usv@bfitNabla}{\mathord}{\um@bfNabla_up_or_it_usv}
459   \um_remap_symbol:nnn {\um@usv@bfsfNabla,\um@usv@bfsfitNabla}{\mathord}{\um@bfsfNabla_up_or_it_usv}
460   \um_remap_symbol:nnn {\um@usv@bfpartial,\um@usv@bfitpartial}{\mathord}{\um@bfpartial_up_or_it_usv}
461   \um_remap_symbol:nnn {\um@usv@bfsfpartial,\um@usv@bfsfitpartial}{\mathord}{\um@bfsfpartial_up_or_it_usv}
462 \fi
463 }

```

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

```

464 \cs_new:Nn \um_remap_symbol_parse:nnn {
465   \um@parse@term {#3} {\@nil} {#2} {
466     \um_remap_symbol_noparse:nnn {#1} {#2} {#3}
467   }
468 }
469 \cs_new:Nn \um_remap_symbol_noparse:nnn {
470   \clist_map_inline:nn {#1} {
471     \um_set_mathcode:nnnn {##1} {#2} {\um_symfont_tl} {#3}
472   }
473 }

```

### 5.3.2 Active math characters

There are more math active chars later in the subscript/superscript section. But they don't need to be able to be typeset directly.

`\um_setup_mathactives:`

```

474 \cs_new:Nn \um_setup_mathactives: {
475   \um_make_mathactive:nNN {"2032} \primesingle \mathord
476 }

```

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

```

477 \cs_new:Nn \um_make_mathactive:nNN {
478   \XeTeXmathchardef #2 = "\mathchar@type #3
479                               \csname sym\um_symfont_t1\endcsname
480                               #1 \scan_stop:
481   \XeTeXmathcodenum #1 = "1FFFFF \scan_stop:
482 }

```

### 5.3.3 Maths alphabets' character mapping

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

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

```

483 \cs_set:Nn \um_setup_alphanum: {
484   \ifx\um@char@range\@empty
485     \um_map_chars_numbers:nn {\um@usv@num}{\um@usv@num}

```

#### Normal weight

```

486   \if@um@literal
487     \um_setup_literals:
488   \else
489     \um_setup_Latin:
490     \um_setup_latin:
491     \um_setup_Greek:
492     \um_setup_greek:
493   \fi

```

#### Bold

```

494   \if@um@bfliteral
495     \um_setup_bf_literals:
496   \else
497     \if@um@bfupLatin
498       \um_map_chars_latin:nn {\um@usv@bfupLatin,\um@usv@bfitLatin}{\um@usv@bfupLatin}
499     \else
500       \um_map_chars_latin:nn {\um@usv@bfupLatin,\um@usv@bfitLatin}{\um@usv@bfitLatin}
501     \fi
502     \if@um@bfuplatin
503       \um_map_chars_latin:nn {\um@usv@bfuplatin,\um@usv@bfitlatin}{\um@usv@bfuplatin}

```

```

504     \else
505     \um_map_chars_latin:nn {\um@usv@bfuplatin,\um@usv@bfitlatin}{\um@usv@bfitlatin}
506     \fi
507     \if@um@bfupGreek
508     \um_map_chars_greek:nn {\um@usv@bfupGreek,\um@usv@bfitGreek}{\um@usv@bfupGreek}
509     \um_map_char:nn {\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfvarTheta}
510     \else
511     \um_map_chars_greek:nn {\um@usv@bfupGreek,\um@usv@bfitGreek}{\um@usv@bfitGreek}
512     \um_map_char:nn {\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um@usv@bfitvarTheta}
513     \fi
514     \if@um@bfupgreek
515     \um_map_chars_greek:nn {\um@usv@bfupgreek,\um@usv@bfitgreek}{\um@usv@bfupgreek}
516     \um_map_char:nn {\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfvarepsilon}
517     \um_map_char:nn {\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}
518     \um_map_char:nn {\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}
519     \um_map_char:nn {\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}
520     \um_map_char:nn {\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}
521     \um_map_char:nn {\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}
522     \else
523     \um_map_chars_greek:nn {\um@usv@bfupgreek,\um@usv@bfitgreek}{\um@usv@bfitgreek}
524     \um_map_char:nn {\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}
525     \um_map_char:nn {\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}
526     \um_map_char:nn {\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}
527     \um_map_char:nn {\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}
528     \um_map_char:nn {\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}
529     \um_map_char:nn {\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}
530     \fi
531     \fi
532     \else
: TODO : what is supposed to happen here?
533     \fi
534 }

```

### 5.3.4 Functions for setting up the maths alphabets

`\um_mathmap_noparse:Nnn` #1 : Maths alphabet, *e.g.*, `\mathbb`  
#2 : Input slot(s), *e.g.*, the slot for ‘A’ (comma separated)  
#3 : Output slot, *e.g.*, the slot for ‘A’  
Adds `\um_set_mathcode:nnnn` 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.

```

535 \cs_set:Nn \um_mathmap_noparse:Nnn {
536   \clist_map_inline:nn {#2} {
537     \exp_args:No \um@addto@mathmap \um_symfont_tl {##1}{#1}{#3}
538   }

```

```

539 }

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

540 \cs_set:Nn \um_mathmap_parse:Nnn {
541   \clist_map_inline:Nn \um@char@num@range {
542     \ifnum##1=#3\relax
543       \clist_map_inline:nn {#2} {
544         \exp_args:No \um@addto@mathmap \um_symfont_t1 {####1}{#1}{#3}
545       }
546     \fi
547   }
548 }

\um@addto@mathmap #1 : Math symbol font, always/usually the expansion of \um_symfont_t1
#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_t1 can be expanded before entering the
\g@addto@macro command.

549 \newcommand\um@addto@mathmap[4]{
550   \expandafter\g@addto@macro
551   \csname um_setup_\cs_to_str:N #3:\endcsname{
552     \um_set_mathcode:nnnn{#2}{\mathalpha}{#1}{#4}
553   }
554 }

```





























## 5.4 (Big) operators







Turns out that  $\text{\LaTeX}$  is clever enough to deal with big operators for us automatically with  $\text{\XeTeXmathchardef}$ . Amazing!


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

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

USV	Ex.	Macro	Description
U+02140	$\sum_{001}^1$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod_{001}^1$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod_{001}^1$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum_{001}^1$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int_0^1$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint_0^1$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint_0^1$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	$\oint_0^1$	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oiint_0^1$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oiint_0^1$	<code>\oiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\oint_0^1$	<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232	$\oint_0^1$	<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\oint_0^1$	<code>\ointctrackwise</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0	$\bigwedge_{001}^1$	<code>\bigwedge</code>	LOGICAL OR OPERATOR
U+022C1	$\bigvee_{001}^1$	<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2	$\bigcap_{001}^1$	<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3	$\bigcup_{001}^1$	<code>\bigcup</code>	UNION OPERATOR
U+027D5	$\Join_{001}^1$	<code>\leftouterjoin</code>	LEFT OUTER JOIN
U+027D6	$\Join_{001}^1$	<code>\rightouterjoin</code>	RIGHT OUTER JOIN
U+027D7	$\Join_{001}^1$	<code>\fullouterjoin</code>	FULL OUTER JOIN
U+027D8	$\Uparrow_{001}^1$	<code>\biguparrow</code>	LARGE UP TACK
U+027D9	$\Downarrow_{001}^1$	<code>\bigdownarrow</code>	LARGE DOWN TACK
U+029F8	$\big/_{001}^1$	<code>\xsol</code>	BIG SOLIDUS
U+029F9	$\big\backslash_{001}^1$	<code>\xbsol</code>	BIG REVERSE SOLIDUS
U+02A00	$\bigodot_{001}^1$	<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR

U+02A01		\bigoplus	N-ARY CIRCLED PLUS OPERATOR
U+02A02		\bigotimes	N-ARY CIRCLED TIMES OPERATOR
U+02A03		\bigcupdot	N-ARY UNION OPERATOR WITH DOT
U+02A04		\biguplus	N-ARY UNION OPERATOR WITH PLUS
U+02A05		\bigsqcap	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06		\bigsqcup	N-ARY SQUARE UNION OPERATOR
U+02A07		\conjquant	TWO LOGICAL AND OPERATOR
U+02A08		\disjquant	TWO LOGICAL OR OPERATOR
U+02A09		\bigtimes	N-ARY TIMES OPERATOR
U+02A0B		\sumint	SUMMATION WITH INTEGRAL
U+02A0C		\iiint	QUADRUPLE INTEGRAL OPERATOR
U+02A0D		\intbar	FINITE PART INTEGRAL
U+02A0E		\intBar	INTEGRAL WITH DOUBLE STROKE
U+02A0F		\fint	INTEGRAL AVERAGE WITH SLASH
U+02A10		\cirfnint	CIRCULATION FUNCTION
U+02A11		\awint	ANTICLOCKWISE INTEGRATION LINE INTEGRATION WITH RECTANGULAR
U+02A12		\rppolint	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A13		\scpolint	PATH AROUND POLE LINE INTEGRATION NOT INCLUDING THE
U+02A14		\npolint	POLE
U+02A15		\pointint	INTEGRAL AROUND A POINT OPERATOR
U+02A16		\sqint	QUATERNION INTEGRAL OPERATOR INTEGRAL WITH LEFTWARDS ARROW WITH
U+02A17		\intlarhk	HOOK
U+02A18		\intx	INTEGRAL WITH TIMES SIGN
U+02A19		\intcap	INTEGRAL WITH INTERSECTION
U+02A1A		\intcup	INTEGRAL WITH UNION
U+02A1B		\upint	INTEGRAL WITH OVERBAR
U+02A1C		\lowint	INTEGRAL WITH UNDERBAR
U+02A1D		\Join	JOIN

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

`\um@nolimits` This macro is a sequence containing those maths operators that require a `\nolimits` suffix. This list is used when processing `unicode-math-table.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 20). I've chosen essentially just the operators that look like integrals; hopefully a better mathematician can help me out here. I've a feeling that it's more useful *not* to include the multiple integrals such as , but that might be a matter of preference.

```

555 \def\um@nolimits{
556   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oiint\@elt\oiint
557   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrackwise\@elt\sumint
558   \@elt\intbar\@elt\intBar\@elt\oint\@elt\cirfnint\@elt\awint\@elt\rppoint
559   \@elt\scpolint\@elt\ntpolint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
560   \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint
561 }

```

`\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`.

```

562 \newcommand\addnolimits[1]{
563   \expandafter\def\expandafter\um@nolimits\expandafter{\um@nolimits\@elt#1}
564 }

```

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

```

565 \def\removenolimits#1{
566   \begingroup
567     \def\@elt##1{
568       \ifx##1#1\else
569         \noexpand\@elt\noexpand##1
570       \fi}
571     \xdef\um@nolimits{\um@nolimits}
572   \endgroup
573 }

```



---


$$\iiint_V \iiint_V \iiint_V$$

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

---

## 5.5 Radicals

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

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

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

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

---

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

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

---

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

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

---

## 5.6 Delimiters

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

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

575 `\let\left@primitive\left`

576 `\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`












---

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

---

Here are all `\mathopen` characters:

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

U+02987		<code>\llparenthesis</code>	Z NOTATION LEFT IMAGE BRACKET
U+02989		<code>\llangle</code>	Z NOTATION LEFT BINDING BRACKET
U+0298B		<code>\lbrackubar</code>	LEFT SQUARE BRACKET WITH UNDERBAR
U+0298D		<code>\lbrackultick</code>	CORNER LEFT SQUARE BRACKET WITH TICK IN
U+0298F		<code>\lbracklltick</code>	BOTTOM CORNER
U+02991		<code>\langedot</code>	LEFT ANGLE BRACKET WITH DOT
U+02993		<code>\lparenless</code>	LEFT ARC LESS-THAN BRACKET
U+02997		<code>\lblrbrak</code>	LEFT BLACK TORTOISE SHELL BRACKET
U+029D8		<code>\lvzigzag</code>	LEFT WIGGLY FENCE
U+029DA		<code>\Lvzigzag</code>	LEFT DOUBLE WIGGLY FENCE
U+029FC		<code>\lcurvyangle</code>	LEFT POINTING CURVED ANGLE BRACKET
U+03014		<code>\lbrbrak</code>	LEFT BROKEN BRACKET
U+03018		<code>\Lbrbrak</code>	LEFT WHITE TORTOISE SHELL BRACKET

And `\mathclose`:

USV	Ex.	Macro	Description
U+00029	)	<code>\rparen</code>	RIGHT PARENTHESIS
U+0005D	]	<code>\rbrack</code>	RIGHT SQUARE BRACKET
U+0007D	}	<code>\rbrace</code>	RIGHT CURLY BRACKET DOUBLE ANGLE QUOTATION MARK
U+000BB	»	<code>\guillemotright</code>	(GUILLEMET), RIGHT
U+02019	'	<code>\rq</code>	SINGLE QUOTATION MARK, RIGHT
U+0201B	‘	<code>\quotsinglright</code>	RISING SINGLE QUOTE, RIGHT (HIGH)
U+0201F	“	<code>\quotdblright</code>	RISING DOUBLE QUOTE, RIGHT (HIGH) SINGLE ANGLE QUOTATION MARK
U+0203A	>	<code>\guilsinglright</code>	(GUILLEMET), RIGHT
U+02309	]̂	<code>\rceil</code>	RIGHT CEILING
U+0230B	]̇	<code>\rfloor</code>	RIGHT FLOOR
U+0231D	]̋	<code>\urcorner</code>	UPPER RIGHT CORNER
U+0231F	]̌	<code>\lrcorner</code>	LOWER RIGHT CORNER LIGHT RIGHT TORTOISE SHELL BRACKET
U+02773		<code>\rbrbrak</code>	ORNAMENT
U+027C6	⎵	<code>\rbag</code>	RIGHT S-SHAPED BAG DELIMITER MATHEMATICAL RIGHT WHITE SQUARE
U+027E7	⎶	<code>\rBrack</code>	BRACKET
U+027E9	⋙	<code>\rangle</code>	MATHEMATICAL RIGHT ANGLE BRACKET MATHEMATICAL RIGHT DOUBLE ANGLE
U+027EB	⋛	<code>\rAngle</code>	BRACKET MATHEMATICAL RIGHT WHITE TORTOISE
U+027ED		<code>\Rbrbrak</code>	SHELL BRACKET
U+02984	⎶̇	<code>\rBrace</code>	RIGHT WHITE CURLY BRACKET
U+02986	⎶̈	<code>\rParen</code>	RIGHT WHITE PARENTHESIS

U+02988	⤵	<code>\rrparenthesis</code>	Z NOTATION RIGHT IMAGE BRACKET
U+0298A	⤵	<code>\rrangle</code>	Z NOTATION RIGHT BINDING BRACKET
U+0298C	⌋	<code>\rbrackubar</code>	RIGHT SQUARE BRACKET WITH UNDERBAR
U+0298E	⌋	<code>\rbracklrtick</code>	BOTTOM CORNER RIGHT SQUARE BRACKET WITH TICK IN TOP
U+02990	⌋	<code>\rbrackurtick</code>	CORNER
U+02992	⤵	<code>\rangledot</code>	RIGHT ANGLE BRACKET WITH DOT
U+02994	⤵	<code>\rpangtr</code>	RIGHT ARC GREATER-THAN BRACKET
U+02998	⌋	<code>\rblbrbrak</code>	RIGHT BLACK TORTOISE SHELL BRACKET
U+029D9	⌋	<code>\rvzigzag</code>	RIGHT WIGGLY FENCE
U+029DB	⌋	<code>\Rvzigzag</code>	RIGHT DOUBLE WIGGLY FENCE
U+029FD	⤵	<code>\rcurvyangle</code>	RIGHT POINTING CURVED ANGLE BRACKET
U+03015		<code>\rbrbrak</code>	RIGHT BROKEN BRACKET
U+03019		<code>\Rbrbrak</code>	RIGHT WHITE TORTOISE SHELL BRACKET

## 5.7 Maths accents

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

USV	Ex.	Macro	Description
U+00300	˘	<code>\grave</code>	GRAVE ACCENT
U+00301	ˆ	<code>\acute</code>	ACUTE ACCENT
U+00302	ˆ	<code>\hat</code>	CIRCUMFLEX ACCENT
U+00303	˜	<code>\tilde</code>	TILDE
U+00304	ˉ	<code>\bar</code>	MACRON
U+00305	ˉ	<code>\overbar</code>	OVERBAR EMBELLISHMENT
U+00306	˘	<code>\breve</code>	BREVE
U+00307	˙	<code>\dot</code>	DOT ABOVE
U+00308	¨	<code>\ddot</code>	DIERESIS
U+00309	ˆ	<code>\ovhook</code>	COMBINING HOOK ABOVE
U+0030A	ˆ	<code>\ocirc</code>	RING
U+0030C	ˇ	<code>\check</code>	CARON
U+00310	ˆ	<code>\candra</code>	CANDRABINDU (NON-SPACING)
U+00312	ˆ	<code>\turnedcomma</code>	COMBINING TURNED COMMA ABOVE GREEK PSILI (SMOOTH BREATHING)
U+00313	ˆ	<code>\osmooth</code>	(NON-SPACING) GREEK DASIA (ROUGH BREATHING)
U+00314	ˆ	<code>\orough</code>	(NON-SPACING)
U+00315	ˆ	<code>\ocommatopright</code>	COMBINING COMMA ABOVE RIGHT
U+0031A	ˆ	<code>\droang</code>	LEFT ANGLE ABOVE (NON-SPACING)
U+020D0	ˆ	<code>\leftharpoonaccent</code>	COMBINING LEFT HARPOON ABOVE
U+020D1	ˆ	<code>\rightharpoonaccent</code>	COMBINING RIGHT HARPOON ABOVE

U+020D2		<code>\vertoverlay</code>	COMBINING LONG VERTICAL LINE OVERLAY
U+020D6		<code>\overleftarrow</code>	COMBINING LEFT ARROW ABOVE
U+020D7		<code>\vec</code>	COMBINING RIGHT ARROW ABOVE
U+020DB		<code>\dddot</code>	COMBINING THREE DOTS ABOVE
U+020DC		<code>\ddddot</code>	COMBINING FOUR DOTS ABOVE
U+020E1		<code>\overleftrightharpoon</code>	COMBINING LEFT RIGHT ARROW ABOVE
U+020E7		<code>\annuity</code>	COMBINING ANNUITY SYMBOL
U+020E8		<code>\threeunderdot</code>	COMBINING TRIPLE UNDERDOT
U+020E9		<code>\widebridgeabove</code>	COMBINING WIDE BRIDGE ABOVE
U+020EC		<code>\underrightharpoondown</code>	BARB DOWNWARDS COMBINING LEFTWARDS HARPOON WITH
U+020ED		<code>\underleftharpoondown</code>	BARB DOWNWARDS
U+020EE		<code>\underleftarrow</code>	COMBINING LEFT ARROW BELOW
U+020EF		<code>\underrightarrow</code>	COMBINING RIGHT ARROW BELOW
U+020F0		<code>\asteraccent</code>	COMBINING ASTERISK ABOVE

---

## 6 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.

```

577 \newcommand\um@zf@feature[2]{
578   \define@key[zf]{options}{#1}[]{
579     \ifum@fontspec@feature
580       #2
581     \else
582       \PackageError{fontspec/unicode-math}
583         {The ‘#1’ font feature can only be used for maths fonts}
584         {The feature you tried to use can only be in commands
585          like \protect\setmathfont}
586     \fi
587   }
588 }
```

### 6.1 OpenType maths font features

```

589 \um@zf@feature{ScriptStyle}{
590   \zf@update@ff{+ssty=0}
591 }
592 \um@zf@feature{ScriptScriptStyle}{
593   \zf@update@ff{+ssty=1}
594 }
```

## 6.2 Script and scriptscript font options

```

595 \define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
596 \define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
597 \define@cmdkey[um]{options}[um@]{ScriptFont}{}
598 \define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}

```

## 6.3 Range processing

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

```

599 \define@choicekey+[um]{options}{Range}[\@tempa\@tempb]{ALL}{
600   \ifcase\@tempb\relax
601     \global\let\um@char@range\empty
602   \fi
603 }{
604   \xdef\um@char@range{#1}
605 }

```

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 10.

Table 10: 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:

```

606 \newcommand\um@parse@term[4]{
607   \clist_map_variable:NNn \um@char@range \@ii {
608     \unless\ifx\@ii\empty
609       \@tempswafalse

```

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

```

610 \expandafter\um@firstchar\expandafter{\@ii}
611 \ifx\@tempa\um@backslash
612 \expandafter\ifx\@ii#2\relax
613 \@tempswatrue
614 \else
615 \expandafter\ifx\@ii#3\relax
616 \@tempswatrue
617 \fi
618 \fi

```

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

```

619 \else
620 \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
621 \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.

```

622 \if@tempswa
623 \ifx\um@char@num@range\empty
624 \g@addto@macro\um@char@num@range{#1}
625 \else
626 \g@addto@macro\um@char@num@range{,#1}
627 \fi
628 #4%
629 \fi
630 \fi
631 }
632 }
633 \def\um@firstof#1#2\@nil{#1}
634 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
635 \def\um@firstchar#1{\edef\@tempa{\expandafter\um@firstof\string#1\@nil}}

```

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

```

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

```

`\um@parse@range` Weird syntax. As shown previously in table 10, this macro can be passed four different input types via `\um@parse@term`.

```

636 \def\um@parse@range#1-#2-#3\@nil#4\@nil{
637 \def\@tempa{#1}
638 \def\@tempb{#2}

```

---

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

---

```

639 \expandafter\ifx\expandafter\@marker\@tempb\relax
640 \ifnum#4=#1\relax
641 \@tempwattrue
642 \fi
643 \else

```

---

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

---

```

644 \ifx\@empty\@tempb
645 \ifnum#4>\numexpr#1-1\relax
646 \@tempwattrue
647 \fi
648 \else

```

---

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

---

```

649 \ifx\@empty\@tempa
650 \ifnum#4<\numexpr#2+1\relax
651 \@tempwattrue
652 \fi

```

---

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

---

```

653 \else
654 \ifnum#4>\numexpr#1-1\relax
655 \ifnum#4<\numexpr#2+1\relax
656 \@tempwattrue
657 \fi
658 \fi
659 \fi
660 \fi
661 \fi
662 }

```

$\backslash um\_map\_char:nn$      $\#1$  : Number of iterations  
                            $\#2$  : Starting input char(s)  
                            $\#3$  : Starting output char



Loops through character ranges setting \mathcode.

```

663 \cs_set:Nn \um_map_chars_range:nnn {
664   \clist_map_variable:nNn {#2} \l_um_input_num {
665     \prg_stepwise_variable:nnnNn{0}{1}{#1} \l_um_incr_num {
666       \um_set_mathcode:nnnn
667       {\numexpr \l_um_incr_num+ \l_um_input_num \relax}
668       {\mathalpha}{\um_symfont_tl}
669       {\numexpr \l_um_incr_num + #3 \relax}
670     }
671   }
672 }
673 \cs_set:Nn \um_map_chars_latin:nn {
674   \um_map_chars_range:nnn {25}{#1}{#2}
675 }
676 \cs_set:Nn \um_map_chars_greek:nn {
677   \um_map_chars_range:nnn {24}{#1}{#2}
678 }
679 \cs_set:Nn \um_map_chars_numbers:nn {
680   \um_map_chars_range:nnn {9}{#1}{#2}
681 }
682 \cs_set:Nn \um_map_char:nn {
683   \um_map_chars_range:nnn {0}{#1}{#2}
684 }

```

\um\_set\_mathalphabet\_char:Nnnn #1 : Maths alphabet  
 #2 : Input char(s)  
 #3 : Output char

Loops through character ranges setting \mathcode.

```

685 \cs_set:Npn \exp_args:Nnff {\::n\::f\::f\:::}
686 \cs_new:Nn \um_set_mathalphabet_char:Nnn {
687   \clist_map_variable:nNn {#2} \l_um_input_num {
688     \exp_args:Nnff \um_mathmap:Nnn {#1}
689     {\number\numexpr\l_um_input_num\relax} {\number\numexpr#3\relax}
690   }
691 }

```

\um\_set\_mathalph\_range:Nnn [*(Number of iterations)*] #1 : Maths alphabet  
 #2 : Starting input char(s)  
 #3 : Starting output char

Loops through character ranges setting \mathcode.

```

692 \cs_new:Nn \um_set_mathalph_range:nnnn {
693   \clist_map_variable:nNn {#3} \l_um_input_num {
694     \prg_stepwise_variable:nnnNn {0}{1}{#1} \l_um_inc_num {
695       \exp_args:Nnff \um_mathmap:Nnn {#2}
696       {\number\numexpr \l_um_inc_num + \l_um_input_num \relax}
697       {\number\numexpr \l_um_inc_num + #4 \relax}

```

```

698     }
699   }
700 }
701 \cs_new:Nn \um_set_mathalphabet_numbers:Nnn {
702   \um_set_mathalph_range:nNnn {9}{#1}{#2}{#3}
703 }
704 \cs_new:Nn \um_set_mathalphabet_latin:Nnn {
705   \um_set_mathalph_range:nNnn {25}{#1}{#2}{#3}
706 }
707 \cs_new:Nn \um_set_mathalphabet_greek:Nnn {
708   \um_set_mathalph_range:nNnn {24}{#1}{#2}{#3}
709 }

```

---

**BCDBCDEABCDEF**

```

\ExplSyntaxOn
{\um_map_chars_range:nnn{3}{'\A,'\D}{'\B}
$ABCEFG$} $ABCEFG$

```

---

## 6.4 Resolving Greek symbol name control sequences

`\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.

```

710 \AtBeginDocument{\um@resolve@greek}
711 \newcommand\um@resolve@greek{
712   \def\Alpha{\mitAlpha}
713   \def\Beta{\mitBeta}
714   \def\Gamma{\mitGamma}
715   \def\Delta{\mitDelta}
716   \def\Epsilon{\mitEpsilon}
717   \def\Zeta{\mitZeta}
718   \def\Eta{\mitEta}
719   \def\Theta{\mitTheta}
720   \def\Iota{\mitIota}
721   \def\Kappa{\mitKappa}
722   \def\Lambda{\mitLambda}
723   \def\Mu{\mitMu}
724   \def\Nu{\mitNu}
725   \def\Xi{\mitXi}
726   \def\Omicron{\mitOmicron}
727   \def\Pi{\mitPi}
728   \def\Rho{\mitRho}
729   \def\varTheta{\mitvarTheta}
730   \def\Sigma{\mitSigma}

```

```

731 \def\Tau{\mitTau}
732 \def\Upsilon{\mitUpsilon}
733 \def\Phi{\mitPhi}
734 \def\Chi{\mitChi}
735 \def\Psi{\mitPsi}
736 \def\Omega{\mitOmega}

Lowercase:
737 \def\alpha{\mitalpha}
738 \def\beta{\mitbeta}
739 \def\gamma{\mitgamma}
740 \def\delta{\mitdelta}
741 \def\epsilon{
742   \bool_if:NTF \g_um_texgreek_bool {\mitvarepsilon}{\mitepsilon}
743 }
744 \def\zeta{\mitzeta}
745 \def\eta{\miteta}
746 \def\theta{\mittheta}
747 \def\iota{\mitiota}
748 \def\kappa{\mitkappa}
749 \def\lambda{\mitlambda}
750 \def\mu{\mitmu}
751 \def\nu{\mitnu}
752 \def\xi{\mitxi}
753 \def\omicron{\mitomicron}
754 \def\pi{\mitpi}
755 \def\rho{\mitrho}
756 \def\varsigma{\mitvarsigma}
757 \def\sigma{\mitsigma}
758 \def\tau{\mittau}
759 \def\upsilon{\mitupsilon}
760 \def\phi{
761   \bool_if:NTF \g_um_texgreek_bool {\mitvarphi}{\mitphi}
762 }
763 \def\chi{\mitchi}
764 \def\psi{\mitpsi}
765 \def\omega{\mitomega}
766 \def\varepsilon{
767   \bool_if:NTF \g_um_texgreek_bool {\mitepsilon}{\mitvarepsilon}
768 }
769 \def\vartheta{\mitvartheta}
770 \def\varkappa{\mitvarkappa}
771 \def\varphi{
772   \bool_if:NTF \g_um_texgreek_bool {\mitphi}{\mitvarphi}
773 }
774 \def\varrho{\mitvarrho}
775 \def\varpi{\mitvarpi}

```

776 }

## 6.5 Setting up the mappings

`\um_setup_literals:` : TODO : other literal symbols

```
777 \cs_set:Nn \um_setup_literals: {
778   \um_map_chars_latin:nn {\um@usv@upLatin}{\um@usv@upLatin}
779   \um_map_chars_latin:nn {\um@usv@itLatin}{\um@usv@itLatin}
780   \um_map_chars_latin:nn {\um@usv@itlatin}{\um@usv@itlatin}
781   \um_map_char:nn {\um@usv@ith}{\um@usv@ith}
782   \um_map_chars_latin:nn {\um@usv@uplatin}{\um@usv@uplatin}
783   \um_map_chars_greek:nn {\um@usv@upGreek}{\um@usv@upGreek}
784   \um_map_char:nn {\um@usv@varTheta}{\um@usv@varTheta}
785   \um_map_chars_greek:nn {\um@usv@itGreek}{\um@usv@itGreek}
786   \um_map_chars_greek:nn {\um@usv@upgreek}{\um@usv@upgreek}
787 }
```

`\um_setup_bf_literals:` TODO: other literal symbols

```
788 \cs_set:Nn \um_setup_bf_literals: {
789   \um_map_chars_latin:nn {\um@usv@bfupLatin}{\um@usv@bfupLatin}
790   \um_map_chars_latin:nn {\um@usv@bfuplatin}{\um@usv@bfuplatin}
791   \um_map_chars_latin:nn {\um@usv@bfitLatin}{\um@usv@bfitLatin}
792   \um_map_chars_latin:nn {\um@usv@bfitlatin}{\um@usv@bfitlatin}
793   \um_map_chars_greek:nn {\um@usv@bfupGreek}{\um@usv@bfupGreek}
794   \um_map_chars_greek:nn {\um@usv@bfupgreek}{\um@usv@bfupgreek}
795   \um_map_chars_greek:nn {\um@usv@bfitGreek}{\um@usv@bfitGreek}
796   \um_map_chars_greek:nn {\um@usv@bfitgreek}{\um@usv@bfitgreek}
797 }
```

`\um_setup_Latin:`

```
798 \cs_set:Nn \um_setup_Latin: {
799   \if@um@upLatin
800     \um_map_chars_latin:nn {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
801   \else
802     \um_map_chars_latin:nn {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
803   \fi
804 }
```

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

```
805 \cs_set:Nn \um_setup_latin: {
806   \if@um@uplatin
807     \um_map_chars_latin:nn {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
808     \um_map_char:nn {\um@usv@ith}{'\h}
809   \else
810     \um_map_chars_latin:nn {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
```

```

811     \um_map_char:nn {\h,\um@usv@ith}{\um@usv@ith}
812   \fi
813 }

\um_setup_Greek:

814 \cs_set:Nn \um_setup_Greek: {
815   \if@um@upGreek
816     \um_map_chars_greek:nn {\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
817     \um_map_char:nn {\um@usv@varTheta,"1D6F3}{\um@usv@varTheta}
818   \else
819     \um_map_chars_greek:nn {\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
820     \um_map_char:nn {\um@usv@varTheta}{\um@usv@itvarTheta}
821   \fi
822 }

\um_setup_greek:

823 \cs_set:Nn \um_setup_greek: {
824   \if@um@upgreek
825     \um_map_chars_greek:nn {\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
826     \um_map_char:nn {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
827     \um_map_char:nn {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
828     \um_map_char:nn {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
829     \um_map_char:nn {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
830     \um_map_char:nn {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
831     \um_map_char:nn {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
832   \else
833     \um_map_chars_greek:nn {\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
834     \um_map_char:nn {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
835     \um_map_char:nn {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
836     \um_map_char:nn {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
837     \um_map_char:nn {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
838     \um_map_char:nn {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
839     \um_map_char:nn {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
840   \fi
841 }

```

## 7 Maths alphabets mapping definitions

Algorithm for setting alphabet fonts. By default, when Range is empty, we are in *implicit* mode. If Range contains the name of the math alphabet, we are in *explicit* mode and do things slightly differently.

Implicit mode:

- Try and set all of the alphabet shapes.

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

Explicit mode:

- Only set the alphabets specified.
- Check for the first glyph of the uppercase Latin alphabet to detect if the font contains the alphabet shape in the unicode math plane.
- For unicode math alphabets, overwrite whatever's already there.
- Otherwise, use the ASCII letters instead.

```

842 \cs_new:Nn \um_setup_alphabets: {
843   \um_setup_math_alphabet:n {up   }
844   \um_setup_math_alphabet:n {it   }
845   \um_setup_math_alphabet:n {bb   }
846   \um_setup_math_alphabet:n {scr  }
847   \um_setup_math_alphabet:n {frak }
848   \um_setup_math_alphabet:n {sf   }
849   \um_setup_math_alphabet:n {sfup }
850   \um_setup_math_alphabet:n {sfit }
851   \um_setup_math_alphabet:n {tt   }
852   \um_setup_math_alphabet:n {bf   }
853   \um_setup_math_alphabet:n {bfup }
854   \um_setup_math_alphabet:n {bfit }
855   \um_setup_math_alphabet:n {bfscr }
856   \um_setup_math_alphabet:n {bffrak}
857   \um_setup_math_alphabet:n {bfsf }
858   \um_setup_math_alphabet:n {bfsfup}
859   \um_setup_math_alphabet:n {bfsfit}
860 }

861 \cs_new:Nn \um_setup_math_alphabet:n {
862   \um_glyph_if_exist:nTF {\csname um@usv@#1Latin \endcsname}{
863     \um_maybe_init_alphabet:n {#1}
864     \um_prepare_alph:n {#1}
865     \use:c {um_config_math#1:}
866   }{
867     \PackageWarningNoLine{unicode-math}{^^J\space\space\space\space
868       Math~ alphabet~ \@backslashchar math#1~ not~ found~ in~ font~ \font-
name\um@font}

```

```

869     \cs_if_exist:cT {um_fix_math#1:} {
870         \use:c {um_fix_math#1:}
871     }
872 }
873 }
874 \cs_set:Nn \um_fix_mathtt: {
875     \SetMathAlphabet\mathtt{normal}\encodingdefault\ttdefault\mddefault\updefault
876 }
877 \cs_set:Nn \um_init_alphabet:n {
878     \cs_set_eq:cN {um_setup_math#1:} \prg_do_nothing:
879 }

```

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

```

880 \prg_new_conditional:Nnn \um_glyph_if_exist:n {p,TF,T,F} {
881     \tex_iffontchar:D \um@font #1 \scan_stop: \prg_return_true: \else: \prg_return_false: \fi:
882 }

```

`\um_prepare_alph:n` If `\mathXY` hasn't been (re-)declared yet, then define it in terms of unicode-math definitions. Use `\bgroup/\egroup` so s'scripts scan the whole thing.

```

883 \cs_new:Nn \um_prepare_alph:n {
884     \cs_if_exist:cF {um_math#1:n} {
885         \cs_set:cpn {um_math#1:n} ##1 {
886             \use:c {um_setup_math#1:} ##1 \egroup
887         }
888         \cs_set_protected:cpn {math#1} {
889             \bgroup
890             \mode_if_math:F {
891                 \egroup\expandafter
892                 \non@alpherr\expandafter{\csname math#1\endcsname\space}
893             }
894             \use:c {um_math#1:n}
895         }
896     }
897 }

```

: TODO : nested alphabets?

## 7.1 Non-bold math alphabets

### 7.1.1 Upright: `\mathup`

---

ABCDEFGHIJKLMNOPQRSTUVWXYZ	<code>\mathup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
abcdefghijklmnopqrstuvwxyz	<code>\mathup{abcdefghijklmnopqrstuvwxyz}\$ \</code>
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ	<code>\mathup{ }\$ \quad\mathup{ }\$ \</code>
αβγδεζηθικλμνξοπρστυφχψω εθκφρτ	<code>\mathup{ }\$ \quad\mathup{ }\$ \</code>

---

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

```
898 \cs_new:Npn \um_config_mathup: {
899   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
900   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
901   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
902   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
903   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}
904   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}
905   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}
906   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
907   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
908   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
909   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
910   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
911   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
912 }
```

### 7.1.2 Italic: `\mathit`

---

ABCDEFGHIJKLMNOPQRSTUVWXYZ	<code>\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \</code>
abcdefghijklmnopqrstuvwxyz	<code>\mathit{abcdefghijklmnopqrstuvwxyz}\$ \</code>
ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ	<code>\mathit{ }\$ \quad\mathit{ }\$ \</code>
αβγδεζηθικλμνξοπρστυφχψω εθκφρτ	<code>\mathit{ }\$ \quad\mathit{ }\$ \</code>

---

Roman:

```
913 \cs_new:Npn \um_config_mathit: {
914   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
915   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
916   \um_set_mathalphabet_char:Nnn{\mathit}{`h,\um@usv@ith}{\um@usv@ith}
```

Greek:

```
917   \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
918   \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
919   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}
```



```

920 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}
921 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarThe
922 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvar
923 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvarthe
924 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkap
925 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
926 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
927 \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
928 }

```

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

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

Numbers:

```

929 \cs_new:Npn \um_config_mathbb: {
930   \um_set_mathalphabet_numbers:Nnn{\mathbb}{\um@usv@num}{\um@usv@bbnum}

```

Roman uppercase:

```

931 \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bbLatin}
932 \um_set_mathalphabet_char:Nnn{\mathbb}{`\C,"1D60A}{ "2102}
933 \um_set_mathalphabet_char:Nnn{\mathbb}{`\H,"1D60F}{ "210D}
934 \um_set_mathalphabet_char:Nnn{\mathbb}{`\N,"1D60F}{ "2115}
935 \um_set_mathalphabet_char:Nnn{\mathbb}{`\P,"1D617}{ "2119}
936 \um_set_mathalphabet_char:Nnn{\mathbb}{`\Q,"1D618}{ "211A}
937 \um_set_mathalphabet_char:Nnn{\mathbb}{`\R,"1D619}{ "211D}
938 \um_set_mathalphabet_char:Nnn{\mathbb}{`\Z,"1D621}{ "2124}

```

Roman lowercase:

```

939 \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bblatin}
940 }

```

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

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

```

941 \cs_new:Npn \um_config_mathscr: {
942   \um_set_mathalphabet_latin:Nnn{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}
943   \um_set_mathalphabet_char:Nnn{\mathscr}{`\B,"1D435}{ "212C}
944   \um_set_mathalphabet_char:Nnn{\mathscr}{`\E,"1D438}{ "2130}
945   \um_set_mathalphabet_char:Nnn{\mathscr}{`\F,"1D439}{ "2131}

```

```

946 \um_set_mathalphabet_char:Nnn{\mathscr}{`\H,"1D43B}{ "210B}
947 \um_set_mathalphabet_char:Nnn{\mathscr}{`\I,"1D43C}{ "2110}
948 \um_set_mathalphabet_char:Nnn{\mathscr}{`\L,"1D43F}{ "2112}
949 \um_set_mathalphabet_char:Nnn{\mathscr}{`\M,"1D440}{ "2133}
950 \um_set_mathalphabet_char:Nnn{\mathscr}{`\R,"1D445}{ "211B}
951 \um_set_mathalphabet_latin:Nnn{\mathscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@scrlatin}
952 \um_set_mathalphabet_char:Nnn{\mathscr}{`\e,"1D452}{ "212F}
953 \um_set_mathalphabet_char:Nnn{\mathscr}{`\g,"1D454}{ "210A}
954 \um_set_mathalphabet_char:Nnn{\mathscr}{`\o,"1D45C}{ "2134}
955 }

```

### 7.1.5 Fraktur or fraktur or blackletter: \mathfrak

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$   
 $\mathfrak{abcdefghijklmnopqrstuvwxyz}$

$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$  \\
 $\mathfrak{abcdefghijklmnopqrstuvwxyz}$  \\

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

```

956 \cs_new:Npn \um_config_mathfrak: {
957 \um_set_mathalphabet_latin:Nnn{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLatin}
958 \um_set_mathalphabet_char:Nnn{\mathfrak}{`\C,"1D436}{ "212D}
959 \um_set_mathalphabet_char:Nnn{\mathfrak}{`\H,"1D43B}{ "210C}
960 \um_set_mathalphabet_char:Nnn{\mathfrak}{`\I,"1D43C}{ "2111}
961 \um_set_mathalphabet_char:Nnn{\mathfrak}{`\R,"1D445}{ "211C}
962 \um_set_mathalphabet_char:Nnn{\mathfrak}{`\Z,"1D44D}{ "2128}
963 \um_set_mathalphabet_latin:Nnn{\mathfrak}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@fraklatin}
964 }

```

### 7.1.6 Sans serif: \mathsf

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

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

```

965 \cs_new:Npn \um_config_mathsf: {
966 \bool_if:NTF \g_um_sf literal_bool {
967 \um_set_mathalphabet_numbers:Nnn{\mathsf}{\um@usv@num}{\um@usv@sfnun}
968 \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@upLatin}{\um@usv@sfulatin}
969 \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@uplatin}{\um@usv@sfulatin}
970 \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@itLatin}{\um@usv@sfitLatin}
971 \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@itlatin}{\um@usv@sfitlatin}
972 }{
973 \bool_if:NTF \g_um_upsans_bool {
974 \um_set_mathalphabet_numbers:Nnn \mathsf {\um@usv@num}{\um@usv@sfnun}

```

```

975 \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfupLati
976 \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfuplati
977 }{
978 \um_set_mathalphabet_numbers:Nnn \mathsf {\um@usv@num}{\um@usv@sfnum}
979 \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLati
980 \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfitlati
981 }
982 }
983 }

```

### 7.1.7 Sans serif upright: \mathsfup

---

0123456789	$\mathsfup{0123456789}$
ABCDEFGHIJKLMNOPQRSTUVWXYZ	$\mathsfup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
abcdefghijklmnopqrstuvwxyz	$\mathsfup{abcdefghijklmnopqrstuvwxyz}$

---

```

984 \cs_new:Npn \um_config_mathsfup: {
985   \um_set_mathalphabet_numbers:Nnn{\mathsfup}{\um@usv@num}{\um@usv@sfnum}
986   \um_set_mathalphabet_latin:Nnn{\mathsfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfupLatin}
987   \um_set_mathalphabet_latin:Nnn{\mathsfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfuplatin}
988 }

```

### 7.1.8 Sans serif italic: \mathsfit

---

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

---

```

989 \cs_new:Npn \um_config_mathsfit: {
990   \um_set_mathalphabet_numbers:Nnn{\mathsfit}{\um@usv@num}{\um@usv@sfnum}
991   \um_set_mathalphabet_latin:Nnn{\mathsfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLatin}
992   \um_set_mathalphabet_latin:Nnn{\mathsfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfitlatin}
993 }

```

### 7.1.9 Typewriter or monospaced: \mathtt

---

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

---

```

994 \cs_new:Npn \um_config_mathtt: {

```

```

995 \um_set_mathalphabet_numbers:Nnn{\mathtt}{\um@usv@enum}{\um@usv@ttnum}
996 \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}
997 \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}
998 }

```

## 7.2 Bold math alphabets

### 7.2.1 Bold: `\mathbf`

0123456789  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ  
Θ Ϳ  
αβγδεζηθικλμνξοπρστυφχψω  
€θκφρωͿ

[illegible]

```

999 \cs_new:Npn \um_config_mathbf: {
1000   \um_set_mathalphabet_numbers:Nnn{\mathbf}{\um@usv@enum}{\um@usv@bfnun}
1001   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@1D7CA}
1002   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@digamma}{\um@usv@1D7CB}
1003   \ifum@bfliteral
1004     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin}{\um@usv@bfulatin}
1005     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@itLatin}{\um@usv@bfitalatin}
1006     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin}{\um@usv@bfuplatin}
1007     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@eitlatin}{\um@usv@bfitalatin}
1008     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek}{\um@usv@bfulatin}
1009     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itGreek}{\um@usv@bfitalatin}
1010     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek}{\um@usv@bfupgreek}
1011     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itgreek}{\um@usv@bfitalatin}
1012     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
1013     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}
1014     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}
1015     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@bfDigamma}
1016     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial}{\um@usv@bfpartial}
1017     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}
1018     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}
1019     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}
1020     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}
1021     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}
1022     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}
1023     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@digamma}{\um@usv@bfdigamma}
1024     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitalvarTheta}
1025     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itNabla}{\um@usv@bfitalNabla}

```

```

1026 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}
1027 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
1028 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}
1029 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
1030 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}
1031 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}
1032 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}
1033 \else
1034 \ifum@bfupLatin
1035 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfupLatin}
1036 \else
1037 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}
1038 \fi
1039 \ifum@bfuplatin
1040 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfuplatin}
1041 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfuph}
1042 \else
1043 \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
1044 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
1045 \fi
1046 \ifum@bfupGreek
1047 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfupGreek}
1048 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvarT}
1049 \else
1050 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}
1051 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitva}
1052 \fi
1053 \ifum@bfupgreek
1054 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfupgreek}
1055 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bf}
1056 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfvarT}
1057 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfvark}
1058 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
1059 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
1060 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
1061 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfpartia}
1062 \else
1063 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}
1064 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bf}
1065 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitva}
1066 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitva}
1067 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi}
1068 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}
1069 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
1070 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpart}
1071 \fi

```

```

1072 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla,\um@usv@itNabla}{\um@bfNabla_up_or_it_
1073 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um@bfpartial_up_or_it_
1074 \fi
1075 }

```

## 7.2.2 Bold Italic: \mathbfit

---

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

---

```

1076 \cs_new:Npn \um_config_mathbfit: {
1077   \um_set_mathalphabet_numbers:Nnn{\mathbfit}{\um@usv@num}{\um@usv@bfnun}
1078   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLatin}
1079   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlatin}
1080   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGreek}
1081   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgreek}
1082   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@bfupLatin}{\um@usv@bfitLatin}
1083   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@bfuplatin}{\um@usv@bfitlatin}
1084   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@bfupGreek}{\um@usv@bfitGreek}
1085   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@bfupgreek}{\um@usv@bfitgreek}
1086   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
1087   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfitNabla}
1088   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpartial}
1089   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
1090   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfitvartheta}
1091   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
1092   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarphi}
1093   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarrho}
1094   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
1095 }

```

## 7.2.3 Bold Italic: \mathbfup

---

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

---

```

1096 \cs_new:Npn \um_config_mathbfup: {
1097   \um_set_mathalphabet_numbers:Nnn{\mathbfup}{\um@usv@num}{\um@usv@bfnum}
1098   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfupLatin}
1099   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfuplatin}
1100   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfupGreek}
1101   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfupgreek}
1102   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@bfupLatin}{\um@usv@bfupLatin}
1103   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@bfuplatin}{\um@usv@bfuplatin}
1104   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@bfupGreek}{\um@usv@bfupGreek}
1105   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@bfupgreek}{\um@usv@bfupgreek}
1106   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfvarTheta}
1107   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfNabla}
1108   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfpartial}
1109   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfvarepsilon}
1110   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfvartheta}
1111   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfvarkappa}
1112   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
1113   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
1114   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
1115 }

```

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

***ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz***

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

```

1116 \cs_new:Npn \um_config_mathbffrak: {
1117   \um_set_mathalphabet_numbers:Nnn{\mathbffrak}{\um@usv@num}{\um@usv@bfnum}
1118   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@upLatin,\um@usv@itLatin,\um@usv@frakLatin}
1119   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@uplatin,\um@usv@itlatin,\um@usv@fraklatin}
1120 }

```

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

***ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz***

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

```

1121 \cs_new:Npn \um_config_mathbfscr: {
1122   \um_set_mathalphabet_numbers:Nnn{\mathbfscr}{\um@usv@num}{\um@usv@bfnum}
1123   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfscrLatin}
1124   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfscrLatin}
1125 }

```

## 7.2.6 Bold sans serif: `\mathbfsf`

0123456789  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz  
 ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ  
 αβγδεζηθικλμνξοπρστυφχψω εδκφρω

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

These use the sans-style settings rather than bold-style.  
 Numbers (always upright) and letters:

```
1126 \cs_new:Npn \um_config_mathbfsf: {
1127   \bool_if:NTF \g_um_sfliteral_bool {
1128     \um_set_mathalphabet_numbers:Nnn \mathbfsf {\um@usv@enum}{\um@usv@bfsfnum}
1129     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin}{\um@usv@bfsfupLatin}
1130     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin}{\um@usv@bfsfupLatin}
1131     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@itLatin}{\um@usv@bfsfitLatin}
1132     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@itLatin}{\um@usv@bfsfitLatin}
1133     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek}{\um@usv@bfsfupGreek}
1134     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek}{\um@usv@bfsfupGreek}
1135     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@itGreek}{\um@usv@bfsfitGreek}
1136     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@itGreek}{\um@usv@bfsfitGreek}
1137     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta}{1D767}
1138     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@Nabla}{1D76F}
1139     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@partial}{1D789}
1140     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon}{1D78A}
1141     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta}{1D78B}
1142     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa}{1D78C}
1143     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi}{1D78D}
1144     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho}{1D78E}
1145     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi}{1D78F}
1146     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta}{1D7A1}
1147     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itNabla}{\um@usv@bfsfitNabla}
1148     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itpartial}{\um@usv@bfsfitpartial}
1149     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarepsilon}{1D7C4}
1150     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvartheta}{1D7C5}
1151     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarkappa}{1D7C6}
1152     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarphi}{1D7C7}
1153     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarrho}{1D7C8}
1154     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarpi}{1D7C9}
1155   }{
1156     \bool_if:NTF \g_um_upsans_bool {
1157       \um_set_mathalphabet_numbers:Nnn \mathbfsf {\um@usv@enum}{\um@usv@bfsfnum}
1158       \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfup}
1159       \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfup}
```



```

1160 \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfup
1161 \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfup
1162 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfsfup
1163 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfsfup
1164 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@partial,\um@usv@itpartial}{\um@usv@bfsfup
1165 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfsfup
1166 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfsfup
1167 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfsfup
1168 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfsfup
1169 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfsfup
1170 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfsfup
1171 }{
1172 \um_set_mathalphabet_numbers:Nnn \mathbfsf {\um@usv@enum}{\um@usv@bfsfnum}
1173 \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfup
1174 \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfup
1175 \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfup
1176 \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfup
1177 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta}{\um@usv@bfsfup
1178 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfsfup
1179 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@partial,\um@usv@itpartial}{\um@usv@bfsfup
1180 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfsfup
1181 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfsfup
1182 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfsfup
1183 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfsfup
1184 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfsfup
1185 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfsfup }
1186 }
1187 }

```

### 7.2.7 Bold upright sans serif: `\mathbfsfup`

<p style="text-align: center;">0123456789</p> <p style="text-align: center;">ABCDEFGHIJKLMNOPQRSTUVWXYZ</p> <p style="text-align: center;">abcdefghijklmnopqrstuvwxyz</p> <p style="text-align: center;">ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ Θ</p> <p style="text-align: center;">αβγδεζηθικλμνξοπρστυφχψω εδκφρτ</p>	<pre> \setmathfont{STIXGeneral-Bold} \$\mathbfsfup{0123456789}\$ \ \$\mathbfsfup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}\$ \ \$\mathbfsfup{abcdefghijklmnopqrstuvwxyz}\$ \ \$\mathbfsfup{ } \$\quad \$\mathbfsfup{ } \$\quad \$\mathbfsfup{ } \$\quad </pre>
--	--

Numbers (always upright) and letters:

```

1188 \cs_new:Npn \um_config_mathbfsfup: {
1189 \um_set_mathalphabet_numbers:Nnn{\mathbfsfup}{\um@usv@enum}{\um@usv@bfsfnum}
1190 \um_set_mathalphabet_latin:Nnn{\mathbfsfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfupL
1191 \um_set_mathalphabet_latin:Nnn{\mathbfsfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfupl
1192 \um_set_mathalphabet_greek:Nnn{\mathbfsfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfupG
1193 \um_set_mathalphabet_greek:Nnn{\mathbfsfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfupg

```

```

1194 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varTheta,\um@usv@itvarTheta}{\ "1D767}
1195 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@Nabla,\um@usv@itNabla}{\ "1D76F}
1196 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@partial,\um@usv@itpartial}{\ "1D789}
1197 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\ "1D78A}
1198 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@vartheta,\um@usv@itvartheta}{\ "1D78B}
1199 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varkappa,\um@usv@itvarkappa}{\ "1D78C}
1200 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varphi,\um@usv@itvarphi}{\ "1D78D}
1201 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varrho,\um@usv@itvarrho}{\ "1D78E}
1202 \um_set_mathalphabet_char:Nnn{\mathbfsup}{\um@usv@varpi,\um@usv@itvarpi}{\ "1D78F}
1203 }

```

```

\setmathfont{STIXGeneral-BoldItalic}
$ \mathbfsfit{0123456789}$ \\\
$ \mathbfsfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\\
$ \mathbfsfit{abcdefghijklmnopqrstuvwxyz}$ \\\
$ \mathbfsfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\\quad
$ \mathbfsfit{ }$ \\\
$ \mathbfsfit{ }$ \\\quad
$ \mathbfsfit{ }$ \\\

```

*$\mathbf{CDEFGHIJKLMNOPQRSTUVWXYZ}$   
 $\mathbf{abcdefghijklmnopqrstuvwxyz}$   
 $\mathbf{\Lambda EZH\Theta IK\Lambda MN\Xi OP\rho\sigma\tau\upsilon\phi\chi\psi\Omega\Theta}$   
 $\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\omicron\rho\sigma\tau\upsilon\phi\chi\psi\omega\epsilon\delta\kappa\phi\rho\omega$*

Other symbols:

```

1210 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varTheta}{\textit{D7A1}}
1211 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfsf{it}Nabla}
1212 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfsf{it}partial}
1213 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\textit{D7C4}}
1214 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@vartheta,\um@usv@itvartheta}{\textit{D7C5}}
1215 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varkappa,\um@usv@itvarkappa}{\textit{D7C6}}
1216 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varphi,\um@usv@itvarphi}{\textit{D7C7}}
1217 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varrho,\um@usv@itvarrho}{\textit{D7C8}}
1218 \um_set_mathalphabet_char:Nnn{\mathbfsf{fit}}{\um@usv@varpi,\um@usv@itvarpi}{\textit{D7C9}}
1219 }

```

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  $\TeX$  'caret input' form `^^^^^abcdef`. It is *very important* that  
the argument has five characters. Otherwise we need to change the number of `^`  
chars.

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

```

1220 \begingroup
1221   \char_make_other:N \^
1222   \cs_gset:Npn \um@scancharlet#1="#2\@nil {
1223     \lowercase{
1224       \scantokens{\global\let#1=^^^^^#2}
1225     }
1226   }

```

Making `^` the right catcode isn't strictly necessary right now but it helps to future  
proof us with, e.g., `breqn`.

```

1227   \gdef\um@scanactivedef"#1\@nil#2{
1228     \lowercase{
1229       \tl_rescan:nn{
1230         \char_make_math_superscript:N\^
1231       }{
1232         \global\def^^^^^#1{#2}
1233       }
1234     }
1235   }
1236 \endgroup

```

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

```

1237 \begingroup
1238   \def\UnicodeMathSymbol#1#2#3#4{
1239     \um@scancharlet#2=#1\@nil
1240   }
1241   \input{unicode-math-table.tex}
1242 \endgroup

```

## 8 Epilogue

Lots of little things to tidy up.

## 8.0.1 Primes

---

$[x']$	$[x''']$	$[x''''']$	<code>\setmathfont{Cambria Math}</code>
$[x']$	$[x''']$	$[x''''']$	<code>[\$x\prime\$] [\$x\prime\prime\prime\$]</code>
$[x']$	$[x''']$	$[x''''']$	<code>[\$x\prime\prime\prime\$] [\$x\prime\prime\prime\prime\prime\$] \~</code>
$[x']$	$[x''']$	$[x''''']$	<code>[\$x'\$] [\$x'''\$] [\$x''''''\$] \~</code>
$[x']$	$[x''']$	$[x''''']$	<code>[\$x\$] [\$x\$] [\$x\$] [\$x\$]</code>

---

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

```
U+2032: PRIME (\primesingle): x'
U+2033: DOUBLE PRIME (\primedouble): x''
U+2034: TRIPLE PRIME (\primetriple): x'''
U+2057: QUADRUPLE PRIME (\primequadruple): x''''
```

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

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

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

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

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

```

1243 \muskip_new:N \g_um_primekern_muskip
1244 \muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }% arbitrary
1245 \num_new:N \l_um_primecount_num

1246 \cs_new:Nn \um_nprimes:n {
1247   ^{
1248     \primesingle
1249     \prg_replicate:nn {#1-1} { \mskip \g_um_primekern_muskip \primesingle }
1250   }
1251 }
1252 \cs_new:Nn \um_nprimes_select:n {
1253   \prg_case_int:nnn {#1}{
1254     {1} { ^{\primesingle} }
1255     {2} {
1256       \um_glyph_if_exist:nTF {"2033} { ^{\primedouble} } {\um_nprimes:n {#1}}
1257     }
1258     {3} {
1259       \um_glyph_if_exist:nTF {"2034} { ^{\primetriples} } {\um_nprimes:n {#1}}
1260     }
1261     {4} {
1262       \um_glyph_if_exist:nTF {"2057} { ^{\primequadruple} } {\um_nprimes:n {#1}}
1263     }
1264   }{
1265     \um_nprimes:n {#1}
1266   }
1267 }

```

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

```

1268 \cs_new:Nn \um_scanprime: {
1269   \num_zero:N \l_um_primecount_num
1270   \um_scanprime_collect:
1271 }
1272 \cs_new:Nn \um_scanprime_collect: {
1273   \num_incr:N \l_um_primecount_num
1274   \peek_meaning_remove:NTF ' {
1275     \um_scanprime_collect:
1276   }{
1277     \peek_meaning_remove:NTF \um_scanprime: {
1278       \um_scanprime_collect:
1279     }{
1280       \peek_meaning_remove:NTF ^^^^2032 {
1281         \um_scanprime_collect:
1282       }{
1283         \um_nprimes_select:n {\l_um_primecount_num}
1284       }

```

```

1285     }
1286   }
1287 }

1288 \cs_set_eq:NN \prime \um_scanprime:
1289 \group_begin:
1290   \char_make_active:N \'
1291   \char_make_active:n {"2032}
1292   \cs_gset_eq:NN ' \um_scanprime:
1293   \cs_gset_eq:NN ^^^^2032 \um_scanprime:
1294 \group_end:

```

## 8.0.2 Unicode radicals

Undo the damage made to `\sqrt`:

```

1295 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}

```

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

```

1296 \def\r@@t#1#2{
1297   \setbox\z@\hbox{$\math #1\sqrtsign{#2}$}
1298   \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}
1299   \raise \dimexpr(
1300     \um@fontdimen@percent{65}{\um@font}\ht\z@-
1301     \um@fontdimen@percent{65}{\um@font}\dp\z@
1302   )\relax
1303   \copy \rootbox
1304   \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}
1305   \box \z@
1306 }

```

## 8.0.3 Unicode sub- and super-scripts

The idea here is to enter a scanning state after a superscript or subscript is encountered. If subsequent superscripts or subscripts (resp.) are found, they are lumped together. Each sub/super has a corresponding regular size glyph which is used by X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X to typeset the results; this means that the actual subscript/superscript glyphs are never seen in the output document — they are only used as input characters.

Open question: should the superscript-like ‘modifiers’ (U+1D2C: MODIFIER CAPITAL LETTER A and on) be included here?

First, the setup of each mathactive char:

```

1307 \prop_new:N \g_um_supers_prop

```

```

1308 \prop_new:N \g_um_subs_prop
1309 \cs_generate_variant:Nn \prop_gput:Nnn {Nxn}
1310 \cs_generate_variant:Nn \prop_get:NnN {cxN}
1311 \cs_generate_variant:Nn \prop_if_in:NnTF {cx}
1312
1313 \group_begin:
1314
1315 % Populate a property list with superscript characters; their mean-
1316 % ing as their key,
1317 % for reasons that will become apparent soon, and their replace-
1318 % ment as each key's value.
1319 % Then make the superscript active and bind it to the scanning function.
1320 %
1321 % \cs{scantokens} makes this process much simpler since we can acti-
1322 % vate the char
1323 % and assign its meaning in one step.
1324 \cs_set:Nn \um_setup_active_superscript:nn {
1325   \prop_gput:Nxn \g_um_supers_prop {\meaning #1} {#2}
1326   \char_make_active:n {'#1}
1327   \global\XeTeXmathcodenum '#1 = "1FFFFF \scan_stop:
1328   \scantokens{
1329     \cs_gset:Npn #1 {
1330       \tl_set:Nn \l_um_ss_chain_tl {#2}
1331       \cs_set_eq:NN \um_sub_or_super:n \sp
1332       \tl_set:Nn \l_um_tmpa_tl {supers}
1333       \um_scan_ssript:
1334     }
1335   }
1336 }
1337
1338 \um_setup_active_superscript:nn {^^^2070} {0}
1339 \um_setup_active_superscript:nn {^^^00b9} {1}
1340 \um_setup_active_superscript:nn {^^^00b2} {2}
1341 \um_setup_active_superscript:nn {^^^00b3} {3}
1342 \um_setup_active_superscript:nn {^^^2074} {4}
1343 \um_setup_active_superscript:nn {^^^2075} {5}
1344 \um_setup_active_superscript:nn {^^^2076} {6}
1345 \um_setup_active_superscript:nn {^^^2077} {7}
1346 \um_setup_active_superscript:nn {^^^2078} {8}
1347 \um_setup_active_superscript:nn {^^^2079} {9}
1348 \um_setup_active_superscript:nn {^^^207a} {+}
1349 \um_setup_active_superscript:nn {^^^207b} {-}
1350 \um_setup_active_superscript:nn {^^^207c} {=}
1351 \um_setup_active_superscript:nn {^^^207d} {(}
1352 \um_setup_active_superscript:nn {^^^207e} {)}
1353 \um_setup_active_superscript:nn {^^^2071} {i}

```

```

1351 \um_setup_active_superscript:nn {^^^^207f} {n}
1352
1353 % Ditto above.
1354 \cs_set:Nn \um_setup_active_subscript:nn {
1355   \prop_gput:Nxn \g_um_subs_prop {\meaning #1} {#2}
1356   \char_make_active:n {'#1}
1357   \global\XeTeXmathcodenum `#1 = "1FFFFF \scan_stop:
1358   \scantokens{
1359     \cs_gset:Npn #1 {
1360       \tl_set:Nn \l_um_ss_chain_tl {#2}
1361       \cs_set_eq:NN \um_sub_or_super:n \sb
1362       \tl_set:Nn \l_um_tmpa_tl {subs}
1363       \um_scan_ssript:
1364     }
1365   }
1366 }
1367
1368 \um_setup_active_subscript:nn {^^^^2080} {0}
1369 \um_setup_active_subscript:nn {^^^^2081} {1}
1370 \um_setup_active_subscript:nn {^^^^2082} {2}
1371 \um_setup_active_subscript:nn {^^^^2083} {3}
1372 \um_setup_active_subscript:nn {^^^^2084} {4}
1373 \um_setup_active_subscript:nn {^^^^2085} {5}
1374 \um_setup_active_subscript:nn {^^^^2086} {6}
1375 \um_setup_active_subscript:nn {^^^^2087} {7}
1376 \um_setup_active_subscript:nn {^^^^2088} {8}
1377 \um_setup_active_subscript:nn {^^^^2089} {9}
1378 \um_setup_active_subscript:nn {^^^^208a} {+}
1379 \um_setup_active_subscript:nn {^^^^208b} {-}
1380 \um_setup_active_subscript:nn {^^^^208c} {=}
1381 \um_setup_active_subscript:nn {^^^^208d} {(}
1382 \um_setup_active_subscript:nn {^^^^208e} {)}
1383 \um_setup_active_subscript:nn {^^^^2090} {a}
1384 \um_setup_active_subscript:nn {^^^^2091} {e}
1385 \um_setup_active_subscript:nn {^^^^1d62} {i}
1386 \um_setup_active_subscript:nn {^^^^2092} {o}
1387 \um_setup_active_subscript:nn {^^^^1d63} {r}
1388 \um_setup_active_subscript:nn {^^^^1d64} {u}
1389 \um_setup_active_subscript:nn {^^^^1d65} {v}
1390 \um_setup_active_subscript:nn {^^^^2093} {x}
1391 \um_setup_active_subscript:nn {^^^^1d66} {\beta}
1392 \um_setup_active_subscript:nn {^^^^1d67} {\gamma}
1393 \um_setup_active_subscript:nn {^^^^1d68} {\rho}
1394 \um_setup_active_subscript:nn {^^^^1d69} {\phi}
1395 \um_setup_active_subscript:nn {^^^^1d6a} {\chi}
1396

```



```

1397 \group_end:
1398
1399 % The scanning command, evident in its purpose:
1400 \cs_new:Nn \um_scan_sscript: {
1401   \um_scan_sscript:TF {
1402     \um_scan_sscript:
1403   }{
1404     \um_sub_or_super:n {\l_um_ss_chain_tl}
1405   }
1406 }
1407
1408 % The main theme here is stolen from the source to the various \cs{peek_} func-
1409   tions.
1410 % Consider this function as simply boilerplate:
1411 \cs_new:Nn \um_scan_sscript:TF {
1412   \tl_set:Nx \l_peek_true_aux_tl { \exp_not:n{ #1 } }
1413   \tl_set_eq:NN \l_peek_true_tl \c_peek_true_remove_next_tl
1414   \tl_set:Nx \l_peek_false_tl { \exp_not:n{ \group_align_safe_end: #2 } }
1415   \group_align_safe_begin:
1416   \peek_after:NN \um_peek_execute_branches_ss:
1417 }
1418
1419 % We do not skip spaces when scanning ahead, and we explicitly wish to
1420 % bail out on encountering a space or an opening brace.
1421 \cs_new:Npn \um_peek_execute_branches_ss: {
1422   \bool_if:nTF {
1423     \token_if_eq_catcode_p:NN \l_peek_token \c_group_begin_token ||
1424     \token_if_eq_meaning_p:NN \l_peek_token \c_space_token
1425   } { \l_peek_false_tl }
1426   { \um_peek_execute_branches_ss_aux: }
1427 }
1428
1429 % This is the actual comparison code.
1430 % Because the peeking has already tokenised the next token,
1431 % it's too late to extract its charcode directly. Instead,
1432 % we look at its meaning, which remains a 'character' even
1433 % though it is itself math-active. If the character is ever
1434 % made fully active, this will break our assumptions!
1435 %
1436 % If the char's meaning exists as a property list key, we
1437 % build up a chain of sub-/superscripts and iterate. (If not, exit and
1438 % typeset what we've already collected.)
1439 \cs_new:Nn \um_peek_execute_branches_ss_aux: {
1440   \prop_if_in:cxTF
1441     {g_um_\l_um_tmpa_tl _prop}

```

```

1442     {\meaning\l_peek_token}
1443     {
1444         \prop_get:cxN
1445         {g_um_\l_um_tmpa_tl _prop}
1446         {\meaning\l_peek_token}
1447         \l_um_tmpb_tl
1448         \tl_put_right:NV \l_um_ss_chain_tl \l_um_tmpb_tl
1449         \l_peek_true_tl
1450     }
1451     {\l_peek_false_tl}
1452 }

```

## 8.0.4 Synonyms and all the rest

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

```

1453 \def\operator@font{\um_setup_mathup:}
1454 \def\to{\rightarrow}
1455 \def\le{\leq}
1456 \def\ge{\geq}
1457 \def\neq{\neq}

```

`\mathcal`

```

1458 \def\mathcal{\mathscr}

```

`\mathrm`

```

1459 \def\mathrm{\mathup}

```

## 8.0.5 Compatibility

Note that `amsmath` will always be loaded before `unicode-math`. (Conflicts occur if you try it the other way around.)

- Since the mathcode of `\-` is greater than eight bits, this piece of `\AtBeginDocument` code from `amsmath` dies if we try and set the maths font in the preamble:

```

1460 \ifpackageloaded{amsmath}{
1461     \tl_remove_in:Nn \@begindocumenthook {
1462         \mathchardef\std@minus\mathcode`\-\relax
1463         \mathchardef\std@equal\mathcode`\=\relax
1464     }
1465 }{}

```

- This code is to improve the output of alphabetic symbols in text of operator names (`\sin`, `\cos`, etc.). Just comment out the offending lines for now:

```

1466 \ifpackageloaded{amsopn}{
1467   \cs_set:Npn \newmcodes@ {
1468     \mathcode`\'39
1469     \mathcode`\*42
1470     \mathcode`\."613A%
1471   % \ifnum\mathcode`\-45 \else
1472   %   \mathchardef\std@minus\mathcode`\-\relax
1473   % \fi
1474   \mathcode`\-45
1475   \mathcode`\-/47
1476   \mathcode`\:"603A\relax
1477 }
1478 }{}

```

Overriding amsmath definitions:

```

1479 \AtBeginDocument{
1480   \def\@cdots{\mathinner{\cdots}}
1481 }

```

Interaction with beamer:

```

1482 \ifclassloaded{beamer}{
1483   \ifbeamer@suppressreplacements\else
1484     \PackageWarningNoLine{unicode-math}{
1485       Disabling~ beamer's~ math~ setup.^J
1486       Please~ load~ beamer~ with~ the~ [professionalfonts]~ class~ option
1487     }
1488     \beamer@suppressreplacementstrue
1489   \fi
1490 }{}

```

The end.

```

1491 \ExplSyntaxOff

```

## File II

# STIX table data extraction

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

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

A single file is produced containing all (more than 3298) symbols. Future optimisations might include generating various (possibly overlapping) subsets so

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

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

```

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

```

```

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

```

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

```

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

```

Fixing up a couple of things in the STIX table.

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

## A Documenting maths support in the NFSS

### A.1 Overview

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

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

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

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

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

```
\DeclareMathAlphabet{\langle cmd\rangle}\langle NFSS decl.\rangle
```

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{\langle cmd\rangle}\{\langle name\rangle\}
```

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{\langle maths version\rangle}` command.

```
\SetSymbolFont{\langle name\rangle}\{\langle maths version\rangle}\langle NFSS decl.\rangle
```

```
\SetMathAlphabet{\langle cmd\rangle}\{\langle maths version\rangle}\langle NFSS decl.\rangle
```

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

Delimiters and radicals use wrappers around  $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ’s `\delimiter`/`\radical` primitives, which are re-designed in  $\mathrm{X}_{\mathrm{E}}\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ . The syntax used in  $\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ ’s NFSS is therefore not so relevant here.

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

```
\DeclareMathDelimiter{\langle symbol\rangle}\{\langle type\rangle\}\{\langle sym. font\rangle\}\{\langle slot\rangle\}\{\langle sym. font\rangle\}\{\langle slot\rangle\}
```

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

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

Accents are not included yet.

**Summary** For symbols, something like:

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

For characters, something like:

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

## File III

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

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

<code>\fontdimen</code>	Dimension name	Description
10	<code>SCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 1. Suggested value: 80%.
11	<code>SCRIPTSCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 2 (ScriptScript). Suggested value: 60%.
12	<code>DELIMITEDSUBFORMULAMINHEIGHT</code>	Minimum height required for a delimited expression to be treated as a subformula. Suggested value: normal line height $\times$ 1.5.
13	<code>DISPLAYOPERATORMINHEIGHT</code>	Minimum height of n-ary operators (such as integral and summation) for formulas in display mode.

\fontdimen	Dimension name	Description
14	MATHLEADING	White space to be left between math formulas to ensure proper line spacing. For example, for applications that treat line gap as a part of line ascender, formulas with ink going above (os2.sTypoAscender + os2.sTypoLineGap – MathLeading) or with ink going below os2.sTypoDescender will result in increasing line height.
15	AXISHEIGHT	Axis height of the font.
16	ACCENTBASEHEIGHT	Maximum (ink) height of accent base that does not require raising the accents. Suggested: x-height of the font (os2.sxHeight) plus any possible overshoots.
17	FLATTENEDACCENTBASE-HEIGHT	Maximum (ink) height of accent base that does not require flattening the accents. Suggested: cap height of the font (os2.sCapHeight).
18	SUBSCRIPTSHIFTDOWN	The standard shift down applied to subscript elements. Positive for moving in the downward direction. Suggested: os2.ySubscriptYOffset.
19	SUBSCRIPTTOPMAX	Maximum allowed height of the (ink) top of subscripts that does not require moving subscripts further down. Suggested: $\frac{1}{5}$ x-height.
20	SUBSCRIPTBASELINEDROPMIN	Minimum allowed drop of the baseline of subscripts relative to the (ink) bottom of the base. Checked for bases that are treated as a box or extended shape. Positive for subscript baseline dropped below the base bottom.
21	SUPERSCRIPSHIFTUP	Standard shift up applied to superscript elements. Suggested: os2.ySuperscriptYOffset.
22	SUPERSCRIPSHIFTUPCRAMPED	Standard shift of superscripts relative to the base, in cramped style.
23	SUPERSCRIPBOTTOMMIN	Minimum allowed height of the (ink) bottom of superscripts that does not require moving subscripts further up. Suggested: $\frac{1}{4}$ x-height.

\fontdimen	Dimension name	Description
24	SUPERSCRIPTBASELINEDROP-MAX	Maximum allowed drop of the baseline of superscripts relative to the (ink) top of the base. Checked for bases that are treated as a box or extended shape. Positive for superscript baseline below the base top.
25	SUBSUPERSCRIPGAPMIN	Minimum gap between the superscript and subscript ink. Suggested: 4×default rule thickness.
26	SUPERSCRIPBTOPMAX-WITHSUBSCRIPT	The maximum level to which the (ink) bottom of superscript can be pushed to increase the gap between superscript and subscript, before subscript starts being moved down. Suggested: /5 x-height.
27	SPACEAFTERSCRIP	Extra white space to be added after each subscript and superscript. Suggested: 0.5pt for a 12 pt font.
28	UPPERLIMITGAPMIN	Minimum gap between the (ink) bottom of the upper limit, and the (ink) top of the base operator.
29	UPPERLIMITBASELINERISEMIN	Minimum distance between baseline of upper limit and (ink) top of the base operator.
30	LOWERLIMITGAPMIN	Minimum gap between (ink) top of the lower limit, and (ink) bottom of the base operator.
31	LOWERLIMITBASELINEDROP-MIN	Minimum distance between baseline of the lower limit and (ink) bottom of the base operator.
32	STACKTOPSHIFTUP	Standard shift up applied to the top element of a stack.
33	STACKTOPDISPLAYSTYLESHIFT-UP	Standard shift up applied to the top element of a stack in display style.
34	STACKBOTTOMSHIFTDOWN	Standard shift down applied to the bottom element of a stack. Positive for moving in the downward direction.
35	STACKBOTTOMDISPLAYSTYLE-SHIFTDOWN	Standard shift down applied to the bottom element of a stack in display style. Positive for moving in the downward direction.



\fontdimen	Dimension name	Description
36	STACKGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element. Suggested: 3×default rule thickness.
37	STACKDISPLAYSTYLEGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element in display style. Suggested: 7×default rule thickness.
38	STRETCHSTACKTOPSHIFTUP	Standard shift up applied to the top element of the stretch stack.
39	STRETCHSTACKBOTTOMSHIFT-DOWN	Standard shift down applied to the bottom element of the stretch stack. Positive for moving in the downward direction.
40	STRETCHSTACKGAPABOVEMIN	Minimum gap between the ink of the stretched element, and the (ink) bottom of the element above. Suggested: UpperLimitGapMin
41	STRETCHSTACKGAPBELOWMIN	Minimum gap between the ink of the stretched element, and the (ink) top of the element below. Suggested: LowerLimitGapMin.
42	FRACTIONNUMERATORSHIFTUP	Standard shift up applied to the numerator.
43	FRACTIONNUMERATOR-DISPLAYSTYLESHIFTUP	Standard shift up applied to the numerator in display style. Suggested: StackTopDisplayStyleShiftUp.
44	FRACTIONDENOMINATORSHIFT-DOWN	Standard shift down applied to the denominator. Positive for moving in the downward direction.
45	FRACTIONDENOMINATOR-DISPLAYSTYLESHIFTDOWN	Standard shift down applied to the denominator in display style. Positive for moving in the downward direction. Suggested: StackBottomDisplayStyleShiftDown.
46	FRACTIONNUMERATORGAP-MIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar. Suggested: default rule thickness

\fontdimen	Dimension name	Description
47	FRACTIONNUMDISPLAYSTYLE-GAPMIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
48	FRACTIONRULETHICKNESS	Thickness of the fraction bar. Suggested: default rule thickness.
49	FRACTIONDENOMINATORGAP-MIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar. Suggested: default rule thickness
50	FRACTIONDENOMDISPLAY-STYLEGAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
51	SKEWEDFRACTION-HORIZONTALGAP	Horizontal distance between the top and bottom elements of a skewed fraction.
52	SKEWEDFRACTIONVERTICAL-GAP	Vertical distance between the ink of the top and bottom elements of a skewed fraction.
53	OVERBARVERTICALGAP	Distance between the overbar and the (ink) top of the base. Suggested: 3×default rule thickness.
54	OVERBARRULETHICKNESS	Thickness of overbar. Suggested: default rule thickness.
55	OVERBAREXTRAASCENDER	Extra white space reserved above the overbar. Suggested: default rule thickness.
56	UNDERBARVERTICALGAP	Distance between underbar and (ink) bottom of the base. Suggested: 3×default rule thickness.
57	UNDERBARRULETHICKNESS	Thickness of underbar. Suggested: default rule thickness.
58	UNDERBAREXTRADESCENDER	Extra white space reserved below the underbar. Always positive. Suggested: default rule thickness.
59	RADICALVERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: 1¼ default rule thickness.

\fontdimen	Dimension name	Description
60	RADICALDISPLAYSTYLE- VERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: default rule thickness + $\frac{1}{4}$ x-height.
61	RADICALRULETHICKNESS	Thickness of the radical rule. This is the thickness of the rule in designed or constructed radical signs. Suggested: default rule thickness.
62	RADICALEXTRAASCENDER	Extra white space reserved above the radical. Suggested: RadicalRuleThickness.
63	RADICALKERNBEFOREDEGREE	Extra horizontal kern before the degree of a radical, if such is present. Suggested: 5/18 of em.
64	RADICALKERNAFTERDEGREE	Negative kern after the degree of a radical, if such is present. Suggested: $-10/18$ of em.
65	RADICALDEGREEBOTTOM- RAISEPERCENT	Height of the bottom of the radical degree, if such is present, in proportion to the ascender of the radical sign. Suggested: 60%.

## File 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{TeX Gyre Pagella}
6 \usepackage{unicode-math}
7 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
8 \def\uplatin{abcdefghijklmnopqrstuvwxyz}

```

```

9 \def\upGreek{
10 \def\upgreek{
11 \def\itLatin{
12 \def\itlatin{
13 \def\itGreek{
14 \def\itgreek{
15 \def\testmath#1{%
16   \makebox[\linewidth][l]{%
17     \makebox[0pt][l]{$\csname up#1\endcsname$}%
18     \makebox[0pt][l]{$\csname it#1\endcsname$}}
19 \begin{document}
20 \setmathfont[Colour=2255FF99]{Cambria Math}
21 \parindent=0pt
22 \voffset=-1in
23 \hoffset=-1in
24 \setbox0=\vbox{%
25   \testmath{Latin}\\
26   \testmath{latin}\\
27   \testmath{Greek}\\
28   \testmath{greek}}
29 \dimen0=\ht0
30 \advance\dimen0\dp0
31 \edef\papersize{papersize=\the\wd0,\the\dimen0}
32 \setbox255=\vbox{\special{\papersize}\box0}
33 \shipout\box255
34 \end{document}
35 \end{test}

```

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

## C The bold alphabets

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

```

36 \begin{testbf}
37 \documentclass{article}
38 \usepackage[a6paper]{geometry}
39 \usepackage{fontspec}
40 \setmainfont{TeX Gyre Pagella}
41 \usepackage{unicode-math}
42 \def\upLatin{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
43 \def\uplatin{abcdefghijklmnopqrstuvwxyz}
44 \def\upGreek{

```

```

45 \def\upgreek{                                }
46 \def\itLatin{                                }
47 \def\itlatin{                                }
48 \def\itGreek{                                }
49 \def\itgreek{                                }
50 \def\bfupLatin{                                }
51 \def\bfuplatin{                                }
52 \def\bfupGreek{                                }
53 \def\bfupgreek{                                }
54 \def\bfItLatin{                                }
55 \def\bfItlatin{                                }
56 \def\bfItGreek{                                }
57 \def\bfItgreek{                                }
58 \providecommand\mathalphabet{\mathbf}
59 \def\testmath#1{%
60   \makebox[\linewidth][l]{%
61     \makebox[0pt][l]{\mathalphabet{\csname up#1\endcsname}}}%
62     \makebox[0pt][l]{\mathalphabet{\csname it#1\endcsname}}}%
63     \makebox[0pt][l]{\csname bfup#1\endcsname}}}%
64     \makebox[0pt][l]{\csname bfit#1\endcsname}}}%
65   }}
66 \begin{document}
67 \setmathfont[Colour=2255FF55]{Cambria Math}
68 \parindent=0pt
69 \voffset=-1in
70 \hoffset=-1in
71 \setbox0=\vbox{%
72   \testmath{Latin}\\
73   \testmath{latin}\\
74   \testmath{Greek}\\
75   \testmath{greek}}
76 \dimen0=\ht0
77 \advance\dimen0\dp0
78 \edef\papersize{papersize=\the\wd0,\the\dimen0}
79 \setbox255=\vbox{\special{\papersize}\box0}
80 \shipout\box255
81 \end{document}
82 </testbf>

```

## Index

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

Symbols	
\"	17
\'	1287, 1465
\*	1466
\-	1459, 1468, 1469, 1471
\.	1467
\/	1472
\:	1473
\:::	682
\::f	682
\::n	682
\=	1460
\DeclareMathDelimiter	284
\DeclareMathSizes	249
\backslashchar	865
\begindocumenthook	1458
\ecclvi	309
\ecdots	1477
\elt	553–557, 560, 564, 566
\empty	360, 361, 398, 481, 598, 605, 620, 641, 646
\ifclassloaded	1479
\ifnextchar	1292
\ifpackageloaded	1457, 1463
\ii	604, 605, 607, 609, 612, 617
\input	415, 1238
\marker	617, 636
\nil	318, 462, 617, 630–633, 1219, 1224, 1236
\preamblecmds	293
\sqrt	1292
\tempa	120, 164, 190, 199, 217, 235, 371, 387, 596, 608, 632, 634, 646
\tempb	120, 121, 164, 165, 190, 191, 199, 200, 217, 218, 235, 236, 596, 597, 635, 636, 641
\tempswafalse	606
\tempswatruetrue	610, 613, 638, 643, 648, 653
\um@bfliteraltrue	159, 184
\um@bfupGreekfalse	126, 166
\um@bfupGreektrue	138, 150, 172, 178
\um@bfupLatinfalse	129, 169
\um@bfupLatintrue	141, 153, 175, 181
\um@bfupgreekfalse	127, 139, 167, 173
\um@bfupgreektrue	151, 179
\um@bfuplatinfalse	130, 170
\um@bfuplatintrue	142, 154, 176, 182
\um@fontspec@featuretrue	362
\um@literaltrue	158
\um@ot@math@true	390
\um@upGreekfalse	122
\um@upGreektrue	134, 146
\um@upLatinfalse	124, 136
\um@upLatintrue	148
\um@upNablafalse	131, 203
\um@upNablatrie	143, 155, 201
\um@upgreekfalse	123, 135
\um@upgreektrue	147
\um@uplatinfalse	125, 137, 149
\um@uppartialfalse	128, 140, 168, 174, 221
\um@uppartialtrue	152, 180, 219
\xDeclareMathDelimiter	285
\xxDeclareMathDelimiter	283
\	10–13, 17, 23–33, 72–74
\^	33, 1218, 1227
Numbers	
\0	23
\_	17–20, 23–32
A	
\A	24
\a	25
\addnolimits	<u>559</u>
\addtoversion	256
\advance	30, 77
\alloc@	309
\Alpha	709

<code>\alpha</code> .....	734	<code>\clist_map_inline:nn</code> ....	467, 533, 540
<code>\alpha@elt</code> .....	260	<code>\clist_map_variable:NNn</code> .....	604
<code>\alpha@list</code> .....	259	<code>\clist_map_variable:nNn</code> .	661, 684, 690
<code>\AtBeginDocument</code> .....	707, 1476	<code>\copy</code> .....	1300
<code>\awint</code> .....	555	<code>\cs</code> .....	1316, 1405
<b>B</b>			
<code>\B</code> .....	940	<code>\cs_generate_variant:Nn</code> ...	1306–1308
<code>\beamer@suppressreplacementstrue</code>	1485	<code>\cs_gset:cpn</code> .....	323, 333
<code>\begin</code> .....	19, 66	<code>\cs_gset:Npn</code> ..	336, 341, 1219, 1323, 1356
<code>\begingroup</code> .....	315, 563, 1217, 1234	<code>\cs_gset:Npx</code> .....	346
<code>\Beta</code> .....	710	<code>\cs_gset_eq:NN</code> .....	1289, 1290
<code>\beta</code> .....	735, 1388	<code>\cs_if_exist:cF</code> .....	881
<code>\bfitGreek</code> .....	56	<code>\cs_if_exist:cT</code> .....	866
<code>\bfitgreek</code> .....	57	<code>\cs_new:Nn</code> .....	422,
<code>\bfitLatin</code> .....	54		434, 461, 466, 471, 474, 683, 689,
<code>\bfitlatin</code> .....	55		698, 701, 704, 839, 858, 880, 1243,
<code>\bfupGreek</code> .....	52		1249, 1265, 1269, 1397, 1407, 1436
<code>\bfupgreek</code> .....	53	<code>\cs_new:Npn</code>	895, 910, 926, 938, 953, 962,
<code>\bfupLatin</code> .....	50		981, 986, 991, 996, 1073, 1093,
<code>\bfuplatin</code> .....	51		1113, 1118, 1123, 1185, 1201, 1417
<code>\bgroup</code> .....	886	<code>\cs_set:cpn</code> .....	882
<code>\bool_if:NTF</code> .....	739,	<code>\cs_set:Nn</code> .....	206, 224,
	758, 764, 769, 963, 970, 1124, 1153		355, 426, 429, 480, 532, 537, 660,
<code>\bool_if:nTF</code> .....	1418		670, 673, 676, 679, 774, 785, 795,
<code>\bool_new:N</code> .....	22, 187, 188		802, 811, 820, 871, 874, 1318, 1351
<code>\bool_set_false:N</code>	132, 156, 161, 194, 237	<code>\cs_set:Npn</code> .....	682, 1464
<code>\bool_set_true:N</code>	144, 160, 192, 196, 239	<code>\cs_set_eq:cN</code> .....	875
<code>\box</code> .....	32, 33, 79, 80, 1302	<code>\cs_set_eq:NN</code> .....	
<b>C</b>			
<code>\C</code> .....	929, 955		401–404, 408–411, 1285, 1325, 1358
<code>\c_group_begin_token</code> .....	1419	<code>\cs_set_protected:cpn</code> .....	885
<code>\c_peek_true_remove_next_tl</code> ....	1409	<code>\cs_to_str:N</code> .....	333, 548
<code>\c_space_token</code> .....	1420	<code>\csname</code> ....	17, 18, 61–64, 312, 318,
<code>\cdots</code> .....	1477		321, 324, 356, 363, 476, 548, 859, 889
<code>\cdp@elt</code> .....	247	<b>D</b>	
<code>\cdp@list</code> .....	246	<code>\DeclareDocumentCommand</code> .....	358
<code>\char_make_active:N</code> .....	1287	<code>\DeclareMathAccent</code> .....	277
<code>\char_make_active:n</code>	316, 1288, 1320, 1353	<code>\DeclareMathAlphabet</code> .....	273
<code>\char_make_math_superscript:N</code> ..	1227	<code>\DeclareMathDelimiter</code> .....	282
<code>\char_make_other:N</code> .....	1218	<code>\DeclareMathRadical</code> .....	288
<code>\chardef</code> .....	309	<code>\DeclareMathSizes</code> .....	248
<code>\Chi</code> .....	731	<code>\DeclareMathSymbol</code> .....	279
<code>\chi</code> .....	760, 1392	<code>\DeclareMathVersion</code> .....	253, 365
<code>\cirfnint</code> .....	555	<code>\DeclareRobustCommand</code> .....	1292
<code>\clist_map_inline:Nn</code> .....	538	<code>\DeclareSymbolFont</code> .....	266, 413
		<code>\DeclareSymbolFontAlphabet</code> .....	290
		<code>\DeclareSymbolFontAlphabet@</code> .....	291

<code>\def</code> .....	7–15, 23–118, 295, 298, 309, 311, 313, 364, 552, 560, 562, 564, 571, 573, 630, 632–635, 709–738, 741–757, 760–763, 766–768, 771, 772, 1229, 1235, 1293, 1450–1456, 1477
<code>\define@choicekey</code> .....	...
<code>\define@cmdkey</code> .....	592–595
<code>\define@key</code> .....	575
<code>\define@mathalphabet</code> .....	254
<code>\define@mathgroup</code> .....	255
<code>\Delta</code> .....	712
<code>\delta</code> .....	737
<code>\dimen</code> .....	29–31, 76–78
<code>\dimexpr</code> .....	296, 389, 1296
<code>\do</code> .....	293
<code>\documentclass</code> .....	2, 37
<code>\dorestore@version</code> .....	263
<code>\dp</code> .....	30, 77, 1298
<b>E</b>	
<code>\E</code> .....	941
<code>\e</code> .....	949
<code>\edef</code> .....	31, 78, 371, 631, 632
<code>\egroup</code> .....	883, 888
<code>\else</code> .....	211, 229, 301, 304, 330, 334, 339, 344, 347, 391, 405, 441, 454, 485, 493, 496, 501, 507, 519, 529, 565, 578, 611, 616, 622, 640, 645, 650, 798, 806, 815, 829, 1030, 1033, 1039, 1046, 1059, 1468, 1480
<code>\else:</code> .....	878
<code>\encodingdefault</code> .....	414, 872
<code>\end</code> .....	34, 81
<code>\endcsname</code> ...	17, 18, 61–64, 312, 318, 321, 324, 356, 363, 476, 548, 859, 889
<code>\endgroup</code> .....	319, 569, 1233, 1239
<code>\Epsilon</code> .....	713
<code>\epsilon</code> .....	738
<code>\Eta</code> .....	715
<code>\eta</code> .....	742
<code>\etex_iffontchar:D</code> .....	878
<code>\ExecuteOptionsX</code> .....	242
<code>\exp_args:Nnff</code> .....	682, 685, 692
<code>\exp_args:No</code> .....	534, 541
<code>\exp_not:n</code> .....	1408, 1410
<code>\expandafter</code> .....	312, 320, 325, 327, 331, 547, 560, 607, 609, 612, 617, 631, 632, 636, 888, 889
<code>\ExplSyntaxOff</code> .....	1488
<code>\ExplSyntaxOn</code> .....	6
<b>F</b>	
<code>\F</code> .....	942
<code>\f@size</code> .....	363
<code>\fi</code> .....	162, 185, 197, 204, 215, 222, 233, 240, 306, 307, 328, 349–353, 397, 412, 444, 459, 490, 498, 503, 510, 527, 528, 530, 543, 567, 583, 599, 614, 615, 618, 624, 626, 627, 639, 644, 649, 654–658, 800, 809, 818, 837, 1035, 1042, 1049, 1068, 1071, 1470, 1486
<code>\fi:</code> .....	878
<code>\fint</code> .....	555
<code>\font</code> .....	388
<code>\fontdimen</code> .....	296, 389, 1295, 1301
<code>\fontname</code> .....	865
<b>G</b>	
<code>\g</code> .....	950
<code>\g@addto@macro</code> .....	547, 621, 623
<code>\g_um_primekern_muskip</code>	1240, 1241, 1246
<code>\g_um_sfliteral_bool</code> .....	...
<code>\g_um_subs_prop</code> .....	1305, 1352
<code>\g_um_supers_prop</code> .....	1304, 1319
<code>\g_um_texgreek_bool</code> ..	22, 132, 144, 156, 161, 237, 239, 739, 758, 764, 769
<code>\g_um_upsans_bool</code>	187, 192, 194, 970, 1153
<code>\Gamma</code> .....	711
<code>\gamma</code> .....	736, 1389
<code>\gdef</code> .....	1224
<code>\ge</code> .....	1453
<code>\geq</code> .....	1453
<code>\get@cdp</code> .....	272
<code>\glb@currsz</code> .....	359
<code>\global</code> .....	317, 320, 337, 338, 342, 343, 348, 598, 1221, 1229, 1321, 1354
<code>\group@elt</code> .....	268
<code>\group@list</code> .....	267
<code>\group_align_safe_begin:</code> .....	1411
<code>\group_align_safe_end:</code> .....	1410



\group_begin: . . . . .	1286, 1310	\intx . . . . .	556
\group_end: . . . . .	1291, 1394	\Iota . . . . .	717
<b>H</b>		\iota . . . . .	744
\H . . . . .	930, 943, 956	\itGreek . . . . .	13, 48
\h . . . . .	805, 808, 913	\itgreek . . . . .	14, 49
\hbox . . . . .	1294	\itLatin . . . . .	11, 46
\hoffset . . . . .	23, 70	\itlatin . . . . .	12, 47
\ht . . . . .	29, 76, 1297	<b>K</b>	
<b>I</b>		\Kappa . . . . .	718
\I . . . . .	944, 957	\kappa . . . . .	745
\if@tempwa . . . . .	619	\kern . . . . .	1295, 1301
\if@um@bfliteral . . . .	15, 445, 491, 1000	<b>L</b>	
\if@um@bfupGreek . . . .	16, 504, 1043	\L . . . . .	945
\if@um@bfupgreek . . . .	17, 511, 1050	\l_peek_false_tl . . . .	1410, 1422, 1448
\if@um@bfupLatin . . . .	18, 494, 1031	\l_peek_token . . . .	1419, 1420, 1439, 1443
\if@um@bfuplatin . . . .	19, 499, 1036	\l_peek_true_aux_tl . . . .	1408
\if@um@fontspec@feature . . . .	8, 576	\l_peek_true_tl . . . .	1409, 1446
\if@um@literal . . . . .	10, 436, 483	\l_um_inc_num . . . . .	691, 693, 694
\if@um@ot@math@ . . . . .	9	\l_um_incr_num . . . . .	662, 664, 666
\if@um@upGreek . . . . .	11, 812	\l_um_input_num 661, 664, 684, 686, 690, 693	
\if@um@upgreek . . . . .	12, 821	\l_um_primecount_num . . . . .	
\if@um@uplatin . . . . .	13, 796	. . . . .	1242, 1266, 1270, 1280
\if@um@uplatin . . . . .	14, 803	\l_um_script_features_tl . . . .	366, 377
\if@um@upNabla . . . . .	20, 207	\l_um_script_font_tl . . . . .	368, 376
\if@um@uppartial . . . . .	21, 225	\l_um_ss_chain_tl 1324, 1357, 1401, 1445	
\ifbeamer@suppressreplacements .	1480	\l_um_sscript_features_tl . . .	367, 381
\ifcase . . . . .	121, 165, 191, 200, 218, 236, 597	\l_um_sscript_font_tl . . . . .	369, 380
\ifdim . . . . .	389	\l_um_tpa_tl . . . . .	1326, 1359, 1438, 1442
\ifin@ . . . . .	326, 332	\l_um_tpb_tl . . . . .	1444, 1445
\ifnum . . . . .	539, 637, 642, 647, 651, 652, 1468	\Lambda . . . . .	719
\ifx . . . . .	299, 302,	\lambda . . . . .	746
	314, 335, 340, 345, 398, 481, 565,	\le . . . . .	1452
	605, 608, 609, 612, 620, 636, 641, 646	\left . . . . .	<u>572</u>
\iiiint . . . . .	553	\left@primitive . . . . .	572, 573
\iiint . . . . .	553	\leq . . . . .	1452
\iint . . . . .	553	\let . . . . .	310, 359–361, 572, 598, 1221
\in@ . . . . .	325, 331	\linewidth . . . . .	16, 60
\init@restore@version . . . . .	262	\lowercase . . . . .	1220, 1225
\int . . . . .	553	\lowint . . . . .	557
\intBar . . . . .	555	<b>M</b>	
\intbar . . . . .	555	\M . . . . .	946
\intcap . . . . .	557	\m@th . . . . .	1294
\intclockwise . . . . .	554	\makebox . . . . .	16–18, 60–64
\intcup . . . . .	557	\mathaccent . . . . .	345
\intlarhk . . . . .	556		

<code>\mathalpha</code> .....	549, 665	<code>\mitDelta</code> .....	712
<code>\mathalphabet</code> .....	58, 61, 62	<code>\mitdelta</code> .....	737
<code>\mathbb</code> .....	927–936	<code>\mitEpsilon</code> .....	713
<code>\mathbf</code> ....	58, 997–999, 1001–1029, 1032, 1034, 1037, 1038, 1040, 1041, 1044, 1045, 1047, 1048, 1051–1058, 1060–1067, 1069, 1070	<code>\mitepsilon</code> .....	739, 764
<code>\mathbffrak</code> .....	1114–1116	<code>\mitEta</code> .....	715
<code>\mathbfit</code> .....	1074–1091	<code>\miteta</code> .....	742
<code>\mathbfscr</code> .....	1119–1121	<code>\mitGamma</code> .....	711
<code>\mathbfssf</code> .....	1125–1151, 1154–1167, 1169–1182	<code>\mitgamma</code> .....	736
<code>\mathbfssfit</code> .....	1202–1215	<code>\mitIota</code> .....	717
<code>\mathbfsfup</code> .....	1186–1199	<code>\mitiota</code> .....	744
<code>\mathbfsup</code> .....	1094–1111	<code>\mitKappa</code> .....	718
<code>\mathbin</code> .....	435	<code>\mitkappa</code> .....	745
<code>\mathcal</code> .....	1455	<code>\mitLambda</code> .....	719
<code>\mathchar@type</code> .....	289, 322, 336, 338, 341, 343, 346, 348, 356, 475	<code>\mitlambda</code> .....	746
<code>\mathchardef</code> .....	1459, 1460, 1469	<code>\mitMu</code> .....	720
<code>\mathclose</code> .....	340	<code>\mitmu</code> .....	747
<code>\mathcode</code> .....	317, 1459, 1460, 1465–1469, 1471–1473	<code>\mitNu</code> .....	721
<code>\mathfrak</code> .....	954–960	<code>\mitnu</code> .....	748
<code>\mathgroup</code> .....	309	<code>\mitOmega</code> .....	733
<code>\mathinner</code> .....	1477	<code>\mitomega</code> .....	762
<code>\mathit</code> .....	911–924	<code>\mitOmicron</code> .....	723
<code>\mathop</code> .....	314	<code>\mitomicron</code> .....	750
<code>\mathopen</code> .....	335, 573	<code>\mitPhi</code> .....	730
<code>\mathord</code> .....	437–440, 442, 443, 446–453, 455–458, 472	<code>\mitphi</code> .....	758, 769
<code>\mathrm</code> .....	1456	<code>\mitPi</code> .....	724
<code>\mathscr</code> .....	939–951, 1455	<code>\mitpi</code> .....	751
<code>\mathsf</code> ....	964–968, 971–973, 975–977	<code>\mitPsi</code> .....	732
<code>\mathsf fit</code> .....	987–989	<code>\mitpsi</code> .....	761
<code>\mathsfup</code> .....	982–984	<code>\mitRho</code> .....	725
<code>\mathtt</code> .....	872, 992–994	<code>\mitrho</code> .....	752
<code>\mathup</code> .....	896–908, 1456	<code>\mitSigma</code> .....	727
<code>\mddefault</code> .....	414, 872	<code>\mitsigma</code> .....	754
<code>\meaning</code> .....	1319, 1352, 1439, 1443	<code>\mitTau</code> .....	728
<code>\mitAlpha</code> .....	709	<code>\mittau</code> .....	755
<code>\mitalpha</code> .....	734	<code>\mitTheta</code> .....	716
<code>\mitBeta</code> .....	710	<code>\mittheta</code> .....	743
<code>\mitbeta</code> .....	735	<code>\mitUpsilon</code> .....	729
<code>\mitChi</code> .....	731	<code>\mitupsilon</code> .....	756
<code>\mitchi</code> .....	760	<code>\mitvarepsilon</code> .....	739, 764
		<code>\mitvarkappa</code> .....	767
		<code>\mitvarphi</code> .....	758, 769
		<code>\mitvarpi</code> .....	772
		<code>\mitvarrho</code> .....	771
		<code>\mitvarsigma</code> .....	753
		<code>\mitvarTheta</code> .....	726
		<code>\mitvartheta</code> .....	766

\mitXi .....	722	\omega .....	762
\mitxi .....	749	\Omicron .....	723
\mitZeta .....	714	\omicron .....	750
\mitzeta .....	741	\operator@font .....	1450
\mode_if_math:F .....	887	\or .....	133, 145, 157, 171, 177, 183, 193, 195, 202, 220, 238
\mskip .....	1246		
\Mu .....	720		
\mu .....	747		
\muskip_gset:Nn .....	1241		
\muskip_new:N .....	1240		
	<b>N</b>		<b>P</b>
\N .....	931	\P .....	932
\ne .....	1454	\PackageError .....	579
\neq .....	1454	\PackageInfo .....	400
\new@mathalphabet .....	274	\PackageWarningNoLine ..	392, 864, 1481
\new@mathgroup .....	245, 309, 310	\papersize .....	31, 32, 78, 79
\new@mathversion .....	265	\parindent .....	21, 68
\new@symbolfont .....	269	\peek_after:NN .....	1412
\newcommand .....	546, 559, 574, 603, 708	\peek_meaning_remove:NTF .....	1271, 1274, 1277
\newcounter .....	7	\Phi .....	730
\newfam .....	310	\phi .....	757, 1391
\newif .....	8–21	\Pi .....	724
\newmathalphabet .....	250	\pi .....	751
\newmathalphabet@@ .....	251	\pointint .....	556
\newmathalphabet@@@ .....	252	\prg_case_int:nnn .....	1250
\newmcodes@ .....	1464	\prg_do_nothing: .....	875
\noexpand .....	371, 566	\prg_new_conditional:Nnn .....	877
\nolimits .....	327	\prg_replicate:nn .....	1246
\non@alpherr .....	889	\prg_return_false: .....	878
\npolint .....	556	\prg_return_true: .....	878
\Nu .....	721	\prg_stepwise_variable:nnnNn	662, 691
\nu .....	748	\prime .....	1285
\num_incr:N .....	1270	\primedouble .....	1253
\num_new:N .....	1242	\primequadruple .....	1259
\num_zero:N .....	1266	\primesingle .....	472, 1245, 1246, 1251
\number .....	686, 693, 694	\primetripel .....	1256
\numexpr .....	642, 647, 651, 652, 664, 666, 686, 693, 694	\process@table .....	264
		\ProcessOptionsX .....	243
		\prop_get:cxN .....	1441
		\prop_get:NnN .....	1307
		\prop_gput:Nnn .....	1306
		\prop_gput:Nxn .....	1319, 1352
		\prop_if_in:cxTF .....	1437
		\prop_if_in:NnTF .....	1308
		\prop_new:N .....	1304, 1305
		\protect .....	582
		\providecommand .....	58
		\ProvidesPackage .....	1
		\Psi .....	732
	<b>O</b>		
\o .....	951		
\oiint .....	553		
\oint .....	553		
\oint .....	553		
\ointctrlockwise .....	554		
\Omega .....	733		

<code>\psi</code> .....	761	<code>\Sigma</code> .....	727
<b>Q</b>		<code>\sigma</code> .....	754
<code>\Q</code> .....	933	<code>\sp</code> .....	1325
<b>R</b>		<code>\space</code> .....	864, 889
<code>\R</code> .....	934, 947, 958	<code>\special</code> .....	32, 79
<code>\r@t</code> .....	1293	<code>\sqint</code> .....	556
<code>\raise</code> .....	1296	<code>\sqrt</code> .....	571, 1292
<code>\relax</code> .....	121,	<code>\sqrtsign</code> .....	1292, 1294
	165, 191, 200, 218, 236, 296, 314,	<code>\std@equal</code> .....	1460
	317, 322, 333, 335–338, 340–343,	<code>\std@minus</code> .....	1459, 1469
	345, 346, 348, 356, 359, 388, 389,	<code>\stepcounter</code> .....	406
	539, 597, 609, 612, 636, 637, 642,	<code>\string</code> .....	318, 321, 323, 324, 631, 632
	647, 651, 652, 664, 666, 686, 693,	<code>\strip@pt</code> .....	296
	694, 1299, 1459, 1460, 1469, 1473	<code>\sumint</code> .....	554
<code>\removenolimits</code> .....	562	<b>T</b>	
<code>\RequirePackage</code> .....	3–5	<code>\Tau</code> .....	728
<code>\restore@mathversion</code> .....	261	<code>\tau</code> .....	755
<code>\Rho</code> .....	725	<code>\testmath</code> .....	15, 25–28, 59, 72–75
<code>\rho</code> .....	752, 1390	<code>\tf@size</code> .....	374, 375
<code>\rightarrow</code> .....	1451	<code>\the</code> .....	31, 78
<code>\rootbox</code> .....	1300	<code>\Theta</code> .....	716
<code>\rppoint</code> .....	555	<code>\theta</code> .....	743
<b>S</b>		<code>\theum@fam</code> .....	407
<code>\sb</code> .....	1358	<code>\thinmuskip</code> .....	1241
<code>\scan_stop:</code> .....	477, 478, 878, 1321, 1354	<code>\tl_map_inline:nn</code> .....	244
<code>\scantokens</code> .....	1221, 1322, 1355	<code>\tl_put_right:N</code> .....	1445
<code>\scpolint</code> .....	556	<code>\tl_remove_in:Nn</code> .....	293, 1458
<code>\scriptscriptstyle</code> .....	302	<code>\tl_rescan:nn</code> .....	1226
<code>\scriptstyle</code> .....	299	<code>\tl_set:Nn</code> .....	208–210,
<code>\set@mathdelimiter</code> .....	287		212–214, 226–228, 230–232,
<code>\set@mathaccent</code> .....	278		366–369, 399, 1324, 1326, 1357, 1359
<code>\set@mathchar</code> .....	280	<code>\tl_set:Nx</code> .....	407, 1408, 1410
<code>\set@mathdelimiter</code> .....	286	<code>\tl_set_eq:NN</code> .....	1409
<code>\set@mathsymbol</code> .....	281	<code>\to</code> .....	1451
<code>\setbox</code> .....	24, 32, 71, 79, 1294	<code>\token_if_eq_catcode_p:NN</code> .....	1419
<code>\setkeys</code> .....	370	<code>\token_if_eq_meaning_p:NN</code> .....	1420
<code>\setmainfont</code> .....	5, 40	<code>\ttdefault</code> .....	872
<code>\SetMathAlphabet</code> .....	275, 872	<b>U</b>	
<code>\SetMathAlphabet@</code> .....	276	<code>\um@addto@mathmap</code> .....	534, 541, 546
<code>\setmathfont</code> .....	20, 67, 358, 582	<code>\um@backslash</code> .....	608, 631
<code>\SetSymbolFont</code> .....	270	<code>\um@char@num@range</code> .....	361, 538, 620, 621, 623
<code>\SetSymbolFont@</code> .....	271	<code>\um@char@range</code> .....	360, 398, 481, 598, 601, 604
<code>\sf@size</code> .....	375, 379	<code>\um@firstchar</code> .....	607, 632
<code>\shipout</code> .....	33, 80	<code>\um@firstof</code> .....	630–632

$\backslash\mathrm{um@font}$ . . . . .	300, 303, 388, 389, 865, 878, 1295, 1297, 1298, 1301
$\backslash\mathrm{um@fontdimen@percent}$ . . . . .	295, 300, 303, 1297, 1298
$\backslash\mathrm{um@mathsymbol}$ . . . . .	311, 427
$\backslash\mathrm{um@mversion}$ . . . . .	364, 365
$\backslash\mathrm{um@nolimits}$ . . . . .	325, 552, 560, 568
$\backslash\mathrm{um@parse@range}$ . . . . .	617, 633
$\backslash\mathrm{um@parse@term}$ . . . . .	430, 462, 603
$\backslash\mathrm{um@radicals}$ . . . . .	331, 571
$\backslash\mathrm{um@resolve@greek}$ . . . . .	707
$\backslash\mathrm{um@scaled@apply}$ . . . . .	298, 1295, 1301
$\backslash\mathrm{um@scanactivedef}$ . . . . .	318, 1217
$\backslash\mathrm{um@scancharlet}$ . . . . .	1217, 1236
$\backslash\mathrm{um@set@mathsymbol}$ . . . . .	312, 313
$\backslash\mathrm{um@usv@bbLatin}$ . . . . .	33, 928
$\backslash\mathrm{um@usv@bblatin}$ . . . . .	34, 936
$\backslash\mathrm{um@usv@bbnum}$ . . . . .	32, 927
$\backslash\mathrm{um@usv@bfDigamma}$ . . . . .	82, 1012
$\backslash\mathrm{um@usv@bfdigamma}$ . . . . .	89, 1020
$\backslash\mathrm{um@usv@bffrakLatin}$ . . . . .	58, 1115
$\backslash\mathrm{um@usv@bffraklatin}$ . . . . .	59, 1116
$\backslash\mathrm{um@usv@bfitGreek}$ . . . . .	56, 505, 508, 792, 1006, 1047, 1077, 1081
$\backslash\mathrm{um@usv@bfitgreek}$ . . . . .	57, 512, 520, 793, 1008, 1060, 1078, 1082
$\backslash\mathrm{um@usv@bfith}$ . . . . .	99, 1009, 1041
$\backslash\mathrm{um@usv@bfitLatin}$ . . . . .	54, 495, 497, 788, 1002, 1034, 1075, 1079
$\backslash\mathrm{um@usv@bfitlatin}$ . . . . .	55, 500, 502, 789, 1004, 1040, 1076, 1080
$\backslash\mathrm{um@usv@bfitNabla}$ . . . . .	110, 213, 447, 455, 1022, 1084
$\backslash\mathrm{um@usv@bfitpartial}$ . . . . .	116, 231, 451, 457, 1023, 1067, 1085
$\backslash\mathrm{um@usv@bfitvarepsilon}$ . . . . .	101, 513, 521, 1024, 1061, 1086
$\backslash\mathrm{um@usv@bfitvarkappa}$ . . . . .	103, 515, 523, 1026, 1063, 1088
$\backslash\mathrm{um@usv@bfitvarphi}$ . . . . .	104, 516, 524, 1027, 1064, 1089
$\backslash\mathrm{um@usv@bfitvarpi}$ . . . . .	106, 518, 526, 1029, 1066, 1091
$\backslash\mathrm{um@usv@bfitvarrho}$ . . . . .	105, 517, 525, 1028, 1065, 1090
$\backslash\mathrm{um@usv@bfitvarTheta}$ . . . . .	100, 506, 509, 1021, 1048, 1083
$\backslash\mathrm{um@usv@bfitvartheta}$ . . . . .	102, 514, 522, 1025, 1062, 1087
$\backslash\mathrm{um@usv@bfLatin}$ . . . . .	50
$\backslash\mathrm{um@usv@bfNabla}$ . . . . .	109, 209, 446, 455, 1011, 1104
$\backslash\mathrm{um@usv@bfnum}$ . . . . .	48, 997, 1074, 1094, 1114, 1119, 1154, 1169, 1186, 1202
$\backslash\mathrm{um@usv@bfpartial}$ . . . . .	115, 227, 450, 457, 1013, 1058, 1105
$\backslash\mathrm{um@usv@bfscrLatin}$ . . . . .	60, 1120
$\backslash\mathrm{um@usv@bfscrlatin}$ . . . . .	61, 1121
$\backslash\mathrm{um@usv@bfsfitGreek}$ . . . . .	70, 1132, 1172, 1205
$\backslash\mathrm{um@usv@bfsfitgreek}$ . . . . .	71, 1133, 1173, 1206
$\backslash\mathrm{um@usv@bfsfitLatin}$ . . . . .	68, 1128, 1170, 1203
$\backslash\mathrm{um@usv@bfsfitlatin}$ . . . . .	69, 1129, 1171, 1204
$\backslash\mathrm{um@usv@bfsfitNabla}$ . . . . .	112, 214, 449, 456, 1144, 1175, 1208
$\backslash\mathrm{um@usv@bfsfitpartial}$ . . . . .	118, 232, 453, 458, 1145, 1176, 1209
$\backslash\mathrm{um@usv@bfsfLatin}$ . . . . .	64
$\backslash\mathrm{um@usv@bfsfNabla}$ . . . . .	111, 210, 448, 456
$\backslash\mathrm{um@usv@bfsfnum}$ . . . . .	62
$\backslash\mathrm{um@usv@bfsfpartial}$ . . . . .	117, 228, 452, 458
$\backslash\mathrm{um@usv@bfsfupGreek}$ . . . . .	66, 1130, 1157, 1189
$\backslash\mathrm{um@usv@bfsfupgreek}$ . . . . .	67, 1131, 1158, 1190
$\backslash\mathrm{um@usv@bfsfupLatin}$ . . . . .	63, 1126, 1155, 1187
$\backslash\mathrm{um@usv@bfsfuplatin}$ . . . . .	65, 1127, 1156, 1188
$\backslash\mathrm{um@usv@bfupGreek}$ . . . . .	52, 505, 508, 790, 1005, 1044, 1081, 1097, 1101
$\backslash\mathrm{um@usv@bfupgreek}$ . . . . .	53, 512, 520, 791, 1007, 1051, 1082, 1098, 1102
$\backslash\mathrm{um@usv@bfuph}$ . . . . .	98, 1038
$\backslash\mathrm{um@usv@bfupLatin}$ . . . . .	49, 495, 497, 786, 1001, 1032, 1079, 1095, 1099
$\backslash\mathrm{um@usv@bfuplatin}$ . . . . .	51, 500, 502, 787, 1003, 1037, 1080, 1096, 1100
$\backslash\mathrm{um@usv@bfvarepsilon}$ . . . . .	83, 513, 521, 1014, 1052, 1106
$\backslash\mathrm{um@usv@bfvarkappa}$ . . . . .	85, 515, 523, 1016, 1054, 1108
$\backslash\mathrm{um@usv@bfvarphi}$ . . . . .	86, 516, 524, 1017, 1055, 1109
$\backslash\mathrm{um@usv@bfvarpi}$ . . . . .	88, 518, 526, 1019, 1057, 1111

$\backslash\mathrm{um@usv@bfvarrho}$ . . . . .	1111, 1151, 1167, 1182, 1199, 1215
. . . . . 87, 517, 525, 1018, 1056, 1110	$\backslash\mathrm{um@usv@itvarrho}$ . . . . . 96, 827, 835,
$\backslash\mathrm{um@usv@bfvartheta}$ . . . . .	907, 923, 1028, 1056, 1065, 1090,
. . . . . 81, 506, 509, 1010, 1045, 1103	1110, 1150, 1166, 1181, 1198, 1214
$\backslash\mathrm{um@usv@bfvartheta}$ . . . . .	$\backslash\mathrm{um@usv@itvartheta}$ . . . . .
. . . . . 84, 514, 522, 1015, 1053, 1107	. . . . . 91, 817, 902, 918, 1021,
$\backslash\mathrm{um@usv@Digamma}$ . . . . . 73, 998, 1012	1045, 1048, 1083, 1103, 1159, 1191
$\backslash\mathrm{um@usv@digamma}$ . . . . . 80, 999, 1020	$\backslash\mathrm{um@usv@itvartheta}$ . . . . . 93, 824, 832,
$\backslash\mathrm{um@usv@frakLatin}$ . . . . . 37, 954, 1115	904, 920, 1025, 1053, 1062, 1087,
$\backslash\mathrm{um@usv@fraklatin}$ . . . . . 38, 960, 1116	1107, 1147, 1163, 1178, 1195, 1211
$\backslash\mathrm{um@usv@itGreek}$ . . . . . 30, 782, 813, 816,	$\backslash\mathrm{um@usv@Nabla}$ . . . . . 107, 208, 437,
898, 914, 1006, 1044, 1047, 1077,	442, 900, 916, 1011, 1069, 1084,
1097, 1132, 1157, 1172, 1189, 1205	1104, 1135, 1160, 1175, 1192, 1208
$\backslash\mathrm{um@usv@itgreek}$ . . . . . 31, 822, 830,	$\backslash\mathrm{um@usv@num}$ . . . . .
899, 915, 1008, 1051, 1060, 1078,	. . . . . 23, 482, 927, 964, 971, 975, 982,
1098, 1133, 1158, 1173, 1190, 1206	987, 992, 997, 1074, 1094, 1114,
$\backslash\mathrm{um@usv@ith}$ . . . . . 90,	1119, 1125, 1154, 1169, 1186, 1202
778, 805, 808, 913, 1009, 1038, 1041	$\backslash\mathrm{um@usv@partial}$ 113, 226, 439, 443, 901,
$\backslash\mathrm{um@usv@itLatin}$ . . . . . 28, 776,	917, 1013, 1058, 1067, 1070, 1085,
797, 799, 896, 911, 928, 939, 954,	1105, 1136, 1161, 1176, 1193, 1209
967, 972, 976, 983, 988, 993, 1002,	$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 35, 939
1032, 1034, 1075, 1095, 1115,	$\backslash\mathrm{um@usv@scrLatin}$ . . . . . 36, 948
1120, 1128, 1155, 1170, 1187, 1203	$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 43, 967, 976, 988
$\backslash\mathrm{um@usv@itlatin}$ . . . . . 29, 777,	$\backslash\mathrm{um@usv@sfitLatin}$ . . . . . 44, 968, 977, 989
804, 807, 897, 912, 936, 948, 960,	$\backslash\mathrm{um@usv@sfLatin}$ . . . . . 41
968, 973, 977, 984, 989, 994, 1004,	$\backslash\mathrm{um@usv@sfnum}$ . . . . .
1037, 1040, 1076, 1096, 1116,	. . . . . 39, 964, 971, 975, 982, 987, 1125
1121, 1129, 1156, 1171, 1188, 1204	$\backslash\mathrm{um@usv@sfupLatin}$ . . . . . 40, 965, 972, 983
$\backslash\mathrm{um@usv@itNabla}$ . . . . . 108, 212, 438,	$\backslash\mathrm{um@usv@sfupLatin}$ . . . . . 42, 966, 973, 984
442, 900, 916, 1022, 1069, 1084,	$\backslash\mathrm{um@usv@ttLatin}$ . . . . . 46, 993
1104, 1144, 1160, 1175, 1192, 1208	$\backslash\mathrm{um@usv@ttlLatin}$ . . . . . 47, 994
$\backslash\mathrm{um@usv@itpartial}$ . . . . .	$\backslash\mathrm{um@usv@ttnum}$ . . . . . 45, 992
. . . . . 114, 230, 440, 443, 901,	$\backslash\mathrm{um@usv@upGreek}$ . . . . . 26, 780, 813, 816,
917, 1023, 1058, 1067, 1070, 1085,	898, 914, 1005, 1044, 1047, 1077,
1105, 1145, 1161, 1176, 1193, 1209	1097, 1130, 1157, 1172, 1189, 1205
$\backslash\mathrm{um@usv@itvarepsilon}$ . . . . . 92, 823, 831,	$\backslash\mathrm{um@usv@upgreek}$ . . . . . 27, 783, 822, 830,
903, 919, 1024, 1052, 1061, 1086,	899, 915, 1007, 1051, 1060, 1078,
1106, 1146, 1162, 1177, 1194, 1210	1098, 1131, 1158, 1173, 1190, 1206
$\backslash\mathrm{um@usv@itvarkappa}$ . . . . . 94, 825, 833,	$\backslash\mathrm{um@usv@upLatin}$ . . . . . 24, 775,
905, 921, 1026, 1054, 1063, 1088,	797, 799, 896, 911, 928, 939, 954,
1108, 1148, 1164, 1179, 1196, 1212	965, 972, 976, 983, 988, 993, 1001,
$\backslash\mathrm{um@usv@itvarphi}$ . . . . . 95, 826, 834,	1032, 1034, 1075, 1095, 1115,
906, 922, 1027, 1055, 1064, 1089,	1120, 1126, 1155, 1170, 1187, 1203
1109, 1149, 1165, 1180, 1197, 1213	$\backslash\mathrm{um@usv@uplatin}$ . . . . . 25, 779,
$\backslash\mathrm{um@usv@itvarpi}$ . . . . . 97, 828, 836,	804, 807, 897, 912, 936, 948, 960,
908, 924, 1029, 1057, 1066, 1091,	966, 973, 977, 984, 989, 994, 1003,

1037, 1040, 1076, 1096, 1116, 1121, 1127, 1156, 1171, 1188, 1204	\um_config_mathtt: ..... 991
\um@usv@varepsilon ... 74, 823, 831, 903, 919, 1014, 1052, 1061, 1086, 1106, 1137, 1162, 1177, 1194, 1210	\um_config_mathup: ..... 895
\um@usv@varkappa .... 76, 825, 833, 905, 921, 1016, 1054, 1063, 1088, 1108, 1139, 1164, 1179, 1196, 1212	\um_fix_mathtt: ..... 871
\um@usv@varphi ..... 77, 826, 834, 906, 922, 1017, 1055, 1064, 1089, 1109, 1140, 1165, 1180, 1197, 1213	\um_glyph_if_exist:n ..... 877
\um@usv@varpi ..... 79, 828, 836, 908, 924, 1019, 1057, 1066, 1091, 1111, 1142, 1167, 1182, 1199, 1215	\um_glyph_if_exist:nTF ..... ..... 859, <u>877</u> , 1253, 1256, 1259
\um@usv@varrho ..... 78, 827, 835, 907, 923, 1018, 1056, 1065, 1090, 1110, 1141, 1166, 1181, 1198, 1214	\um_init_alphabet:n ..... 404, 874
\um@usv@varTheta 72, 781, 814, 817, 902, 918, 1010, 1045, 1048, 1083, 1103, 1134, 1143, 1159, 1174, 1191, 1207	\um_make_mathactive:nNN ..... <u>472</u> , <u>474</u>
\um@usv@vartheta .... 75, 824, 832, 904, 920, 1015, 1053, 1062, 1087, 1107, 1138, 1163, 1178, 1195, 1211	\um_map_char:nn ..... 506, 509, 513–518, 521–526, 679, 778, 781, 805, 808, 814, 817, 823–828, 831–836
\um@zf@feature ..... <u>574</u> , 586, 589	\um_map_char:nn_ ..... <u>660</u>
\um@bfNabla_up_or_it_usv ..... ..... 209, 213, 455, 1069	\um_map_chars_greek:nn ..... ... 505, 508, 512, 520, 673, 780, 782, 783, 790–793, 813, 816, 822, 830
\um@bfpartial_up_or_it_usv ..... ..... 227, 231, 457, 1070	\um_map_chars_latin:nn ..... 495, 497, 500, 502, 670, 775–777, 779, 786–789, 797, 799, 804, 807
\um@bfsfNabla_up_or_it_usv 210, 214, 456	\um_map_chars_numbers:nn .... 482, 676
\um@bfsfpartial_up_or_it_usv .... ..... 228, 232, 458	\um_map_chars_range:nnn ..... ..... 660, 671, 674, 677, 680
\um_config_mathbb: ..... 926	\um_mathmap:Nnn ..... 402, 409, 685, 692
\um_config_mathbf: ..... 996	\um_mathmap_noparse:Nnn .... 402, <u>532</u>
\um_config_mathbffrak: ..... 1113	\um_mathmap_parse:Nnn ..... 409, <u>537</u>
\um_config_mathbfit: ..... 1073	\um_maybe_init_alphabet:n 404, 411, 860
\um_config_mathbfscr: ..... 1118	\um_Nabla_up_or_it_usv .. 208, 212, 442
\um_config_mathbfssf: ..... 1123	\um_nprimes:n 1243, 1253, 1256, 1259, 1262
\um_config_mathbfssfit: ..... 1201	\um_nprimes_select:n ..... 1249, 1280
\um_config_mathbfsfup: ..... 1185	\um_partial_up_or_it_usv . 226, 230, 443
\um_config_mathbfup: ..... 1093	\um_peek_execute_branches_ss: ... ..... 1412, 1417
\um_config_mathfrak: ..... 953	\um_peek_execute_branches_ss_aux: ..... 1423, 1436
\um_config_mathhit: ..... 910	\um_prepare_alph:n ..... 861, <u>880</u>
\um_config_mathscr: ..... 938	\um_process_symbol_noparse:nnnn .. ..... 401, <u>426</u>
\um_config_mathsf: ..... 962	\um_process_symbol_parse:nnnn 408, <u>426</u>
\um_config_mathsfrit: ..... 986	\um_remap_symbol:nnn . 403, 410, 435, 437–440, 442, 443, 446–453, 455–458
\um_config_mathsfup: ..... 981	\um_remap_symbol_noparse:nnn 403, <u>434</u>
	\um_remap_symbol_parse:nnn 410, <u>434</u> , 461
	\um_remap_symbols: ..... 417, <u>434</u>
	\um_scan_sscript: 1327, 1360, 1397, 1399
	\um_scan_sscript:TF ..... 1398, 1407
	\um_scanprime: ..... ..... 1265, 1274, 1285, 1289, 1290

<code>\um_scanprime_collect:</code> .....	<code>\um_setup_Latin:</code> ..... 486, <u>795</u>
..... 1267, 1269, 1272, 1275, 1278	<code>\um_setup_latin:</code> ..... 487, <u>802</u>
<code>\um_set_mathalph_range:Nnn</code> ..... <u>689</u>	<code>\um_setup_literals:</code> ..... 484, <u>774</u>
<code>\um_set_mathalph_range:nNnn</code> .....	<code>\um_setup_math_alphabet:n</code> 840–856, 858
..... 689, 699, 702, 705	<code>\um_setup_mathactives:</code> ..... 418, <u>471</u>
<code>\um_set_mathalphabet_char:Nnn</code> ...	<code>\um_setup_mathup:</code> ..... 1450
... 683, 900–908, 913, 916–924,	<code>\um_setup_nabla:</code> ..... 206, 423
929–935, 940–947, 949–951,	<code>\um_setup_partial:</code> ..... 224, 424
955–959, 998, 999, 1009–1029,	<code>\um_setup_shapes:</code> ..... 416, 422
1038, 1041, 1045, 1048,	<code>\um_sub_or_super:n</code> ... 1325, 1358, 1401
1052–1058, 1061–1067, 1069,	<code>\um_symfont_tl</code> ..... 399,
1070, 1083–1091, 1103–1111,	407, 413, 427, 468, 476, 534, 541, 665
1134–1151, 1159–1167, 1174–1182,	<code>\UnicodeMathSymbol</code> ..... 401, 408, 1235
1191–1199, 1207–1215	<code>\unless</code> ..... 605
<code>\um_set_mathalphabet_char:Nnnn</code> .. <u>682</u>	<code>\updefault</code> ..... 414, 872
<code>\um_set_mathalphabet_greek:Nnn</code> ..	<code>\upGreek</code> ..... 9, 44
..... 704, 898,	<code>\upgreek</code> ..... 10, 45
899, 914, 915, 1005–1008, 1044,	<code>\upint</code> ..... 557
1047, 1051, 1060, 1077, 1078,	<code>\upLatin</code> ..... 7, 42
1081, 1082, 1097, 1098, 1101,	<code>\uplatin</code> ..... 8, 43
1102, 1130–1133, 1157, 1158,	<code>\Upsilon</code> ..... 729
1172, 1173, 1189, 1190, 1205, 1206	<code>\upsilon</code> ..... 756
<code>\um_set_mathalphabet_latin:Nnn</code> ..	<code>\use:c</code> ..... 862, 867, 883, 891
... 701, 896, 897, 911, 912, 928,	<code>\use_none:n</code> ..... 411
936, 939, 948, 954, 960, 965–968,	<code>\usepackage</code> ..... 3, 4, 6, 38, 39, 41
972, 973, 976, 977, 983, 984,	
988, 989, 993, 994, 1001–1004,	<b>V</b>
1032, 1034, 1037, 1040, 1075,	<code>\varepsilon</code> ..... 763
1076, 1079, 1080, 1095, 1096,	<code>\varkappa</code> ..... 767
1099, 1100, 1115, 1116, 1120,	<code>\varointclockwise</code> ..... 554
1121, 1126–1129, 1155, 1156,	<code>\varphi</code> ..... 768
1170, 1171, 1187, 1188, 1203, 1204	<code>\varpi</code> ..... 772
<code>\um_set_mathalphabet_numbers:Nnn</code> ..	<code>\varrho</code> ..... 771
... 698, 927, 964, 971, 975, 982,	<code>\varsigma</code> ..... 753
987, 992, 997, 1074, 1094, 1114,	<code>\varTheta</code> ..... 726
1119, 1125, 1154, 1169, 1186, 1202	<code>\vartheta</code> ..... 766
<code>\um_set_mathcode:nnnn</code> <u>355</u> , 468, 549, 663	<code>\vbox</code> ..... 24, 32, 71, 79
<code>\um_setup_active_subscript:nn</code> ...	<code>\version@elt</code> ..... 258
..... 1351, 1365–1392	<code>\version@list</code> ..... 257
<code>\um_setup_active_superscript:nn</code> ..	<code>\voffset</code> ..... 22, 69
..... 1318, 1332–1348	
<code>\um_setup_alphabets:</code> ..... 420, 839	<b>W</b>
<code>\um_setup_alphanum:</code> ..... 419, <u>480</u>	<code>\wd</code> ..... 31, 78
<code>\um_setup_bf_literals:</code> ..... 492, <u>785</u>	
<code>\um_setup_Greek:</code> ..... 488, <u>811</u>	<b>X</b>
<code>\um_setup_greek:</code> ..... 489, <u>820</u>	<code>\xdef</code> ..... 568, 601
	<code>\XeTeXdelcode</code> ..... 337, 342



<code>\XeTeXdelimiter</code> .....	336, 341		<b>Z</b>	
<code>\XeTeXmathaccent</code> .....	346	<code>\Z</code> .....	935, 959	
<code>\XeTeXmathchardef</code> .....	320, 475	<code>\z@</code> .....	1294, 1297, 1298, 1302	
<code>\XeTeXmathcode</code> .....	338, 343, 348, 356	<code>\Zeta</code> .....	714	
<code>\XeTeXmathcodenum</code> .....	478, 1321, 1354	<code>\zeta</code> .....	741	
<code>\XeTeXradical</code> .....	333	<code>\zf@family</code> .....	414	
<code>\Xi</code> .....	722	<code>\zf@fontspec</code> .....	371	
<code>\xi</code> .....	749	<code>\zf@update@ff</code> .....	587, 590	
<code>\XKV@rm</code> .....	384			