# Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2007/01/03 v0.2

## Contents

1 Introduction 1

2 Specification 1
- 2.1 Using multiple fonts 2
- 2.2 Script and scriptscript fonts/features 2

3 Maths input 2

4 Package options 3
- 4.1 Math 'style' 3
- 4.2 Bold switching 4
- 4.3 Other upright vs. italic symbols 4

I The unicode-math package 5

5 Things we need 6
- 5.1 Package options 7
- 5.2 Overcoming \@onlypreamble 9

6 Fundamentals 9
- 6.1 Enlarging the number of maths families 9
- 6.2 \DeclareMathSymbol for unicode ranges 10
- 6.3 User interface to \DeclareSymbolFont 13
- 6.4 Maths alphabets' character mapping 16
- 6.5 (Big) operators 20
- 6.6 Radicals 24
- 6.7 Delimiters 24
- 6.8 Maths accents 27

7 Font features 27
- 7.1 OpenType maths font features 27
- 7.2 Script and scriptscript font options 28
- 7.3 Range processing 28
- 7.4 Resolving Greek letters 30

II Maths alphabets mapping definitions 36
- 7.5 Bold alphabets' character mappings 39

III stix table data extraction 43

**A Documenting maths support in the NFSS 45**
A.1 Overview 45
A.2 Detailed code investigation 46
A.3 Maths symbols 46
A.4 Delimiters 48
A.5 Symbol fonts 51

# 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 XƎTEX, although it is conjectured that small effect needs to be spent to create a cross-format package that would also work with Omega.

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

# 2 Specification

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

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

`\setmathfont[`⟨*font features*⟩`]{`⟨*font name*⟩`}`

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

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

Finally, maths versions must also be provided for. While I guess version selection in LaTeX will remain the same, the specification for choosing the version fonts will probably be an optional argument:

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

Instances above of

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

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

### 2.1 Using multiple fonts

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

`\setmathfont[Range=`⟨*unicode range*⟩`,`⟨*font features*⟩`]{`⟨*font name*⟩`}`

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

### 2.2 Script and scriptscript fonts/features

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

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

## 3 Maths input

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

: TODO: describe alphabet inputs

## 4 Package options

### 4.1 Math 'style'

Classically, 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

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

| | Example | |
|---|---|---|
| Package option | `(a,z,B,X)` | `(α,β,Γ,Ξ)` |
| `math-style=ISO` | $(a,z,B,X)$ | $(\alpha,\beta,\mathit{\Gamma},\mathit{\Xi})$ |
| `math-style=TeX` | $(a,z,B,X)$ | $(\alpha,\beta,\Gamma,\Xi)$ |
| `math-style=French` | $(a,z,\mathrm{B},\mathrm{X})$ | $(\upalpha,\upbeta,\Gamma,\Xi)$ |

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

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

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

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

## 4.2 Bold switching

Similar as in the previous section, ISO standards differ somewhat to TeX's conventions (and classical typesetting) for 'boldness' in mathematics. In the past, it has been customary to use bold *upright* letters to denote things like vectors and matrices. For example, $\mathbf{M} = (M_x, M_y, M_z)$. Presumably, this was due to the rel-

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

| | Example | |
|---|---|---|
| Package option | (a,z,B,X) | (α,β,Γ,Ξ) |
| `bold-style=ISO` | $(\boldsymbol{a},\boldsymbol{z},\boldsymbol{B},\boldsymbol{X})$ | $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Gamma},\boldsymbol{\Xi})$ |
| `bold-style=TeX` | $(\mathbf{a},\mathbf{z},\mathbf{B},\mathbf{X})$ | $(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{\Gamma},\mathbf{\Xi})$ |
| `bold-style=French` | $(\mathbf{a},\mathbf{z},\mathbf{B},\mathbf{X})$ | $(\mathbf{\upalpha},\mathbf{\upbeta},\mathbf{\Gamma},\mathbf{\Xi})$ |

atively scarcity of bold italic fonts in the pre-digital typesetting era. It has been suggested that *italic* bold symbols are used nowadays instead.

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

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

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

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

## 4.3 Other upright vs. italic symbols

**Nabla** The symbol ∇ comes in the six forms shown in table 3. We want an individual option to specify whether we want upright or italic nabla by default (when either upright or italic nabla is used in the source). TeX classically uses an upright nabla, but ISO standards differ (I think). The package options `[Nabla=upright]` and `[Nabla=italic]` switch between the two choices. This is then inherited through `\mathbf`; `\mathit` and `\mathup` can be used to force one way or the other.

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

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

Table 3: The various forms of nabla.

| Description | | Glyph |
|---|---|---|
| Upright | Serif | ∇ |
| | Bold serif | 𝛁 |
| | Bold sans | □ |
| Italic | Serif | 𝛻 |
| | Bold serif | 𝜵 |
| | Bold sans | □ |

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
*abcdefghijklmnopqrstuvwxyz*
*ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩϴ*
*αβγδεζηθικλμνξοπρςστυφχψωϵϑϰϕϱϖ*

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

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
*abcdefghijklmnopqrstuvwxyz*
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ∇ϴ
*αβγδεζηθικλμνξοπρςστυφχψω∂ϵϑϰϕϱϖ*

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
*abcdefghijklmnopqrstuvwxyz*
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ∇ϴ
αβγδεζηθικλμνξοπρςστυφχψω∂ϵϑϰϕϱϖ

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

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

.

***ABCDEFGHIJKLMNOPQRSTUVWXYZ***
***abcdefghijklmnopqrstuvwxyz***
***ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩϴ***
***αβγδεζηθικλμνξοπρςστυφχψωϵϑϰϕϱϖ***

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

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**
**abcdefghijklmnopqrstuvwxyz**
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ∇ϴ**
***αβγδεζηθικλμνξοπρςστυφχψω∂ϵϑϰϕϱϖ***

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

**ABCDEFGHIJKLMNOPQRSTUVWXYZ**
**abcdefghijklmnopqrstuvwxyz**
**ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ∇ϴ**
**αβγδεζηθικλμνξοπρςστυφχψω∂ϵϑϰϕϱϖ**

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

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

## File I

# The unicode-math package

This is the package.

```
\ProvidesPackage{unicode-math}
  [2007/01/03 v0.2 Unicode maths in XeLaTeX]
```

## 5 Things we need

### Packages

```
\RequirePackage{fontspec}
```

### Counters and conditionals

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

For `math-style`:

```
\newif\if@um@literal
\newif\if@um@upGreek
\newif\if@um@upgreek
\newif\if@um@upLatin
\newif\if@um@uplatin
```

For `bold-style`:

```
\newif\if@um@bfliteral
\newif\if@um@bfupGreek
\newif\if@um@bfupgreek
\newif\if@um@bfupLatin
\newif\if@um@bfuplatin
```

For `nabla` and `partial`:

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

### Programming niceties

`\def@cn`

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

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

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

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

```
\long\def\um@FOR#1=[#2:#3]\do#4{%
   #1=#2\relax
   \um@Loop #4%
     \advance#1\@ne
     \ifnum#1>#3\relax
       \expandafter\um@Break
     \fi
   \um@Pool}
```

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

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

### Shortcuts

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

## 5.1 Package options

xkeyval's package support is used here.

### math-style

```
\define@choicekey*{unicode-math.sty}
    {math-style}[\@tempa\@tempb]{iso,tex,french,literal}{%
  \ifcase\@tempb\relax
    \@um@upGreekfalse
    \@um@upgreekfalse
    \@um@upLatinfalse
    \@um@uplatinfalse
    \@um@bfupGreekfalse
    \@um@bfupgreekfalse
    \@um@bfupLatinfalse
    \@um@bfuplatinfalse
    \@um@upNablafalse
    \@um@uppartialfalse
  \or
    \@um@upGreektrue
    \@um@upgreekfalse
    \@um@upLatinfalse
    \@um@uplatinfalse
    \@um@bfupGreektrue
    \@um@bfupgreekfalse
    \@um@bfupLatintrue
    \@um@bfuplatintrue
    \@um@upNablatrue
    \@um@uppartialtrue
  \or
    \@um@upGreektrue
    \@um@upgreektrue
    \@um@upLatintrue
    \@um@uplatinfalse
    \@um@bfupGreektrue
    \@um@bfupgreektrue
    \@um@bfupLatintrue
    \@um@bfuplatintrue
    \@um@upNablatrue
    \@um@uppartialtrue
  \or
    \@um@literaltrue
  \fi}
```

### bold-style

```
\define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,french,literal}{%
  \ifcase\@tempb\relax
    \@um@bfupGreekfalse
    \@um@bfupgreekfalse
    \@um@bfupLatinfalse
    \@um@bfuplatinfalse
  \or
    \@um@bfupGreektrue
    \@um@bfupgreekfalse
    \@um@bfupLatintrue
    \@um@bfuplatintrue
  \or
    \@um@bfupGreektrue
    \@um@bfupgreektrue
    \@um@bfupLatintrue
    \@um@bfuplatintrue
  \or
    \@um@bfliteraltrue
  \fi}
```

```
\define@choicekey*{unicode-math.sty}{Nabla}[\@tempa\@tempb]{upright,italic}{%
  \ifcase\@tempb\relax
    \@um@upNablatrue
  \or
    \@um@upNablafalse
  \fi}
```

```
\define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{%
  \ifcase\@tempb\relax
    \@um@uppartialtrue
  \or
    \@um@uppartialfalse
  \fi}
```

```
\ExecuteOptionsX{math-style=iso}
\ProcessOptionsX
```

### 5.2 Overcoming `\@onlypreamble`

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

```
\def\@preamblecmds{}
```

## 6 Fundamentals

### 6.1 Enlarging the number of maths families

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

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

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

## 6.2 \DeclareMathSymbol for unicode ranges

This is mostly an adaptation from LaTeX's definition.

\DeclareUnicodeMathSymbol

#1 : Symbol, *e.g.*, \alpha or a
#2 : Type, *e.g.*, \mathalpha
#3 : Math font name, *e.g.*, operators
#4 : Slot, *e.g.*, "221E

```
\def\DeclareUnicodeMathSymbol#1#2#3#4{%
```

First ensure the math font (*e.g.*, operators) exists:

```
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
```

No longer need here to perform the obfuscated hex conversion, since \XeTeXmathchar (and friends) has a more simplified input than TeX's \mathchar.

```
    \begingroup
```

The symbol to be defined can be either a command (\alpha) or a character (a). Branch for the former:

```
      \if\relax\noexpand#1% is command?
     \edef\reserved@a{\noexpand\in@{\string\XeTeXmathchar}{\meaning#1}}%
        \reserved@a
```

If the symbol command definition contains \XeTeXmathchar, then we can provide the info that a previous symbol definition is being overwritten:

```
        \ifin@
          \expandafter\um@set@mathsymbol
             \csname sym#3\endcsname#1#2{#4}%
          \@font@info{Redeclaring math symbol \string#1}%
```

Otherwise, overwrite it if the symbol command definition contains plain old \mathchar:

```
        \else
         %\edef\reserved@a{\noexpand\in@{\string\mathchar}{\meaning#1}}%
          %\reserved@a
          %\ifin@
          %  \expandafter\set@xmathsymbol
          %     \csname sym#3\endcsname#1#2{#4}%
```

Otherwise, throw an error if the command name is already taken by a non-symbol definition:

```
          %\else
            %\expandafter\ifx
            %\csname\expandafter\@gobble\string#1\endcsname
            %\relax
              \expandafter\um@set@mathsymbol
                 \csname sym#3\endcsname#1#2{#4}%
            %\else
            %  \@latex@error{Command `\string#1' already defined}\@eha
            %\fi
          %\fi
        \fi
```

And if the symbol input is a character:

```
      \else
        \expandafter\um@set@mathchar
          \csname sym#3\endcsname#1#2{#4}%
      \fi
    \endgroup
```

Everything previous was skipped if the maths font doesn't exist in the first place:

```
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi}
```

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

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

**Operators** 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.

```
  \ifx\mathop#3\relax
    \expandafter\global\expandafter\XeTeXmathchardef
      \csname\expandafter\@gobble\string#2 op\endcsname
      ="\mathchar@type#3 #1 #4\relax
    \gdef#2{%
      \csname\expandafter\@gobble\string#2 op\endcsname
      \expandafter\in@\expandafter#2\expandafter{\um@nolimits}%
      \ifin@\expandafter\nolimits\fi}%
  \else
```

**Radicals**

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

**Delimiters** : TODO: sort out which of these three declarations are necessary!

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

And finally, the general case. We define both the macro and the unicode mathcode for the character.

```
     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
     \global\XeTeXmathchardef#2="\mathchar@type#3 #1 #4\relax
    \fi
   \fi
  \fi
 \fi}
```

`\um@set@mathchar` `#1` : Symbol font number
`#2` : Symbol, *e.g.*, `\alpha` or `a`
`#3` : Type, *e.g.*, `\mathalpha`
`#4` : Slot, *e.g.*, `"221E`
Or if it's for a character:

```
\def\um@set@mathchar#1#2#3#4{%
 \global\XeTeXmathcode`#2="\mathchar@type#3 #1 #4\relax}
```

∞

```
\zf@fontspec{}{Cambria Math}
\let\glb@currsize\relax
\DeclareSymbolFont{test}{EU1}{\zf@family}{m}{n}
\DeclareUnicodeMathSymbol{\infinity}{\mathord}{test}{"221E}
$\infinity$
```

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

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

$$A$$

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

## 6.3 User interface to \DeclareSymbolFont

Here's the simplest usage:

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

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

And an example of the Range feature:

$(a, a, \boldsymbol{a}, \mathbf{a}, \alpha)$ $(a, a, \boldsymbol{a}, \mathbf{a}, \square)$

```
\setmathfont{Cambria Math}
$(a, \ita, \mathbf{a}, \bfa, \alpha)$
\setmathfont[Range={"2133-"2135,\alpha}]{Lucida Sans}
$(a, \ita, \mathbf{a}, \bfa, \alpha)$
```

A less useful (perhaps) example of the Range feature:

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

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

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

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

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

### Init

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

```
      \let\glb@currsize\relax
```

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

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

- Tell fontspec that maths font features are actually allowed.

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

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

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

- Set the name of the math version being defined

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

Define default font features for the script and scriptscript font.

```
  \def\um@ScriptFeatures{ScriptStyle}%
  \def\um@ScriptScriptFeatures{ScriptScriptStyle}%
  \def\um@ScriptFont{#2}%
  \def\um@ScriptScriptFont{#2}%
```

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

Probably in the future we want options to change the hard-coded fontspec maths-related features.

```
  \setkeys*[um]{options}{#1}%
%\rule{1ex}{1ex}%
  \edef\@tempa{\noexpand\zf@fontspec{%
    Script=Math,SizeFeatures={%
      {Size=\tf@size-},%
      {Size=\sf@size-\tf@size,%
        Font=\um@ScriptFont,%
        \um@ScriptFeatures},%
      {Size=-\sf@size,%
        Font=\um@ScriptScriptFont,%
        \um@ScriptScriptFeatures}},%
    \XKV@rm}{#2}}\@tempa
%\rule{1ex}{1ex}%
```

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

```
  \ifx\um@char@range\@empty
    \def\um@symfont{um@allsym}%
    \um@PackageInfo{Defining the default maths font as '#2'}%
    \let\UnicodeMathSymbol\um@mathsymbol@noparse
  \else
    \stepcounter{um@fam}%
    \edef\um@symfont{um@fam\theum@fam}%
    \let\UnicodeMathSymbol\um@mathsymbol@parse
  \fi
%\rule{1ex}{1ex}%
  \DeclareSymbolFont{\um@symfont}
    {\encodingdefault}{\zf@family}{\mddefault}{\updefault}%
```

And now we input every single maths char. See File III for the source to unicode-math.tex. There is an added space somewhere in here.

```
  \if@um@init
    \input unicode-math.tex
    \input unicode-math-add.tex
  \fi
  \unless\ifx\um@char@range\@empty
    \input unicode-math.tex
    \input unicode-math-add.tex
  \fi

%\rule{1ex}{1ex}%
  \ifx\um@char@range\@empty
    \let\um@mathbb\@empty
    \let\um@mathbf\@empty
    \let\um@mathfrak\@empty
    \let\um@mathup\@empty
    \let\um@mathit\@empty
    \let\um@mathscr\@empty
    \let\um@mathsf\@empty
    \let\um@mathsfit\@empty
    \let\um@mathtt\@empty
    \let\um@mathbf\@empty
    \let\um@mathbfit\@empty
    \let\um@mathbffrak\@empty
    \let\um@mathbfscr\@empty
    \let\um@mathbfsf\@empty
    \let\um@mathbfsfit\@empty
    \let\SetMathAlphChar\um@mathmap@noparse
  \else
    \let\SetMathAlphChar\um@mathmap@parse
```

```
\fi
```

## 6.4 Maths alphabets' character mapping

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

Numbers:

```
\ifx\um@char@range\@empty
  \um@def@numbers
```

### Normal weight

```
  \if@um@literal
    \um@setmathcodes{\um@usv@upLatin}{26}{\um@usv@upLatin}%
    \um@setmathcodes{\um@usv@itLatin}{26}{\um@usv@itLatin}%
    \um@setmathcodes{\um@usv@itlatin}{26}{\um@usv@itlatin}%
    \SetMathCode{\um@usv@ith}{\mathalpha}{\um@symfont}{\um@usv@ith}%
    \um@setmathcodes{\um@usv@uplatin}{26}{\um@usv@uplatin}%
    \um@setmathcodes{\um@usv@upGreek}{25}{\um@usv@upGreek}%
  \SetMathCode{\um@usv@varTheta}{\mathalpha}{\um@symfont}{\um@usv@varTheta}%
    \um@setmathcodes{\um@usv@itGreek}{25}{\um@usv@itGreek}%
    \um@setmathcodes{\um@usv@upgreek}{25}{\um@usv@upgreek}%
```

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

```
  \SetMathCode{\um@usv@Nabla}{\mathalpha}{\um@symfont}{\um@usv@Nabla}%
  \SetMathCode{\um@usv@itNabla}{\mathalpha}{\um@symfont}{\um@usv@itNabla}%
  \SetMathCode{\um@usv@partial}{\mathalpha}{\um@symfont}{\um@usv@partial}%
  \SetMathCode{\um@usv@itpartial}{\mathalpha}{\um@symfont}{\um@usv@itpartial}%
  \else
```

Latin letters:

```
    \if@um@upLatin\um@def@upLatin\else\um@def@itLatin\fi
    \if@um@uplatin\um@def@uplatin\else\um@def@itlatin\fi
```

0123456789
*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
*abcdefghijklmnopqrstuvwxyz*

```
\setmathfont{Cambria Math}
$0123456789$ \\
$ABCDEFGHIJKLMNOPQRSTUVWXYZ$ \\
$abcdefghijklmnopqrstuvwxyz$ \\
```

Normal weight Greek, italic uppercase and lowercase respectively:

```
    \if@um@upGreek\um@def@upGreek\else\um@def@itGreek\fi
    \if@um@upgreek\um@def@upgreek\else\um@def@itgreek\fi
```

Nabla and partial:

```
    \if@um@upNabla
   \SetMathCode{\um@usv@Nabla}{\mathalpha}{\um@symfont}{\um@usv@Nabla}%
   \SetMathCode{\um@usv@itNabla}{\mathalpha}{\um@symfont}{\um@usv@Nabla}%
    \else
   \SetMathCode{\um@usv@Nabla}{\mathalpha}{\um@symfont}{\um@usv@itNabla}%
   \SetMathCode{\um@usv@itNabla}{\mathalpha}{\um@symfont}{\um@usv@itNabla}%
    \fi
    \if@um@uppartial
   \SetMathCode{\um@usv@partial}{\mathalpha}{\um@symfont}{\um@usv@partial}%
   \SetMathCode{\um@usv@itpartial}{\mathalpha}{\um@symfont}{\um@usv@partial}%
    \else
   \SetMathCode{\um@usv@partial}{\mathalpha}{\um@symfont}{\um@usv@itpartial}%
   \SetMathCode{\um@usv@itpartial}{\mathalpha}{\um@symfont}{\um@usv@itpartial}%
    \fi
  \fi
```

**Bold** : TODO: finish this off for the ‘var’ symbols.

```
  \if@um@bfliteral
    \um@setmathcodes{\um@usv@bfLatin}{26}{\um@usv@bfLatin}
    \um@setmathcodes{\um@usv@bflatin}{26}{\um@usv@bflatin}
    \um@setmathcodes{\um@usv@bfitLatin}{26}{\um@usv@bfitLatin}
    \um@setmathcodes{\um@usv@bfitlatin}{26}{\um@usv@bfitlatin}
    \um@setmathcodes{\um@usv@bfGreek}{25}{\um@usv@bfGreek}
    \um@setmathcodes{\um@usv@bfgreek}{25}{\um@usv@bfgreek}
    \um@setmathcodes{\um@usv@bfitGreek}{25}{\um@usv@bfitGreek}
    \um@setmathcodes{\um@usv@bfitgreek}{25}{\um@usv@bfitgreek}
  \else
    \um@setmathcodes{\um@usv@bfLatin,\um@usv@bfitLatin}{26}{%
      \if@um@bfupLatin\um@usv@bfLatin\else\um@usv@bfitLatin\fi}
    \um@setmathcodes{\um@usv@bflatin,\um@usv@bfitlatin}{26}{%
      \if@um@bfupLatin\um@usv@bflatin\else\um@usv@bfitlatin\fi}
    \um@setmathcodes{\um@usv@bfGreek,\um@usv@bfitGreek}{25}{%
      \if@um@bfupLatin\um@usv@bfGreek\else\um@usv@bfitGreek\fi}
    \um@setmathcodes{\um@usv@bfGreek,\um@usv@bfitGreek}{25}{%
      \if@um@bfupLatin\um@usv@bfGreek\else\um@usv@bfitGreek\fi}
```

Bold nabla and partial symbols:

```
    \if@um@upNabla
   \SetMathCode{\um@usv@bfNabla}{\mathalpha}{\um@symfont}{\um@usv@bfNabla}%
   \SetMathCode{\um@usv@bfitNabla}{\mathalpha}{\um@symfont}{\um@usv@bfNabla}%
   \SetMathCode{\um@usv@bfsfNabla}{\mathalpha}{\um@symfont}{\um@usv@bfsfNabla}%
   \SetMathCode{\um@usv@bfsfitNabla}{\mathalpha}{\um@symfont}{\um@usv@bfsfNabla}%
    \else
   \SetMathCode{\um@usv@bfNabla}{\mathalpha}{\um@symfont}{\um@usv@bfitNabla}%
   \SetMathCode{\um@usv@bfitNabla}{\mathalpha}{\um@symfont}{\um@usv@bfitNabla}%
   \SetMathCode{\um@usv@bfsfitNabla}{\mathalpha}{\um@symfont}{\um@usv@bfsfitNabla}%
   \SetMathCode{\um@usv@bfsfitNabla}{\mathalpha}{\um@symfont}{\um@usv@bfsfitNabla}%
    \fi
```

```
      \if@um@uppartial
     \SetMathCode{\um@usv@bfpartial}{\mathalpha}{\um@symfont}{\um@usv@bfpartial}%
     \SetMathCode{\um@usv@bfitpartial}{\mathalpha}{\um@symfont}{\um@usv@bfpartial}%
     \SetMathCode{\um@usv@bfsfpartial}{\mathalpha}{\um@symfont}{\um@usv@bfsfpartial}%
     \SetMathCode{\um@usv@bfsfitpartial}{\mathalpha}{\um@symfont}{\um@usv@bfsfpartial}%
      \else
     \SetMathCode{\um@usv@bfpartial}{\mathalpha}{\um@symfont}{\um@usv@bfitpartial}%
     \SetMathCode{\um@usv@bfitpartial}{\mathalpha}{\um@symfont}{\um@usv@bfitpartial}%
     \SetMathCode{\um@usv@bfsfpartial}{\mathalpha}{\um@symfont}{\um@usv@bfsfitpartial}%
     \SetMathCode{\um@usv@bfsfitpartial}{\mathalpha}{\um@symfont}{\um@usv@bfsfitpartial}%
      \fi
    \fi
```

: TODO: implement behaviour when char@range is NOT empty

```
  \fi
```

*ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ   ϴ*
*αβγδεζηθικλμνξοπρστυφχψω   ϵϑϰϕϱϖ*

```
\setmathfont{Cambria Math}
$ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ$\quad$□$ \\
$αβγδεζηθικλμνξοπρστυφχψω$\quad$□ϑϰϕϱϖ$ \\
```

Set up the maths alphabets:

```
  \um@setup@alphabets
```

End of the \setmathfont macro.

```
}
```

\um@mathsymbol@noparse

```
\newcommand\um@mathsymbol@noparse[4]{%
  \DeclareUnicodeMathSymbol{#2}{#3}{\um@symfont}{#1}}
```

\um@mathsymbol@parse

If the Range font feature has been used, then only a subset of the unicode glyphs are to be defined. See section §7.3 for the code that enables this.

```
\newcommand\um@mathsymbol@parse[4]{%
  \um@parse@term{#1}{#2}{#3}{%
    %\um@PackageInfo{Defining \string#2 as mathchar #1}%
    \DeclareUnicodeMathSymbol{#2}{#3}{\um@symfont}{#1}}}
```

\um@mk@alph

Wrapper to define maths alphabets.

```
\newcommand\um@mk@math[1]{%
  \expandafter\def\csname math#1\endcsname##1{%
    \begingroup
      \csname um@math#1\endcsname
      ##1
    \endgroup}}
```

Maths alphabets' base definition. See section §6.4 for the internal definitions.

```
\um@mk@math{up}
```

```
\um@mk@math{it}
\um@mk@math{scr}
\um@mk@math{bb}
\um@mk@math{frak}
\um@mk@math{sf}
\um@mk@math{sfit}
\um@mk@math{tt}
```

And bold maths alphabets. See section §7.5 for the internal definitions.

```
\um@mk@math{bf}
\um@mk@math{bfit}
\um@mk@math{bfscr}
\um@mk@math{bffrak}
\um@mk@math{bfsf}
\um@mk@math{bfsfit}
```

`\mathcal`

```
\let\mathcal\mathscr
```

`\um@mathmap@noparse`

`#1` : Maths alphabet, *e.g.*, `\mathbb`<br>
`#2` : Input slot(s), *e.g.*, the slot for 'A' (comma separated)<br>
`#3` : Output slot, *e.g.*, the slot for '𝔸'

Adds `\SetMathCode` declaractions 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.

```
\newcommand\um@mathmap@noparse[3]{%
  \@for\@ii:=#2\do{%
    \expandafter\expandafter
    \expandafter\um@addto@mathmap
    \expandafter\expandafter
    \expandafter{%
    \expandafter\um@symfont
    \expandafter}%
    \expandafter{\@ii}{#1}{#3}%
   }}%
```

`\um@mathmap@parse`

`#1` : Maths alphabet, *e.g.*, `\mathbb`<br>
`#2` : Input slot(s), *e.g.*, the slot for 'A' (comma separated)<br>
`#3` : Output slot, *e.g.*, the slot for '𝔸'

When `\um@parse@term` is executed, it populates the `\um@char@num@range` macro with slot numbers corresponding to the specified range. This range is used to conditionally add `\SetMathCode` declaractions to the maths alphabet definition (*e.g.*, `\um@mathscr`).

```
\newcommand\um@mathmap@parse[3]{%
  \@for\@ii:=\um@char@num@range\do{%
    \ifnum\@ii=#3\relax
```

```
      \@for\@jj:=#2\do{
        \expandafter\expandafter
        \expandafter\um@addto@mathmap
        \expandafter\expandafter
        \expandafter{%
        \expandafter\um@symfont
        \expandafter}%
        \expandafter{\@jj}{#1}{#3}}%
    \fi}}%
```

\um@addto@mathmap

`#1` : Math symbol font, always/usually the expansion of `\um@symfont`
`#2` : Input slot, *e.g.*, the slot for 'A'
`#3` : Maths alphabet, *e.g.*, `\mathbb`
`#4` : Output slot, *e.g.*, the slot for '𝔸'

This macro is used so that `\um@symfont` can be expanded before entering the `\g@addto@macro` command.

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

## 6.5 (Big) operators

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

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

: TODO use `\mathchar "8000` to create active operators that have `\nolimits` suffices.

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

| USV |
|---|
| U+0220F |
| U+02210 |

U+02211

U+0222B

U+0222C

U+0222D

U+0222E

U+0222F

U+02230

U+02231

U+02232

U+02233

U+022C0

U+022C1

U+022C2

U+022C3

U+027D5

U+027D6

U+027D7

U+027D8

U+027D9

U+029F8

U+029F9

U+02A00

U+02A01

U+02A02

U+02A03

U+02A04

U+02A05

U+02A06

U+02A07

U+02A08

U+02A09

U+02A0B

U+02A0C

U+02A0D

U+02A0E

U+02A0F

U+02A10

U+02A11

U+02A12

U+02A13

U+02A14

U+02A15

U+02A16

U+02A17

U+02A18

U+02A19

U+02A1A

U+02A1B

U+02A1C

U+02A1D

U+02A1E

U+02A1F

U+02A20

U+02A21

U+02AFC

U+02AFF

\um@nolimits This macro is a commalist containing those maths operators that require a \nolimits suffix. This list is used when processing unicode-math.tex to define such commands automatically (see the macro \um@set@mathsymbol on page 11). 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.

```
\def\um@nolimits{%
 \@elt\int\@elt\iint\@elt\iiint\@elt\iiiint\@elt\oint\@elt\oiint\@elt\oiiint
 \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrclockwise\@elt\sumint
 \@elt\intbar\@elt\intBar\@elt\fint\@elt\cirfnint\@elt\awint\@elt\rppolint
 \@elt\scpolint\@elt\npolint\@elt\pointint\@elt\sqint\@elt\intlarhk\@elt\intx
  \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint}
```

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

```
\newcommand\addnolimits[1]{%
  \expandafter\def
  \expandafter\um@nolimits
  \expandafter{\um@nolimits\@elt#1}}
```

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

```
\def\removenolimits#1{%
  \begingroup
    \def\@elt##1{%
      \ifx##1#1\else
        \noexpand\@elt\noexpand##1
     \fi}
    \xdef\um@nolimits{\um@nolimits}%
  \endgroup}
```

$$\iiint_V$$

$$\iiint\limits_V$$

$$\iiint_V$$

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

## 6.6 Radicals

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

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

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

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

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

## 6.7 Delimiters

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

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

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

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

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

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

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

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

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

Here are all `\mathopen` characters:

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

|  |  |  |  |
|---|---|---|---|
| U+0298B | ⍰ | `\lbrackubar` | LEFT SQUARE BRACKET WITH UNDERBAR |
| U+0298D | ⍰ | `\lbrackultick` | LEFT SQUARE BRACKET WITH TICK IN TOP CORNER |
| U+0298F | ⍰ | `\lbracklltick` | LEFT SQUARE BRACKET WITH TICK IN BOTTOM CORNER |
| U+02991 | ⍰ | `\langledot` | LEFT ANGLE BRACKET WITH DOT |
| U+02993 | ⍰ | `\lparenless` | LEFT ARC LESS-THAN BRACKET |
| U+02997 | ⍰ | `\lblkbrbrak` | LEFT BLACK TORTOISE SHELL BRACKET |
| U+029D8 | ⍰ | `\lvzigzag` | LEFT WIGGLY FENCE |
| U+029DA | ⍰ | `\Lvzigzag` | LEFT DOUBLE WIGGLY FENCE |
| U+029FC | ⧼ | `\lcurvyangle` | LEFT POINTING CURVED ANGLE BRACKET |
| U+03014 | 〔 | `\lbrbrak` | LEFT BROKEN BRACKET |
| U+03018 | ⍰ | `\Lbrbrak` | LEFT WHITE TORTOISE SHELL BRACKET |

And `\mathclose`:

| USV | Ex. | Macro | Description |
|---|---|---|---|
| U+00029 | ) | `\rparen` | RIGHT PARENTHESIS |
| U+0005D | ] | `\rbrack` | RIGHT SQUARE BRACKET |
| U+0007D | } | `\rbrace` | RIGHT CURLY BRACKET |
| U+000BB | » | `\guillemotright` | DOUBLE ANGLE QUOTATION MARK (GUILLEMET), RIGHT |
| U+02019 | ’ | `\rq` | SINGLE QUOTATION MARK, RIGHT |
| U+0201B | ⍰ | `\quotsinglright` | RISING SINGLE QUOTE, RIGHT (HIGH) |
| U+0201F | ⍰ | `\quotdblright` | RISING DOUBLE QUOTE, RIGHT (HIGH) |
| U+0203A | › | `\guilsinglright` | SINGLE ANGLE QUOTATION MARK (GUILLEMET), RIGHT |
| U+02309 | ⌉ | `\rceil` | RIGHT CEILING |
| U+0230B | ⌋ | `\rfloor` | RIGHT FLOOR |
| U+0231D | ⍰ | `\urcorner` | UPPER RIGHT CORNER |
| U+0231F | ⍰ | `\lrcorner` | LOWER RIGHT CORNER |
| U+02773 | ❳ | `\rbrbrak` | LIGHT RIGHT TORTOISE SHELL BRACKET ORNAMENT |
| U+027C6 | ⍰ | `\rbag` | RIGHT S-SHAPED BAG DELIMITER |
| U+027E7 | ⟧ | `\rBrack` | MATHEMATICAL RIGHT WHITE SQUARE BRACKET |
| U+027E9 | ⟩ | `\rangle` | MATHEMATICAL RIGHT ANGLE BRACKET |
| U+027EB | ⟫ | `\rAngle` | MATHEMATICAL RIGHT DOUBLE ANGLE BRACKET |
| U+027ED | ⍰ | `\Rbrbrak` | MATHEMATICAL RIGHT WHITE TORTOISE SHELL BRACKET |
| U+02984 | ⍰ | `\rBrace` | RIGHT WHITE CURLY BRACKET |

| | | | |
|---|---|---|---|
| U+02986 | ⍰ | `\rParen` | RIGHT WHITE PARENTHESIS |
| U+02988 | ⍰ | `\rrparenthesis` | Z NOTATION RIGHT IMAGE BRACKET |
| U+0298A | ⍰ | `\rrangle` | Z NOTATION RIGHT BINDING BRACKET |
| U+0298C | ⍰ | `\rbrackubar` | RIGHT SQUARE BRACKET WITH UNDERBAR |
| U+0298E | ⍰ | `\rbracklrtick` | RIGHT SQUARE BRACKET WITH TICK IN BOTTOM CORNER |
| U+02990 | ⍰ | `\rbrackurtick` | RIGHT SQUARE BRACKET WITH TICK IN TOP CORNER |
| U+02992 | ⍰ | `\rangledot` | RIGHT ANGLE BRACKET WITH DOT |
| U+02994 | ⍰ | `\rparengtr` | RIGHT ARC GREATER-THAN BRACKET |
| U+02998 | ⍰ | `\rblkbrbrak` | RIGHT BLACK TORTOISE SHELL BRACKET |
| U+029D9 | ⍰ | `\rvzigzag` | RIGHT WIGGLY FENCE |
| U+029DB | ⍰ | `\Rvzigzag` | RIGHT DOUBLE WIGGLY FENCE |
| U+029FD | ⧽ | `\rcurvyangle` | RIGHT POINTING CURVED ANGLE BRACKET |
| U+03015 | 〕 | `\rbrbrak` | RIGHT BROKEN BRACKET |
| U+03019 | ⍰ | `\Rbrbrak` | RIGHT WHITE TORTOISE SHELL BRACKET |

## 6.8 Maths accents

: TODO: maths accents

# 7 Font features

`\um@zf@feature`

Use the same method as fontspec for feature definition (*i.e.*, using xkeyval) but with a conditional to restrict the scope of these features to unicode-math commands.

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

## 7.1 OpenType maths font features

```
\um@zf@feature{ScriptStyle}{%
  \zf@update@ff{+ssty=0}}
```

```
\um@zf@feature{ScriptScriptStyle}{%
  \zf@update@ff{+ssty=1}}
```

## 7.2 Script and scriptscript font options

```
\define@cmdkey[um]{options}[um@]{ScriptFeatures}{}
\define@cmdkey[um]{options}[um@]{ScriptScriptFeatures}{}
\define@cmdkey[um]{options}[um@]{ScriptFont}{}
\define@cmdkey[um]{options}[um@]{ScriptScriptFont}{}
```

## 7.3 Range processing

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

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

`\um@parse@term`

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

This macro expands to `#4`[1] 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 7.

Table 7: Ranges accepted by `\um@parse@range`.

| Input | Range |
|---|---|
| `x` | □ = □ |
| `x-` | □ ≥ □ |
| `-y` | □ ≤ □ |
| `x-y` | □ ≤ □ ≤ □ |

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

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

[1] Unless I've got my terminology twisted again.

`\if\relax\noexpand##` is true if `##` is a control sequence; then match to either the character macro (`\alpha`) or the math type (`\mathbin`):

```
      \expandafter\if\expandafter\relax\expandafter\noexpand\@ii
        \expandafter\ifx\@ii#2
          \@tempswatrue
        \else
          \expandafter\ifx\@ii#3
            \@tempswatrue
          \fi
        \fi
```

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

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

```
      \if@tempswa
        \ifx\um@char@num@range\@empty
          \g@addto@macro\um@char@num@range{#1}%
        \else
          \g@addto@macro\um@char@num@range{,#1}%
        \fi
        #4%
      \fi
    \fi}}
```

‘1’ or ‘\a’ or ‘\b’ is included ‘1’ or ‘\b’ or ‘\c’ is included ‘3’ or ‘\a’ or ‘\b’ is included‘3’ or ‘\a’ or ‘\b’ is included

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

`\um@parse@range` Weird syntax. As shown previously in table 7, this macro can be passed four different input types via `\um@parse@term`.

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

| | |
|---|---|
| Range | □ = □ |
| C-list input | `\@ii=X` |
| Macro input | `\um@parse@range X-\@marker-\@nil#1\@nil` |
| Arguments | `#1-#2-#3` = `X-\@marker-{}` |

```
  \ifx\@marker\@tempb\relax
```

```
    \ifnum#4=#1\relax
      \@tempswatrue
    \fi
  \else
```

| | |
|---|---|
| Range | □ ≥ □ |
| C-list input | \@ii=X- |
| Macro input | \um@parse@range X--\@marker-\@nil#1\@nil |
| Arguments | #1-#2-#3 = X-{}-\@marker- |

```
    \ifx\@empty\@tempb
      \ifnum#4>\numexpr#1-1\relax
        \@tempswatrue
      \fi
    \else
```

| | |
|---|---|
| Range | □ ≤ □ |
| C-list input | \@ii=-Y |
| Macro input | \um@parse@range -Y-\@marker-\@nil#1\@nil |
| Arguments | #1-#2-#3 = {}-Y-\@marker- |

```
      \ifx\@empty\@tempa
        \ifnum#4<\numexpr#2+1\relax
          \@tempswatrue
        \fi
```

| | |
|---|---|
| Range | □ ≤ □ ≤ □ |
| C-list input | \@ii=X-Y |
| Macro input | \um@parse@range X-Y-\@marker-\@nil#1\@nil |
| Arguments | #1-#2-#3 = X-Y-\@marker- |

```
      \else
        \ifnum#4>\numexpr#1-1\relax
          \ifnum#4<\numexpr#2+1\relax
            \@tempswatrue
          \fi\fi\fi\fi\fi}
```

## 7.4 Resolving Greek letters

**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.

```
\def\um@usv@num{`\0}
\def\um@usv@upLatin{`\A}
\def\um@usv@uplatin{`\a}
\def\um@usv@itLatin{"1D434}
\def\um@usv@itlatin{"1D44E}
\def\um@usv@upGreek{"391}
\def\um@usv@upgreek{"3B1}
\def\um@usv@itGreek{"1D6E2}
\def\um@usv@itgreek{"1D6FC}
```

```
\def\um@usv@bbnum{"1D7D8}
\def\um@usv@bbLatin{"1D538}
\def\um@usv@bblatin{"1D552}
\def\um@usv@scrLatin{"1D434}
\def\um@usv@scrlatin{"1D4B6}
\def\um@usv@frakLatin{"1D504}
\def\um@usv@fraklatin{"1D51E}
\def\um@usv@sfnum{"1D7E2}
\def\um@usv@sfLatin{"1D5A0}
\def\um@usv@sflatin{"1D5BA}
\def\um@usv@sfitLatin{"1D608}
\def\um@usv@sfitlatin{"1D622}
\def\um@usv@ttnum{"1D7F6}
\def\um@usv@ttLatin{"1D670}
\def\um@usv@ttlatin{"1D68A}
```

Bold:

```
\def\um@usv@bfnum{"1D7CE}
\def\um@usv@bfLatin{"1D400}
\def\um@usv@bflatin{"1D41A}
\def\um@usv@bfGreek{"1D6A8}
\def\um@usv@bfgreek{"1D6C2}
\def\um@usv@bfitLatin{"1D468}
\def\um@usv@bfitlatin{"1D482}
\def\um@usv@bfitGreek{"1D71C}
\def\um@usv@bfitgreek{"1D736}
\def\um@usv@bffrakLatin{"1D56C}
\def\um@usv@bffraklatin{"1D586}
\def\um@usv@bfscrLatin{"1D4D0}
\def\um@usv@bfscrlatin{"1D4EA}
\def\um@usv@bfsfnum{"1D7EC}
\def\um@usv@bfsfLatin{"1D5D4}
\def\um@usv@bfsflatin{"1D5EE}
\def\um@usv@bfsfGreek{"1D756}
\def\um@usv@bfsfgreek{"1D770}
\def\um@usv@bfsfitLatin{"1D63C}
\def\um@usv@bfsfitlatin{"1D656}
\def\um@usv@bfsfitGreek{"1D790}
\def\um@usv@bfsfitgreek{"1D7AA}
```

Greek variants:

```
\def\um@usv@varTheta{"3F4}
\def\um@usv@Digamma{"3DC}
\def\um@usv@varepsilon{"3F5}
\def\um@usv@vartheta{"3D1}
\def\um@usv@varkappa{"3F0}
\def\um@usv@varphi{"3D5}
\def\um@usv@varrho{"3F1}
```

```
\def\um@usv@varpi{"3D6}
\def\um@usv@digamma{"3DD}
```

Italic Greek variants:

```
\def\um@usv@ith{"210E}
\def\um@usv@itvarTheta{"1D6F3}
\def\um@usv@itvarepsilon{"1D716}
\def\um@usv@itvartheta{"1D717}
\def\um@usv@itvarkappa{"1D718}
\def\um@usv@itvarphi{"1D719}
\def\um@usv@itvarrho{"1D71A}
\def\um@usv@itvarpi{"1D71B}
```

Nabla:

```
\def\um@usv@Nabla{"2207}
\def\um@usv@itNabla{"1D6FB}
\def\um@usv@bfNabla{"1D6C1}
\def\um@usv@bfitNabla{"1D735}
\def\um@usv@bfsfNabla{"1D76F}
\def\um@usv@bfsfitNabla{"1D7A9}
```

Partial:

```
\def\um@usv@partial{"2202}
\def\um@usv@itpartial{"1D715}
\def\um@usv@bfpartial{"1D6DB}
\def\um@usv@bfitpartial{"1D74F}
\def\um@usv@bfsfpartial{"1D789}
\def\um@usv@bfsfitpartial{"1D7C3}
```

`\um@setmathcodes`

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

```
\newcommand\um@setmathcodes[3]{%
  \@for\um@inp:=#1\do{%
    \um@FOR\@tempcnta=[1:#2]\do{%
      \SetMathCode{\numexpr\um@inp+\@tempcnta-1\relax}
        {\mathalpha}{\um@symfont}{\numexpr#3+\@tempcnta-1\relax}}}}
```

`\um@setmathcodes`

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

```
\newcommand\um@setmathalphs[4]{%
  \@for\um@inp:=#2\do{%
    \um@FOR\@tempcnta=[1:#3]\do{%
```

```
        \edef\@tempa{%
      \noexpand\SetMathAlphChar{\noexpand#1}{\number\numexpr\um@inp+\@tempcnta-1\relax}
            {\number\numexpr#4+\@tempcnta-1\relax}}\@tempa}}
```

BCDBCD □ □ □ □ □ □

```
{\um@setmathcodes{`\A,`\D}{3}{`\B}
$ABCDEF$} $ABCDEF$
```

`\um@resolve@greek` This macro defines `\Alpha`…`\omega` as their corresponding unicode (mathematical italic) character. Remember that the mapping to upright or italic happens with the mathcode definitions, whereas these macros just stand for the literal unicode characters.

```
\AtBeginDocument{\um@resolve@greek}%
\newcommand\um@resolve@greek{%
  \def\Alpha{\itAlpha}%
  \def\Beta{\itBeta}%
  \def\Gamma{\itGamma}%
  \def\Delta{\itDelta}%
  \def\Epsilon{\itEpsilon}%
  \def\Zeta{\itZeta}%
  \def\Eta{\itEta}%
  \def\Theta{\itTheta}%
  \def\Iota{\itIota}%
  \def\Kappa{\itKappa}%
  \def\Lambda{\itLambda}%
  \def\Mu{\itMu}%
  \def\Nu{\itNu}%
  \def\Xi{\itXi}%
  \def\Omicron{\itOmicron}%
  \def\Pi{\itPi}%
  \def\Rho{\itRho}%
  \def\varTheta{\itvarTheta}%
  \def\Sigma{\itSigma}%
  \def\Tau{\itTau}%
  \def\Upsilon{\itUpsilon}%
  \def\Phi{\itPhi}%
  \def\Chi{\itChi}%
  \def\Psi{\itPsi}%
  \def\Omega{\itOmega}%
```

Lowercase:

```
  \def\alpha{\italpha}%
  \def\beta{\itbeta}%
  \def\gamma{\itgamma}%
  \def\delta{\itdelta}%
  \def\varepsilon{\itvarepsilon}%
  \def\zeta{\itzeta}%
  \def\eta{\iteta}%
  \def\theta{\ittheta}%
  \def\iota{\itiota}%
  \def\kappa{\itkappa}%
  \def\lambda{\itlambda}%
  \def\mu{\itmu}%
  \def\nu{\itnu}%
  \def\xi{\itxi}%
  \def\omicron{\itomicron}%
  \def\pi{\itpi}%
  \def\rho{\itrho}%
  \def\varsigma{\itvarsigma}%
  \def\sigma{\itsigma}%
  \def\tau{\ittau}%
  \def\upsilon{\itupsilon}%
  \def\phi{\itphi}%
  \def\chi{\itchi}%
  \def\psi{\itpsi}%
  \def\omega{\itomega}%
  \def\partial{\uppartial}%
  \def\varepsilon{\itvarepsilon}%
  \def\vartheta{\itvartheta}%
  \def\varkappa{\itvarkappa}%
  \def\varphi{\itvarphi}%
  \def\varrho{\itvarrho}%
  \def\varrho{\itvarsigma}%
  \def\varpi{\itvarpi}}
```

\um@def@numbers

```
\newcommand\um@def@numbers{%
  \um@setmathcodes{\um@usv@num}{10}{\um@usv@num}}
```

\um@def@upLatin

```
\newcommand\um@def@upLatin{%
   \um@setmathcodes{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@upLatin}}
```

\um@def@itLatin

```
\newcommand\um@def@itLatin{%
   \um@setmathcodes{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@itLatin}}
```

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

```
\newcommand\um@def@itlatin{%
```

```
 \um@setmathcodes{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@itlatin}%
  \SetMathCode{`\h}{\mathalpha}{\um@symfont}{\um@usv@ith}%
  \SetMathCode{\um@usv@ith}{\mathalpha}{\um@symfont}{\um@usv@ith}%
}
```

\um@def@uplatin

```
\newcommand\um@def@uplatin{%
 \um@setmathcodes{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@uplatin}%
  \SetMathCode{\um@usv@ith}{\mathalpha}{\um@symfont}{`\h}%
}
```

\um@def@upGreek

```
\newcommand\um@def@upGreek{%
   \um@setmathcodes{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@upGreek}%
   \SetMathCode{\um@usv@varTheta}{\mathalpha}{\um@symfont}{\um@usv@varTheta}%
    \SetMathCode{"1D6F3}{\mathalpha}{\um@symfont}{\um@usv@varTheta}%
}
```

\um@def@itGreek

```
\newcommand\um@def@itGreek{%
   \um@setmathcodes{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@itGreek}%
    \SetMathCode{\um@usv@varTheta}{\mathalpha}{\um@symfont}{"1D6F3}%
}
```

\um@def@upgreek

```
\newcommand\um@def@upgreek{%
   \um@setmathcodes{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@upgreek}%
   \SetMathCode{\um@usv@varepsilon}{\mathalpha}{\um@symfont}{\um@usv@varepsilon}%
   \SetMathCode{\um@usv@vartheta}{\mathalpha}{\um@symfont}{\um@usv@vartheta}%
   \SetMathCode{\um@usv@varkappa}{\mathalpha}{\um@symfont}{\um@usv@varkappa}%
   \SetMathCode{\um@usv@varphi}{\mathalpha}{\um@symfont}{\um@usv@varphi}%
   \SetMathCode{\um@usv@varrho}{\mathalpha}{\um@symfont}{\um@usv@varrho}%
    \SetMathCode{\um@usv@varpi}{\mathalpha}{\um@symfont}{\um@usv@varpi}%
}
```

\um@def@itgreek

```
\newcommand\um@def@itgreek{%
   \um@setmathcodes{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@itgreek}%
    \SetMathCode{\um@usv@varepsilon}{\mathalpha}{\um@symfont}{"1D716}%
    \SetMathCode{\um@usv@vartheta}{\mathalpha}{\um@symfont}{"1D717}%
    \SetMathCode{\um@usv@varkappa}{\mathalpha}{\um@symfont}{"1D718}%
    \SetMathCode{\um@usv@varphi}{\mathalpha}{\um@symfont}{"1D719}%
    \SetMathCode{\um@usv@varrho}{\mathalpha}{\um@symfont}{"1D71A}%
    \SetMathCode{\um@usv@varpi}{\mathalpha}{\um@symfont}{"1D71B}%
}
```

# File II
# Maths alphabets mapping definitions

```
\newcommand\um@setup@alphabets{%
```

: TODO: nested alphabets?

### 7.4.1 Upright: `\mathup`

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

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ *ϴ*
αβγδεζηθικλμνξοπρστυφχψω *ϵϑϰϕϱϖ*

```
\setmathfont{Cambria Math}
$\mathup{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathup{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathup{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathup{□}$ \\
$\mathup{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathup{□ϑϰϕϱϖ}$ \\
```

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

```
 \um@setmathalphs{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@upLatin}%
 \um@setmathalphs{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@uplatin}%
 \um@setmathalphs{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@upGreek}%
 \um@setmathalphs{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@upgreek}%
 \SetMathAlphChar{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}%
 \SetMathAlphChar{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}%
 \SetMathAlphChar{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}%
 \SetMathAlphChar{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}%
 \SetMathAlphChar{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}%
 \SetMathAlphChar{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}%
 \SetMathAlphChar{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}%
 \SetMathAlphChar{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}%
 \SetMathAlphChar{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}%
```

### 7.4.2 Italic: `\mathit`

*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
*abcdefghijklmnopqrstuvwxyz*
*ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ* *ϴ*
*αβγδεζηθικλμνξοπρστυφχψω*
*ϵϑϰϕϱϖ*

```
\setmathfont{Cambria Math}
$\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathit{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathit{□}$ \\
$\mathit{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathit{□ϑϰϕϱϖ}$ \\
```

Roman:

```
 \um@setmathalphs{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@itLatin}%
 \um@setmathalphs{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@itlatin}%
  \SetMathAlphChar{\mathit}{`\h}{\um@usv@ith}%
  \SetMathAlphChar{\mathit}{\um@usv@ith}{\um@usv@ith}%
```

Greek:

```
\um@setmathalphs{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@itGreek}%
\um@setmathalphs{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@itgreek}%
\SetMathAlphChar{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}%
\SetMathAlphChar{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}%
\SetMathAlphChar{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarTheta}%
\SetMathAlphChar{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}%
\SetMathAlphChar{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}%
\SetMathAlphChar{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}%
\SetMathAlphChar{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}%
\SetMathAlphChar{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}%
\SetMathAlphChar{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}%
```

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

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

```
\setmathfont{Cambria Math}
$\mathbb{0123456789}$ \\
$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbb{abcdefghijklmnopqrstuvwxyz}$ \\
```

Numbers: (always upright)

```
  \um@setmathalphs{\mathbb}{\um@usv@num}{10}{\um@usv@bbnum}%
```

Letters:

```
\um@setmathalphs{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bbLatin}%
\SetMathAlphChar{\mathbb}{`\C,"1D60A}{"2102}%
\SetMathAlphChar{\mathbb}{`\H,"1D60F}{"210D}%
\SetMathAlphChar{\mathbb}{`\N,"1D60F}{"2115}%
\SetMathAlphChar{\mathbb}{`\P,"1D617}{"2119}%
\SetMathAlphChar{\mathbb}{`\Q,"1D618}{"211A}%
\SetMathAlphChar{\mathbb}{`\R,"1D619}{"211D}%
\SetMathAlphChar{\mathbb}{`\Z,"1D621} {"2124}%
```

Roman lowercase:

```
\um@setmathalphs{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bblatin}%
```

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

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

$\mathscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
$\mathscr{abcdefghijklmnopqrstuvwxyz}$

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

```
\um@setmathalphs{\mathscr}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@scrLatin}%
```

```
\SetMathAlphChar{\mathscr}{`\B,"1D435}{"212C}%
\SetMathAlphChar{\mathscr}{`\E,"1D438}{"2130}%
\SetMathAlphChar{\mathscr}{`\F,"1D439}{"2131}%
\SetMathAlphChar{\mathscr}{`\H,"1D43B}{"210B}%
\SetMathAlphChar{\mathscr}{`\I,"1D43C}{"2110}%
\SetMathAlphChar{\mathscr}{`\L,"1D43F}{"2112}%
\SetMathAlphChar{\mathscr}{`\M,"1D440}{"2133}%
\SetMathAlphChar{\mathscr}{`\R,"1D445}{"211B}%
\um@setmathalphs{\mathscr}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@scrlatin}%
\SetMathAlphChar{\mathscr}{`\e,"1D452}{"212F}%
\SetMathAlphChar{\mathscr}{`\g,"1D454}{"210A}%
\SetMathAlphChar{\mathscr}{`\o,"1D45C}{"2134}%
```

### 7.4.5 Fractur or fraktur or blackletter: `\mathfrak`

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

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

Letters, with exceptions $\{\mathfrak{C}, \mathfrak{H}, \mathfrak{I}, \mathfrak{R}, \mathfrak{Z}\}$:

```
\um@setmathalphs{\mathfrak}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@frakLatin}%
\SetMathAlphChar{\mathfrak}{`\C,"1D436}{"212D}%
\SetMathAlphChar{\mathfrak}{`\H,"1D43B}{"210C}%
\SetMathAlphChar{\mathfrak}{`\I,"1D43C}{"2111}%
\SetMathAlphChar{\mathfrak}{`\R,"1D445}{"211C}%
\SetMathAlphChar{\mathfrak}{`\Z,"1D44D}{"2128}%
\um@setmathalphs{\mathfrak}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@fraklatin}%
```

### 7.4.6 Sans serif: `\mathsf`

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

```
\setmathfont{Cambria Math}
$\mathsf{0123456789}$ \\
$\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathsf{abcdefghijklmnopqrstuvwxyz}$ \\
```

```
\um@setmathalphs{\mathsf}{\um@usv@num}{10}{\um@usv@sfnum}%
\um@setmathalphs{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@sfLatin}%
\um@setmathalphs{\mathsf}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@sflatin}%
```

### 7.4.7 Sans serif italic: `\mathsfit`

*0123456789*
*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
*abcdefghijklmnopqrstuvwxyz*

```
\setmathfont{Cambria Math}
$\mathsfit{0123456789}$ \\
$\mathsfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathsfit{abcdefghijklmnopqrstuvwxyz}$ \\
```

```
\um@setmathalphs{\mathsfit}{\um@usv@num}{10}{\um@usv@sfnum}%
\um@setmathalphs{\mathsfit}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@sfitLatin}%
\um@setmathalphs{\mathsfit}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@sfitlatin}%
```

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

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

```
\setmathfont{Code2001}% ugly
$\mathtt{0123456789}$ \\
$\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathtt{abcdefghijklmnopqrstuvwxyz}$ \\
```

```
\um@setmathalphs{\mathtt}{\um@usv@num}{10}{\um@usv@ttnum}%
\um@setmathalphs{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@ttLatin}%
\um@setmathalphs{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@ttlatin}%
```

## 7.5 Bold alphabets' character mappings

### 7.5.1 Bold: `\mathbf`

***0123456789***
***ABCDEFGHIJKLMNOPQRSTUVWXYZ***
***abcdefghijklmnopqrstuvwxyz***
***ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ***
***ϴ□***
***αβγδεζηθικλμνξοπρστυφχψω***
***ϵϑϰϕϱϖ□***

```
\setmathfont{Cambria Math}
$\mathbf{0123456789}$ \\
$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbf{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathbf{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathbf{□F}$ \\
$\mathbf{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathbf{□ϑϰϕϱϖ□}$ \\
```

```
  \um@setmathalphs{\mathbf}{\um@usv@num}{10}{\um@usv@bfnum}%
  \SetMathAlphChar{\mathbf}{\um@usv@Digamma}{"1D7CA}%
  \SetMathAlphChar{\mathbf}{\um@usv@digamma}{"1D7CB}%
\if@um@bfliteral
  \um@setmathalphs{\mathbf}{\um@usv@upLatin}{26}{\um@usv@bfLatin}%
  \um@setmathalphs{\mathbf}{\um@usv@itLatin}{26}{\um@usv@bfitLatin}%
  \um@setmathalphs{\mathbf}{\um@usv@uplatin}{26}{\um@usv@bflatin}%
  \um@setmathalphs{\mathbf}{\um@usv@itlatin}{26}{\um@usv@bfitlatin}%
  \um@setmathalphs{\mathbf}{\um@usv@upGreek}{25}{\um@usv@bfGreek}%
  \um@setmathalphs{\mathbf}{\um@usv@itGreek}{25}{\um@usv@bfitGreek}%
  \um@setmathalphs{\mathbf}{\um@usv@upgreek}{25}{\um@usv@bfgreek}%
  \um@setmathalphs{\mathbf}{\um@usv@itgreek}{25}{\um@usv@bfitgreek}%
  \SetMathAlphChar{\mathbf}{\um@usv@ith}{"1D489}%
  \SetMathAlphChar{\mathbf}{\um@usv@varTheta}{"1D6B9}%
  \SetMathAlphChar{\mathbf}{\um@usv@Nabla}{"1D6C1}%
  \SetMathAlphChar{\mathbf}{\um@usv@Digamma}{"1D7CA}%
  \SetMathAlphChar{\mathbf}{\um@usv@partial}{"1D6DB}%
  \SetMathAlphChar{\mathbf}{\um@usv@varepsilon}{"1D6DC}%
  \SetMathAlphChar{\mathbf}{\um@usv@vartheta}{"1D6DD}%
  \SetMathAlphChar{\mathbf}{\um@usv@varkappa}{"1D6DE}%
  \SetMathAlphChar{\mathbf}{\um@usv@varphi}{"1D6DF}%
  \SetMathAlphChar{\mathbf}{\um@usv@varrho}{"1D6E0}%
  \SetMathAlphChar{\mathbf}{\um@usv@varpi}{"1D6E1}%
  \SetMathAlphChar{\mathbf}{\um@usv@digamma}{"1D7CB}%
\else
  \if@um@bfupLatin
   \um@setmathalphs{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfLatin}%
  \else
   \um@setmathalphs{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfitLatin}%
  \fi
  \if@um@bfuplatin
   \um@setmathalphs{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bflatin}%
    \SetMathAlphChar{\mathbf}{\um@usv@ith}{"1D421}%
  \else
   \um@setmathalphs{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bfitlatin}%
    \SetMathAlphChar{\mathbf}{\um@usv@ith}{"1D489}%
  \fi
  \if@um@bfupGreek
   \um@setmathalphs{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@bfGreek}%
   \SetMathAlphChar{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{"1D6B9}%
  \else
   \um@setmathalphs{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@bfitGreek}%
   \SetMathAlphChar{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{"1D72D}%
  \fi
  \if@um@bfupgreek
   \um@setmathalphs{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@bfgreek}%
   \SetMathAlphChar{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{"1D6DC}%
   \SetMathAlphChar{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{"1D6DD}%
   \SetMathAlphChar{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{"1D6DE}%
    \SetMathAlphChar{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{"1D6DF}%
    \SetMathAlphChar{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{"1D6E0}%
    \SetMathAlphChar{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{"1D6E1}%
  \else
   \um@setmathalphs{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@bfitgreek}%
   \SetMathAlphChar{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{"1D750}%
```

```
    \SetMathAlphChar{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{"1D751}%
    \SetMathAlphChar{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{"1D752}%
     \SetMathAlphChar{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{"1D753}%
     \SetMathAlphChar{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{"1D754}%
     \SetMathAlphChar{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{"1D755}%
  \fi
\fi
%  \SetMathAlphChar{\mathbf}{\um@usv@Nabla}{"1D6C1}%
%  \SetMathAlphChar{\mathbf}{\um@usv@partial}{"1D6DB}%
%  \SetMathAlphChar{\mathbf}{\um@usv@itNabla}{"1D6C1}%
%  \SetMathAlphChar{\mathbf}{\um@usv@itpartial}{"1D6DB}%
```

### 7.5.2 Bold Italic: \mathbfit

***0123456789***
***ABCDEFGHIJKLMNOPQRSTUVWXYZ***
***abcdefghijklmnopqrstuvwxyz***
***ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ***
***ϴ***
***αβγδεζηθικλμνξοπρστυφχψω***
***ϵϑϰϕϱϖ***

```
\setmathfont{Cambria Math}
$\mathbfit{0123456789}$ \\
$\mathbfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbfit{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathbfit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathbfit{▯}$ \\
$\mathbfit{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathbfit{▯ϑϰϕϱϖ}$ \\
```

```
\um@setmathalphs{\mathbfit}{\um@usv@num}{10}{\um@usv@bfnum}%
\um@setmathalphs{\mathbfit}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfitLatin}%
\um@setmathalphs{\mathbfit}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bfitlatin}%
\um@setmathalphs{\mathbfit}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@bfitGreek}%
\um@setmathalphs{\mathbfit}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@bfitgreek}%
\um@setmathalphs{\mathbfit}{\um@usv@bfLatin}{26}{\um@usv@bfitLatin}%
\um@setmathalphs{\mathbfit}{\um@usv@bflatin}{26}{\um@usv@bfitlatin}%
\um@setmathalphs{\mathbfit}{\um@usv@bfGreek}{25}{\um@usv@bfitGreek}%
\um@setmathalphs{\mathbfit}{\um@usv@bfgreek}{25}{\um@usv@bfitgreek}%
\SetMathAlphChar{\mathbfit}{\um@usv@varTheta}{"1D72D}%
\SetMathAlphChar{\mathbfit}{\um@usv@Nabla}{\um@usv@bfitNabla}%
\SetMathAlphChar{\mathbfit}{\um@usv@partial}{"1D74F}%
\SetMathAlphChar{\mathbfit}{\um@usv@varepsilon}{"1D750}%
\SetMathAlphChar{\mathbfit}{\um@usv@vartheta}{"1D751}%
\SetMathAlphChar{\mathbfit}{\um@usv@varkappa}{"1D752}%
\SetMathAlphChar{\mathbfit}{\um@usv@varphi}{"1D753}%
\SetMathAlphChar{\mathbfit}{\um@usv@varrho}{"1D754}%
\SetMathAlphChar{\mathbfit}{\um@usv@varpi}{"1D755}%
```

: TODO: bold symbols to bold+ symbols

### 7.5.3 Bold fractur or fraktur or blackletter: \mathbffrak

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

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

```
\um@setmathalphs{\mathbffrak}{\um@usv@num}{10}{\um@usv@bfnum}%
\um@setmathalphs{\mathbffrak}{\um@usv@upLatin, \um@usv@itLatin,\um@usv@frakLatin}{26}{\um@u
\um@setmathalphs{\mathbffrak}{\um@usv@uplatin,\um@usv@itlatin,\um@usv@fraklatin}{26}{\um@u
```

### 7.5.4 Bold script or calligraphic: \mathbfscr

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

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

```
\um@setmathalphs{\mathbfscr}{\um@usv@num}{10}{\um@usv@bfnum}%
\um@setmathalphs{\mathbfscr}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfscrLatin}%
\um@setmathalphs{\mathbfscr}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bfscrlatin}%
```

### 7.5.5 Bold sans serif: \mathbfsf

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
□□□□□□□□□□□□□□□□□□□□□□□□
□
□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□

```
\setmathfont{Cambria Math}
$\mathbfsf{0123456789}$ \\
$\mathbfsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbfsf{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathbfsf{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathbfsf{□}$ \\
$\mathbfsf{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathbfsf{□ϑϰϕϱϖ}$ \\
```

Numbers: (always upright)

```
\um@setmathalphs{\mathbfsf}{\um@usv@num}{10}{\um@usv@bfnum}%
\um@setmathalphs{\mathbfsf}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfsfLatin}%
\um@setmathalphs{\mathbfsf}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bfsflatin}%
\um@setmathalphs{\mathbfsf}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@bfsfGreek}%
\um@setmathalphs{\mathbfsf}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@bfsfgreek}%
```

Theta symbol:

```
\SetMathAlphChar{\mathbfsf}{\um@usv@varTheta}{"1D767}%
```

Nabla:

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

Partial, epsilon symbol, theta symbol, kappa symbol, phi symbol, rho symbol, pi symbol:

```
\SetMathAlphChar{\mathbfsf}{\um@usv@partial}{"1D789}%
\SetMathAlphChar{\mathbfsf}{\um@usv@varepsilon}{"1D78A}%
\SetMathAlphChar{\mathbfsf}{\um@usv@vartheta}{"1D78B}%
\SetMathAlphChar{\mathbfsf}{\um@usv@varkappa}{"1D78C}%
\SetMathAlphChar{\mathbfsf}{\um@usv@varphi}{"1D78D}%
\SetMathAlphChar{\mathbfsf}{\um@usv@varrho}{"1D78E}%
\SetMathAlphChar{\mathbfsf}{\um@usv@varpi}{"1D78F}%
```

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

**0123456789**
***ABCDEFGHIJKLMNOPQRSTUVWXYZ***
***abcdefghijklmnopqrstuvwxyz***
□□□□□□□□□□□□□□□□□□□□□□□□
□
□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□

```
\setmathfont{Cambria Math}
$\mathbfsfit{0123456789}$ \\
$\mathbfsfit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ \\
$\mathbfsfit{abcdefghijklmnopqrstuvwxyz}$ \\
$\mathbfsfit{ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ}$\quad$\mathbfsfit{□}$ \\
$\mathbfsfit{αβγδεζηθικλμνξοπρστυφχψω}$\quad$\mathbfsfit{□ϑϰϕϱϖ}$ \\
```

```
\um@setmathalphs{\mathbfsfit}{\um@usv@num}{10}{\um@usv@bfnum}%
\um@setmathalphs{\mathbfsfit}{\um@usv@upLatin,\um@usv@itLatin}{26}{\um@usv@bfsfitLatin}%
\um@setmathalphs{\mathbfsfit}{\um@usv@uplatin,\um@usv@itlatin}{26}{\um@usv@bfsfitlatin}%
\um@setmathalphs{\mathbfsfit}{\um@usv@upGreek,\um@usv@itGreek}{25}{\um@usv@bfsfitGreek}%
\um@setmathalphs{\mathbfsfit}{\um@usv@upgreek,\um@usv@itgreek}{25}{\um@usv@bfsfitgreek}%
```

Other symbols:

```
\SetMathAlphChar{\mathbfsfit}{\um@usv@varTheta}{"1D7A1}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@Nabla}{"1D7A9}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@partial}{"1D7C3}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@varepsilon}{"1D7C4}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@vartheta}{"1D7C5}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@varkappa}{"1D7C6}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@varphi}{"1D7C7}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@varrho}{"1D7C8}%
\SetMathAlphChar{\mathbfsfit}{\um@usv@varpi}{"1D7C9}%
```

```
}
```

```
\AtBeginDocument{\setmathfont[Range=ALL]{Cambria Math}}
```

# File III

# stix table data extraction

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

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

```
#!/bin/sh

cat stix-tbl.asc |
awk '
```

If the USV isn't repeated (TODO: check this is valid!) and the entry isn't one of the weird ones in the big block at the end of the STIX table (TODO: check that out!)…

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

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

```
     if (texname      ~ /[\\]/ &&
         substr(texname,0,5) != "\\text"    &&
         substr(texname,3,1) != " "    &&
         class       != " "    &&
         description !~ /<reserved>/ )
```

Print the actual entry corresponding to the unicode character:

```
     print "\\UnicodeMathSymbol{\"" \
           usv "}{" \
           texname "}{" \
           class "}{" \
           description "}%";
    }}' - |
```

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

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

A 'fence' defined by the STIX table is something like `\vert`; in XeTeX this is just a `\mathord` that will grow with the magic of `\XeTeXmathchardef`.

```
    -e ' s/{F}/{\\mathord}/   ' \
    -e ' s/{A}/{\\mathalpha}/ ' \
    -e ' s/{P}/{\\mathpunct}/ ' \
    -e ' s/{B}/{\\mathbin}/   ' \
    -e ' s/{R}/{\\mathrel}/   ' \
    -e ' s/{L}/{\\mathop}/    ' \
    -e ' s/{O}/{\\mathopen}/  ' \
    -e ' s/{C}/{\\mathclose}/ ' > unicode-math.tex
```

# A Documenting maths support in the NFSS

## A.1 Overview

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

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

`\DeclareSymbolFont{`⟨*name*⟩`}`⟨*NFSS decl.*⟩
Declares a named maths font such as `operators` from which symbols are defined with `\DeclareMathSymbol`.

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

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

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

`\DeclareSymbolFontAlphabet{`⟨*cmd*⟩`}{`⟨*name*⟩`}`

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

**Maths 'versions'** Different maths weights can be defined with the following, switched in text with the `\mathversion{`⟨*maths version*⟩`}` command.

`\SetSymbolFont{`⟨*name*⟩`}{`⟨*maths version*⟩`}`⟨*NFSS decl.*⟩
`\SetMathAlphabet{`⟨*cmd*⟩`}{`⟨*maths version*⟩`}`⟨*NFSS decl.*⟩

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

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

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

`\DeclareMathDelimiter{`⟨*symbol*⟩`}{`⟨*type*⟩`}{`⟨*sym. font*⟩`}{`⟨*slot*⟩`}{`⟨*sym. font*⟩`}{`⟨*slot*⟩`}`

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

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

Accents are not included yet.

## A.2 Detailed code investigation

This section contains an abridged and documented version of (bits and pieces of) LATEX's NFSS. Changes are mostly cosmetic and omission of irrelevant things.

## A.3 Maths symbols

`\DeclareMathSymbol`

`#1` : Symbol, e.g., `\alpha` or 'a'
`#2` : Type, e.g., `\mathalpha`
`#3` : Math font name, e.g., `operators`
`#4` : Slot, e.g., `F1`

```
\def\DeclareMathSymbol#1#2#3#4{%
```

First ensure the math font (e.g., `operators`) exists:

```
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
```

Convert the slot number to two hex digits stored in `\count\z@` and `\count\tw@`, respectively:

```
    \begingroup
      \count\z@=#4\relax
      \count\tw@\count\z@
      \divide\count\z@\sixt@@n
      \count@\count\z@
      \multiply\count@\sixt@@n
      \advance\count\tw@-\count@
```

The symbol to be defined can be either a command (`\alpha`) or a character (a). Branch for the former:

```
      \if\relax\noexpand#1% is command?
        \edef\reserved@a{\noexpand\in@{\string\mathchar}{\meaning#1}}%
        \reserved@a
```

If the symbol command definition contains `\mathchar`, then we can provide the info that a previous symbol definition is being overwritten:

```
        \ifin@
          \expandafter\set@mathsymbol
             \csname sym#3\endcsname#1#2%
             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \@font@info{Redeclaring math symbol \string#1}%
```

Otherwise, throw an error if the command name is already taken by a non-symbol definition:

```
        \else
            \expandafter\ifx
            \csname\expandafter\@gobble\string#1\endcsname
            \relax
            \expandafter\set@mathsymbol
               \csname sym#3\endcsname#1#2%
               {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \else
            \@latex@error{Command `\string#1' already defined}\@eha
          \fi
        \fi
```

And if the symbol input is a character:

```
      \else
        \expandafter\set@mathchar
          \csname sym#3\endcsname#1#2
          {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      \fi
    \endgroup
```

Everything previous was skipped if the maths font doesn't exist in the first place:

```
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi}
```

The final macros that actually define the maths symbol with TEX primitives. If the symbol definition is for a macro:

```
\def\set@mathsymbol#1#2#3#4{%
  \global\mathchardef#2"\mathchar@type#3\hexnumber@#1#4\relax}
```

Or if it's for a character:

```
\def\set@mathchar#1#2#3#4{%
  \global\mathcode‘#2="\mathchar@type#3\hexnumber@#1#4\relax}
```

**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@}}}
```

## A.4 Delimiters

The code here is slightly better documented originally than the other maths commands.

\DeclareMathDelimiter

```
\def\DeclareMathDelimiter#1{%
  \if\relax\noexpand#1%
    \expandafter\@DeclareMathDelimiter
  \else
    \expandafter\@xxDeclareMathDelimiter
  \fi
  #1}
\@onlypreamble\DeclareMathDelimiter
```

\@xxDeclareMathDelimiter This macro checks if the second arg is a "math type" such as \mathopen. The undocumented original code didn't use math types when the delimiter was a single letter. For this reason the coding is a bit strange as it tries to support the undocumented syntax for compatibility reasons.

```
\def\@xxDeclareMathDelimiter#1#2#3#4{%
```

7 is the default value returned in the case that \mathchar@type is passed something unexpected, like a math symbol font name. We locally move \mathalpha out of the way so if you use that the right branch is taken. This will still fail if an explicit number 7 is used!

```
   \begingroup
    \let\mathalpha\mathord
    \ifnum7=\mathchar@type{#2}%
      \endgroup
```

If this branch is taken we have old syntax (5 arguments).

```
      \expandafter\@firstofone
    \else
```

If this branch is taken `\mathchar@type` is different from 7 so we assume new syntax. In this case we also use the arguments to set up the letter as a math symbol for the case where it is not used as a delimiter.

```
      \endgroup
      \DeclareMathSymbol#1{#2}{#3}{#4}%
```

Then we arrange that `\@xDeclareMathDelimiter` only gets `#1`, `#3`, `#4` … as it does not expect a math type as argument.

```
      \expandafter\@firstoftwo
    \fi
    {\@xDeclareMathDelimiter#1}{#2}{#3}{#4}}
\@onlypreamble\@xxDeclareMathDelimiter
```

`\@DeclareMathDelimiter`

```
\def\@DeclareMathDelimiter#1#2#3#4#5#6{%
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \expandafter\in@\csname sym#5\expandafter\endcsname
       \expandafter{\group@list}%
    \ifin@
      \begingroup
        \count\z@=#4\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \count\z@=#6\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \edef\reserved@a{\noexpand\in@{\string\delimiter}{\meaning#1}}%
        \reserved@a
        \ifin@
          \expandafter\set@mathdelimiter
             \csname sym#3\expandafter\endcsname
             \csname sym#5\endcsname#1#2%
             \reserved@c\reserved@d
          \@font@info{Redeclaring math delimiter \string#1}%
        \else
```

```
            \expandafter\ifx
            \csname\expandafter\@gobble\string#1\endcsname
            \relax
            \expandafter\set@mathdelimiter
              \csname sym#3\expandafter\endcsname
              \csname sym#5\endcsname#1#2%
              \reserved@c\reserved@d
          \else
            \@latex@error{Command `\string#1' already defined}\@eha
          \fi
        \fi
      \endgroup
    \else
      \@latex@error{Symbol font `#5' is not defined}\@eha
    \fi
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi
}
\@onlypreamble\@DeclareMathDelimiter
```

\@xDeclareMathDelimiter

```
\def\@xDeclareMathDelimiter#1#2#3#4#5{%
  \expandafter\in@\csname sym#2\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \expandafter\in@\csname sym#4\expandafter\endcsname
       \expandafter{\group@list}%
    \ifin@
      \begingroup
        \count\z@=#3\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \count\z@=#5\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
        \expandafter\set@@mathdelimiter
           \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
           \reserved@c\reserved@d
      \endgroup
    \else
      \@latex@error{Symbol font `#4' is not defined}\@eha
    \fi
  \else
    \@latex@error{Symbol font `#2' is not defined}\@eha
  \fi
}
\@onlypreamble\@xDeclareMathDelimiter
```

\set@mathdelimiter

We have to end the definition of a math delimiter like \lfloor with a space and not with \relax as we did before, because otherwise contructs involving \abovewithdelims will prematurely end (pr/1329)

```
\def\set@mathdelimiter#1#2#3#4#5#6{%
  \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
                                     \hexnumber@#2#6 }}
\@onlypreamble\set@mathdelimiter
```

\set@@mathdelimiter

```
\def\set@@mathdelimiter#1#2#3#4#5{%
  \global\delcode`#3="\hexnumber@#1#4\hexnumber@#2#5\relax}
\@onlypreamble\set@@mathdelimiter
```

## A.5 Symbol fonts

\DeclareSymbolFont

#1 : font name, *e.g.*, letters
#2 : font encoding, *e.g.*, OT1
#3 : font family, *e.g.*, cmr
#4 : font series, *e.g.*, m
#5 : font shape, *e.g.*, n

```
\def\DeclareSymbolFont#1#2#3#4#5{%
```

First check that the font encoding is defined.

```
 \@tempswafalse
 \edef\reserved@b{#2}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
```

So far so good. Now branch depending if this symbol font has been declared yet or not. If not, the symbol font is defined as the macro \sym#1; *i.e.*, for the letters symbol font, the associated command name is \symletters. (Funny it's not \sym@#1.)

```
 \if@tempswa
   \@ifundefined{sym#1}{%
```

```
      \expandafter\new@mathgroup\csname sym#1\endcsname
     \expandafter\new@symbolfont\csname sym#1\endcsname{#2}{#3}{#4}{#5}%
   }%
```

If the symbol font has been already declared:

```
     {\@font@info{Redeclaring symbol font `#1'}%
```

[Update the group list.]

```
      \def\group@elt##1##2{%
           \noexpand\group@elt\noexpand##1%
           \expandafter\ifx\csname sym#1\endcsname##1%
             \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
           \else
               \noexpand##2%
           \fi}%
      \xdef\group@list{\group@list}%
```

[Update the version list.]

```
      \def\version@elt##1{%
          \expandafter
          \SetSymbolFont@\expandafter##1\csname#2/#3/#4/#5\expandafter
              \endcsname \csname sym#1\endcsname
          }%
      \version@list
     }%
```

If the font encoding wasn't defined, all of the above was skipped.

```
  \else
    \@latex@error{Encoding scheme  `#2' unknown}\@eha
  \fi}
```

\new@symbolfont

#1 : internal symbol font name, *e.g.*, \symletters
#2 : font encoding, *e.g.*, OT1
#3 : font family, *e.g.*, cmr
#4 : font series, *e.g.*, m
#5 : font shape, *e.g.*, n

```
\def\new@symbolfont#1#2#3#4#5{%
```

Update the group list:

```
    \toks@\expandafter{\group@list}%
    \edef\group@list{\the\toks@\noexpand\group@elt\noexpand#1%
                     \expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
    \def\version@elt##1{\toks@\expandafter{##1}%
                   \edef##1{\the\toks@\noexpand\getanddefine@fonts
                   #1\expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
                  \global\advance\csname c@\expandafter
                                 \@gobble\string##1\endcsname\@ne
                 }%
    \version@list}
```

\SetSymbolFont

#1 : math font version, *e.g.*, `normal`
#2 : font name, *e.g.*, `letters`
#3 : font encoding, *e.g.*, `OT1`
#4 : font family, *e.g.*, `cmr`
#5 : font series, *e.g.*, `m`
#6 : font shape, *e.g.*, `n`

```
\def\SetSymbolFont#1#2#3#4#5#6{%
 \@tempswafalse
 \edef\reserved@b{#3}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
 \if@tempswa
  \expandafter\SetSymbolFont@
    \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
    \endcsname \csname sym#1\endcsname
 \else
  \@latex@error{Encoding scheme  `#3' unknown}\@eha
 \fi
}
```

\SetSymbolFont@

#1 : internal math font version, *e.g.*, `\mv@normal`
#2 : NFSS font, *e.g.*, `\OT1/cmr/m/n`
#3 : internal symbol name, *e.g.*, `\symletters`

```
\def\SetSymbolFont@#1#2#3{%
```

If the maths version has been defined:

```
  \expandafter\in@\expandafter#1\expandafter{\version@list}%
  \ifin@
```

If the symbol font has been defined:

```
    \expandafter\in@\expandafter#3\expandafter{\group@list}%
    \ifin@
      \begingroup
        \expandafter\get@cdp\string#2\@nil\reserved@a
        \toks@{}%
        \def\install@mathalphabet##1##2{%
             \addto@hook\toks@{\install@mathalphabet##1{##2}}%
            }%
        \def\getanddefine@fonts##1##2{%
          \ifnum##1=#3%
             \addto@hook\toks@{\getanddefine@fonts#3#2}%
             \expandafter\get@cdp\string##2\@nil\reserved@b
             \ifx\reserved@a\reserved@b\else
                \@font@warning{Encoding `\reserved@b' has changed
                    to `\reserved@a' for symbol font\MessageBreak
```

```
                 `\expandafter\@gobblefour\string#3' in the
                  math version `\expandafter
                  \@gobblefour\string#1'}%
           \fi
           \@font@info{%
              Overwriting symbol font
              `\expandafter\@gobblefour\string#3' in
               version `\expandafter
              \@gobblefour\string#1'\MessageBreak
              \@spaces \expandafter\@gobble\string##2 -->
                       \expandafter\@gobble\string#2}%
        \else
           \addto@hook\toks@{\getanddefine@fonts##1##2}%
        \fi}%
       #1%
       \xdef#1{\the\toks@}%
    \endgroup
```

If the symbol font wasn't defined, all of the above was skipped:

```
  \else
     \@latex@error{Symbol font `\expandafter\@gobblefour\string#3'
                TBD
```

If the maths version wasn't defined, all of the above was skipped:

```
\else
  \@latex@error{Math version ‘\expandafter\@gobblefour\string#1’
     is not
     defined}{You probably mispelled the name of the math
     version.^^JOr you have to specify an additional package.}%
\fi}
```

# Change History

v0.01

- General: Tidied up awk code 45
- `\addnolimits`: Implemented for `\nolimits` processing 23
- `\um@nolimits`: Implemented for `\nolimits` processing 23
- `\um@radicals`: Implemented for more general radicals processing. 24

v0.1

- General: Ignore `\text`.. STIX commands. 44

v0.2

- `\removenolimits`: Implemented for `\nolimits` processing 24
- `\um@setmathcodes`: Beginning to sort out the input encoding. 32, 33

# 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**

\" . . . . . 14
\@DeclareMathDelimiter . . . . . 74, 93
\@cclvi . . . . . 106
\@eha . . . . . 56, 66, 133, 134, 139, 142, 143, 174, 177, 216, 240, 280
\@elt . . . . . 380–384, 388, 391, 393
\@empty . . . . . 182, 183, 206, 222, 227–242, 247, 422, 425, 439, 455, 460
\@firstofone . . . . . 85
\@firstoftwo . . . . . 89
\@font@info . . . . . 47, 119, 124, 199, 265
\@font@warning . . . . . 259
\@for . . . . . 355, 365, 367, 424, 546, 551
\@gobble . . . . . 50, 127, 128, 148, 151, 226, 270, 271, 377
\@gobblefour 261, 263, 267, 269, 279, 283
\@ifundefined . . . . . 195
\@ii . . . . . 355, 362, 365, 366, 424, 425, 427, 428, 431, 436
\@jj . . . . . 367, 374
\@latex@error . . 56, 66, 133, 134, 139, 142, 143, 174, 177, 216, 240, 279, 283
\@marker . . . . . 436, 450
\@ne . . . . . 26, 226
\@nil . . . . . 249, 257, 436, 447
\@onlypreamble . 79, 92, 145, 180, 184, 187
\@preamblecmds . . . . . 105
\@spaces . . . . . 270
\@tempa . . . . . 35, 72, 91, 97, 195, 204, 418, 448, 460, 553, 555
\@tempb . . . . . 35, 36, 72, 73, 91, 92, 97, 98, 418, 419, 449, 450, 455
\@tempcnta . . . . . 547–549, 552, 554, 555
\@tempswafalse . . . . . 189, 230, 426
\@tempswatrue . . . . . 192, 233, 429, 432, 452, 457, 462, 467
\@um@bfliteraltrue . . . . . 89
\@um@bfupGreekfalse . . . . . 41, 74
\@um@bfupGreektrue . . . . . 52, 63, 79, 84
\@um@bfupLatinfalse . . . . . 43, 76
\@um@bfupLatintrue . . . . . 54, 65, 81, 86
\@um@bfupgreekfalse . . . . . 42, 53, 75, 80
\@um@bfupgreektrue . . . . . 64, 85
\@um@bfuplatinfalse . . . . . 44, 77
\@um@bfuplatintrue . . . . . 55, 66, 82, 87
\@um@fontspec@featuretrue . . . . . 185
\@um@initfalse . . . . . 184
\@um@inittrue . . . . . 420
\@um@literaltrue . . . . . 70
\@um@upGreekfalse . . . . . 37
\@um@upGreektrue . . . . . 48, 59
\@um@upLatinfalse . . . . . 39, 50
\@um@upLatintrue . . . . . 61
\@um@upNablafalse . . . . . 45, 95
\@um@upNablatrue . . . . . 56, 67, 93
\@um@upgreekfalse . . . . . 38, 49
\@um@upgreektrue . . . . . 60
\@um@uplatinfalse . . . . . 40, 51, 62
\@um@uppartialfalse . . . . . 46, 101
\@um@uppartialtrue . . . . . 57, 68, 99
\@xDeclareMathDelimiter . . . . . 91, 146
\@xxDeclareMathDelimiter . . . . . 76, 80
\\ . . . . . 10, 11, 14, 20–28

**Numbers**

\0 . . . . . 469

\  . . . . . 14–17, 20–27

**A**

\A . . . . . 470
\a . . . . . 471
\addnolimits . . . . . 385
\addto@hook . . . . . 252, 256, 273
\advance . . 26, 39, 106, 114, 159, 167, 225
\alloc@ . . . . . 106
\Alpha . . . . . 558
\alpha . . . . . 583

\AtBeginDocument 188, 556
\awint 382

B

\B 41
\begingroup 33, 81, 100, 112, 153, 248, 335, 390
\Beta 559
\beta 584

C

\C 32, 54
\cdp@elt 191, 232
\cdp@list 193, 234
\chardef 106
\Chi 580
\chi 605
\cirfnint 382
\count 34–37, 39, 46, 54, 62, 101–104, 106, 107, 109–112, 114, 115, 154–157, 159, 160, 162–165, 167, 168
\count@ 37–39, 104–106, 112–114, 157–159, 165–167
\csname 20, 30, 45, 50, 53, 61, 94, 97, 109, 118, 121, 122, 125, 127, 128, 130, 131, 139, 147, 148, 150, 151, 170, 179, 186, 196, 197, 202, 203, 210, 211, 221, 224, 225, 237, 238, 334, 336, 377

D

\DeclareMathDelimiter 72
\DeclareMathSymbol 29, 88
\DeclareMathVersion 188
\DeclareSymbolFont 188, 216
\DeclareUnicodeMathSymbol 108, 328, 332
\def 20–23, 29, 68, 70, 72, 80, 93, 105, 106, 108, 145, 146, 175, 181, 185, 187–192, 200, 207, 208, 218, 222, 229, 232, 243, 251, 254, 334, 379, 386, 389, 391, 397, 399, 447–449, 469–544, 558–615
\def@cn 19
\define@choicekey 34, 72, 91, 97, 418
\define@cmdkey 414–417
\define@key 401
\delcode 186
\delimiter 117, 182
\Delta 561
\delta 586
\divide 36, 103, 111, 156, 164
\do 23, 355, 365, 367, 424, 546, 547, 551, 552

E

\E 42
\e 50
\edef 41, 107, 114, 115, 117, 121, 160, 168, 190, 195, 212, 220, 223, 231, 553
\else 48, 55, 59, 65, 75, 86, 93, 96, 102, 109, 120, 121, 125, 126, 132, 133, 137, 138, 141, 142, 154, 158, 163, 168, 173, 176, 204, 210, 215, 239, 244, 258, 263–267, 271, 272, 278, 282, 292, 294, 296, 298, 300, 306, 317, 392, 404, 430, 435, 441, 454, 459, 464
\encodingdefault 217
\endcsname 20, 30, 45, 50, 53, 61, 94, 97, 109, 118, 121, 122, 125, 127, 128, 130, 131, 139, 147, 148, 150, 151, 170, 179, 186, 196, 197, 202, 203, 211, 221, 224, 226, 237, 238, 334, 336, 377
\endgroup 64, 84, 87, 137, 141, 172, 277, 338, 396
\Epsilon 562
\Eta 564
\eta 589
\ExecuteOptionsX 103
\expandafter 20, 28, 30, 31, 44, 49, 50, 52, 60, 74, 76, 85, 89, 94, 95, 97, 98, 109, 110, 117, 120, 121, 124, 126–130, 138, 147, 148, 150–153, 155, 169, 170, 196, 197, 202, 203, 209, 210, 219, 221, 222, 224, 225, 236, 237, 244, 246, 249, 257, 261, 262, 267, 268, 270, 271, 279, 283, 334, 356–362, 368–374, 377, 386–388, 427, 428, 431, 436

F

\F 43
\f@size 186

\fi 29, 57, 58, 63, 67, 71, 77, 90, 96, 98, 102, 105, 112, 129, 130, 134–136, 140, 143, 144, 153, 171–175, 178, 192, 206, 214, 217, 221, 225, 233, 241, 246, 264–267, 274, 281, 282, 287, 294, 296, 298, 300, 311, 322–324, 375, 394, 409, 421, 433, 434, 437, 443, 445, 446, 453, 458, 463, 468

\fint 382

G

\g 51

\g@addto@macro 377, 440, 442

\Gamma 560

\gamma 585

\gdef 150, 157, 160, 165

\get@cdp 249, 257

\getanddefine@fonts 223, 254, 256, 273

\glb@currsize 181

\global 69, 71, 147, 161, 162, 166, 167, 169, 170, 176, 186, 225

\group@elt 200, 201, 220

\group@list 31, 95, 98, 110, 148, 151, 207, 219, 220, 246

H

\H 33, 44, 55

\h 17, 624, 629

\hexnumber@ 46, 54, 62, 69, 71, 107, 115, 160, 168, 182, 183, 186

I

\I 45, 56

\if 40, 73, 113, 427

\if@tempswa 194, 235, 438

\if@um@bfliteral 12, 72, 283

\if@um@bfupGreek 13, 106

\if@um@bfupgreek 14, 113

\if@um@bfupLatin 15, 94, 294, 296, 298, 300

\if@um@bfuplatin 16, 99

\if@um@fontspec@feature 5, 402

\if@um@init 6, 218

\if@um@literal 7, 249

\if@um@upGreek 8, 266

\if@um@upgreek 9, 267

\if@um@upLatin 10, 264

\if@um@uplatin 11, 265

\if@um@upNabla 17, 268, 301

\if@um@uppartial 18, 275, 312

\ifcase 36, 73, 92, 98, 419

\ifin@ 32, 43, 96, 99, 111, 116, 119, 123, 149, 152, 153, 156, 245, 247

\ifnum 27, 83, 255, 366, 451, 456, 461, 465, 466

\ifx 49, 126, 127, 146, 159, 164, 192, 202, 206, 222, 227, 233, 247, 258, 392, 425, 428, 431, 439, 450, 455, 460

\iiiint 380

\iiint 380

\iint 380

\in@ 30, 41, 94, 97, 109, 114, 117, 121, 147, 150, 152, 155, 244, 246

\input 219, 220, 223, 224

\install@mathalphabet 251, 252

\int 380

\intBar 382

\intbar 382

\intcap 384

\intclockwise 381

\intcup 384

\intlarhk 383

\intx 383

\Iota 566

\iota 591

\itAlpha 558

\italpha 583

\itBeta 559

\itbeta 584

\itChi 580

\itchi 605

\itDelta 561

\itdelta 586

\itEpsilon 562

\itEta 564

\iteta 589

\itGamma 560

\itgamma 585

\itIota 566

\itiota 591

\itKappa 567

\itkappa 592

\itLambda 568

\itlambda . . . . . . . . . . . . . . . . . . . . 593
\itMu . . . . . . . . . . . . . . . . . . . . . . . 569
\itmu . . . . . . . . . . . . . . . . . . . . . . . 594
\itNu . . . . . . . . . . . . . . . . . . . . . . . 570
\itnu . . . . . . . . . . . . . . . . . . . . . . . 595
\itOmega . . . . . . . . . . . . . . . . . . . . 582
\itomega . . . . . . . . . . . . . . . . . . . . 607
\itOmicron . . . . . . . . . . . . . . . . . . 572
\itomicron . . . . . . . . . . . . . . . . . . 597
\itPhi . . . . . . . . . . . . . . . . . . . . . . 579
\itphi . . . . . . . . . . . . . . . . . . . . . . 604
\itPi . . . . . . . . . . . . . . . . . . . . . . . 573
\itpi . . . . . . . . . . . . . . . . . . . . . . . 598
\itPsi . . . . . . . . . . . . . . . . . . . . . . 581
\itpsi . . . . . . . . . . . . . . . . . . . . . . 606
\itRho . . . . . . . . . . . . . . . . . . . . . . 574
\itrho . . . . . . . . . . . . . . . . . . . . . . 599
\itSigma . . . . . . . . . . . . . . . . . . . . 576
\itsigma . . . . . . . . . . . . . . . . . . . . 601
\itTau . . . . . . . . . . . . . . . . . . . . . . 577
\ittau . . . . . . . . . . . . . . . . . . . . . . 602
\itTheta . . . . . . . . . . . . . . . . . . . . 565
\ittheta . . . . . . . . . . . . . . . . . . . . 590
\itUpsilon . . . . . . . . . . . . . . . . . . 578
\itupsilon . . . . . . . . . . . . . . . . . . 603
\itvarepsilon . . . . . . . . . . . . . 587, 609
\itvarkappa . . . . . . . . . . . . . . . . . 611
\itvarphi . . . . . . . . . . . . . . . . . . . 612
\itvarpi . . . . . . . . . . . . . . . . . . . . 615
\itvarrho . . . . . . . . . . . . . . . . . . . 613
\itvarsigma . . . . . . . . . . . . . . 600, 614
\itvarTheta . . . . . . . . . . . . . . . . . 575
\itvartheta . . . . . . . . . . . . . . . . . 610
\itXi . . . . . . . . . . . . . . . . . . . . . . . 571
\itxi . . . . . . . . . . . . . . . . . . . . . . . 596
\itZeta . . . . . . . . . . . . . . . . . . . . . 563
\itzeta . . . . . . . . . . . . . . . . . . . . . 588

K

\Kappa . . . . . . . . . . . . . . . . . . . . . . 567
\kappa . . . . . . . . . . . . . . . . . . . . . . 592

L

\L . . . . . . . . . . . . . . . . . . . . . . . . . 46
\Lambda . . . . . . . . . . . . . . . . . . . . . 568
\lambda . . . . . . . . . . . . . . . . . . . . . 593
\left . . . . . . . . . . . . . . . . . . . . . . . 398
\left@primitive . . . . . . . . . . . 398, 399
\let . . . . . . . . . . . . 82, 107, 181–183, 209, 213, 228–243, 245, 353, 398
\long . . . . . . . . . . . . . . . . . . . . . . . 23
\lowint . . . . . . . . . . . . . . . . . . . . . 384

M

\M . . . . . . . . . . . . . . . . . . . . . . . . . 47
\mathalpha . . . 82, 253, 256, 259–262, 269, 270, 272, 273, 276, 277, 279, 280, 302–305, 307–310, 313–316, 318–321, 378, 549, 624, 625, 629, 633, 634, 638, 642–647, 651–656
\mathbb . . . . . . . . . . . . . . . . . . . 30–39
\mathbf . . . . . . . . . . . . 69–71, 73–92, 95, 97, 100, 101, 103, 104, 107, 108, 110, 111, 114–120, 122–128, 131–134
\mathbffrak . . . . . . . . . . . . . . 153–155
\mathbfit . . . . . . . . . . . . . . . . 135–152
\mathbfscr . . 156–158, 160, 161, 174, 175
\mathbfsf . . . . . . . . . . . . . 159, 164–173
\mathbfsfit . . . . . . . . . . . . . . 178–186
\mathcal . . . . . . . . . . . . . . . . . . . . 353
\mathchar . . . . . . . . . . . . . . . . . 41, 121
\mathchar@type . 69, 71, 83, 149, 160, 162, 165, 167, 169, 170, 176, 179, 182
\mathchardef . . . . . . . . . . . . . . . . 69
\mathclose . . . . . . . . . . . . . . . . . . 164
\mathcode . . . . . . . . . . . . . . . . . . . 71
\mathfrak . . . . . . . . . . . . . . . . . 53–59
\mathgroup . . . . . . . . . . . . . . . . . . 106
\mathit . . . . . . . 15–20, 162, 163, 176, 177
\mathop . . . . . . . . . . . . . . . . . . . . . 146
\mathopen . . . . . . . . . . . . . . . . 159, 399
\mathord . . . . . . . . . . . . . . . . . . . . 82
\mathscr . . . . . . . . . . . . . . . . 40–52, 353
\mathsf . . . . . . . . . . . . . . . . . . . . 60–62
\mathsfit . . . . . . . . . . . . . . . . . . 63–65
\mathtt . . . . . . . . . . . . . . . . . . . . 66–68
\mathup . . . . . . . . . . . . . . . 2–14, 21–29
\mddefault . . . . . . . . . . . . . . . . . . 217
\meaning . . . . . . . . . . . 41, 114, 117, 121
\MessageBreak . . . . . . . . . . . . . 260, 269
\Mu . . . . . . . . . . . . . . . . . . . . . . . . 569
\mu . . . . . . . . . . . . . . . . . . . . . . . . 594
\multiply . . . . . . . 38, 105, 113, 158, 166

**N**

\N 34
\new@mathgroup 106, 107, 196
\new@symbolfont 197, 218
\newcommand 1, 31–33, 177, 180, 327, 329, 333, 354, 364, 376, 385, 400, 423, 545, 550, 557, 616, 618, 620, 622, 627, 631, 636, 640, 649
\newcounter 4
\newfam 107
\newif 5–18
\noexpand 40, 41, 73, 113, 114, 117, 121, 195, 201, 203, 205, 220, 221, 223, 224, 393, 427, 554
\nolimits 153
\npolint 383
\Nu 570
\nu 595
\number 554, 555
\numexpr 456, 461, 465, 466, 548, 549, 554, 555

**O**

\o 52
\oiiint 380
\oiint 380
\oint 380
\ointctrclockwise 381
\Omega 582
\omega 607
\Omicron 572
\omicron 597
\or 47, 58, 69, 78, 83, 88, 94, 100

**P**

\P 35
\PackageError 31, 405
\PackageInfo 33
\PackageWarning 32
\partial 608
\Phi 579
\phi 604
\Pi 573
\pi 598
\pointint 383
\ProcessOptionsX 104
\protect 408
\providecommand 19
\ProvidesPackage 1
\Psi 581
\psi 606

**Q**

\Q 36

**R**

\R 37, 48, 57
\relax 24, 27, 34, 36, 40, 51, 69, 71, 73, 92, 98, 101, 109, 113, 128, 129, 146, 149, 154, 157, 159, 161, 162, 164, 166, 167, 169, 170, 176, 179, 181, 186, 366, 419, 427, 450, 451, 456, 461, 465, 466, 548, 549, 554, 555
\removenolimits 389
\RequirePackage 3
\reserved@a 41, 42, 114, 115, 117, 118, 121, 122, 249, 258, 260
\reserved@b 190, 192, 231, 233, 257–259
\reserved@c 107, 123, 132, 160, 171, 191, 192, 232, 233
\reserved@d 115, 123, 132, 168, 171
\Rho 574
\rho 599
\rppolint 382
\rule 194, 205, 215, 226

**S**

\scpolint 383
\set@@mathdelimiter 169, 185
\set@mathchar 60, 70
\set@mathdelimiter 120, 129, 181
\set@mathsymbol 44, 52, 68
\set@xmathsymbol 124
\setkeys 193
\SetMathAlphChar 6–14, 17, 18, 21–29, 32–38, 41–48, 50–52, 54–58, 70, 71, 81–92, 101, 104, 108, 111, 115–120, 123–128, 131–134, 144–152, 164–172, 178–186, 243, 245, 554
\SetMathCode 177, 253, 256, 259–262, 269, 270, 272, 273, 276, 277, 279, 280, 302–305, 307–310, 313–316,

318–321, 378, 548, 624, 625, 629, 633, 634, 638, 642–647, 651–656
\setmathfont 180, 188, 408
\SetSymbolFont 229
\SetSymbolFont@ 210, 236, 243
\sf@size 198, 201
\Sigma 576
\sigma 601
\sixt@@n 36, 38, 103, 105, 111, 113, 156, 158, 164, 166
\sqint 383
\sqrt 397
\stepcounter 211
\string 41, 47, 50, 56, 114, 117, 119, 121, 124, 127, 128, 133, 134, 148, 151, 226, 249, 257, 261, 263, 267, 269–271, 279, 283, 331, 377
\sumint 381

T

\Tau 577
\tau 602
\tf@size 197, 198
\the 220, 223, 276
\Theta 565
\theta 590
\theum@fam 212
\toks@ 219, 220, 222, 223, 250, 252, 256, 273, 276
\tw@ 35, 39, 46, 54, 62, 102, 106, 107, 110, 114, 115, 155, 159, 160, 163, 167, 168

U

\um@addto@mathmap 357, 369, 376
\um@Break 21, 28
\um@char@num@range 183, 365, 439, 440, 442
\um@char@range 182, 206, 222, 227, 247, 422, 424
\um@def@itGreek 266, 636
\um@def@itgreek 267, 649
\um@def@itLatin 264, 620
\um@def@itlatin 265, 622
\um@def@numbers 248, 616
\um@def@upGreek 266, 631
\um@def@upgreek 267, 640
\um@def@upLatin 264, 618
\um@def@uplatin 265, 627
\um@FOR 23, 547, 552
\um@inp 546, 548, 551, 554
\um@Loop 21, 25
\um@mathbb 228
\um@mathbf 229, 237
\um@mathbffrak 239
\um@mathbfit 238
\um@mathbfscr 240
\um@mathbfsf 241
\um@mathbfsfit 242
\um@mathfrak 230
\um@mathit 232
\um@mathmap@noparse 243, 354
\um@mathmap@parse 245, 364
\um@mathscr 233
\um@mathsf 234
\um@mathsfit 235
\um@mathsymbol@noparse 209, 327
\um@mathsymbol@parse 213, 329
\um@mathtt 236
\um@mathup 231
\um@mk@alph 333
\um@mk@math 333, 339–352
\um@mversion 187, 188
\um@nolimits 152, 379, 387, 388, 395
\um@PackageError 31
\um@PackageInfo 33, 208, 331
\um@PackageWarning 32
\um@parse@range 436, 447
\um@parse@term 330, 423
\um@Pool 21, 22, 30
\um@radicals 155, 397
\um@resolve@greek 556
\um@ScriptFeatures 189, 200
\um@ScriptFont 191, 199
\um@ScriptScriptFeatures 190, 203
\um@ScriptScriptFont 192, 202
\um@set@mathchar 138, 175
\um@set@mathsymbol 117, 130, 145
\um@setmathalphs 2–5, 15, 16, 19, 20, 30, 31, 39, 40, 49, 53, 59–69, 73–80, 95, 97, 100, 103, 107, 110, 114, 122, 135–143, 153–163, 173–177, 550

\um@setmathcodes 250–252, 254, 255, 257, 258, 284–291, 293, 295, 297, 299, 545, 550, 617, 619, 621, 623, 628, 632, 637, 641, 650

\um@setup@alphabets 1, 325

\um@symfont 207, 212, 216, 253, 256, 259–262, 269, 270, 272, 273, 276, 277, 279, 280, 302–305, 307–310, 313–316, 318–321, 328, 332, 360, 372, 549, 624, 625, 629, 633, 634, 638, 642–647, 651–656

\um@usv@bbLatin 31, 479

\um@usv@bblatin 39, 480

\um@usv@bbnum 30, 478

\um@usv@bffrakLatin 154, 502

\um@usv@bffraklatin 155, 503

\um@usv@bfGreek 77, 107, 142, 288, 297–300, 496

\um@usv@bfgreek 79, 114, 143, 289, 497

\um@usv@bfitGreek 78, 110, 138, 142, 290, 297–300, 500

\um@usv@bfitgreek 80, 122, 139, 143, 291, 501

\um@usv@bfitLatin 74, 97, 136, 140, 286, 293, 294, 498

\um@usv@bfitlatin 76, 103, 137, 141, 287, 295, 296, 499

\um@usv@bfitNabla 145, 303, 307, 308, 536

\um@usv@bfitpartial 314, 318, 319, 542

\um@usv@bfLatin 73, 95, 140, 284, 293, 294, 494

\um@usv@bflatin 75, 100, 141, 285, 295, 296, 495

\um@usv@bfNabla 302, 303, 307, 535

\um@usv@bfnum 69, 135, 153, 156, 159, 173, 493

\um@usv@bfpartial 313, 314, 318, 541

\um@usv@bfscrLatin 157, 505

\um@usv@bfscrlatin 158, 506

\um@usv@bfscrnum 504

\um@usv@bfsfGreek 162, 510

\um@usv@bfsfgreek 163, 511

\um@usv@bfsfitGreek 176, 514

\um@usv@bfsfitgreek 177, 515

\um@usv@bfsfitLatin 174, 512

\um@usv@bfsfitlatin 175, 513

\um@usv@bfsfitNabla 305, 309, 310, 538

\um@usv@bfsfitpartial 316, 320, 321, 544

\um@usv@bfsfLatin 160, 508

\um@usv@bfsflatin 161, 509

\um@usv@bfsfNabla 304, 305, 537

\um@usv@bfsfnum 507

\um@usv@bfsfpartial 315, 316, 320, 543

\um@usv@Digamma 70, 84, 517

\um@usv@digamma 71, 92, 524

\um@usv@frakLatin 53, 154, 483

\um@usv@fraklatin 59, 155, 484

\um@usv@itGreek 4, 19, 78, 107, 110, 138, 162, 176, 257, 476, 632, 637

\um@usv@itgreek 5, 20, 80, 114, 122, 139, 163, 177, 477, 641, 650

\um@usv@ith 17, 18, 81, 101, 104, 253, 525, 624, 625, 629

\um@usv@itLatin 2, 15, 31, 40, 53, 61, 62, 64, 67, 74, 95, 97, 136, 154, 157, 160, 174, 251, 472, 619, 621

\um@usv@itlatin 3, 16, 39, 49, 59, 65, 68, 76, 100, 103, 137, 155, 158, 161, 175, 252, 473, 623, 628

\um@usv@itNabla 6, 21, 133, 260, 270, 272, 273, 534

\um@usv@itpartial 7, 22, 134, 262, 277, 279, 280, 540

\um@usv@itvarepsilon 9, 24, 115, 123, 527

\um@usv@itvarkappa 11, 26, 117, 125, 529

\um@usv@itvarphi 12, 27, 118, 126, 530

\um@usv@itvarpi 14, 29, 120, 128, 532

\um@usv@itvarrho 13, 28, 119, 127, 531

\um@usv@itvarTheta 8, 23, 108, 111, 526

\um@usv@itvartheta 10, 25, 116, 124, 528

\um@usv@Nabla 6, 21, 83, 131, 145, 165, 179, 259, 269, 270, 272, 533

\um@usv@num 30, 60, 63, 66, 69, 135, 153, 156, 159, 173, 469, 617

\um@usv@partial 7, 22, 85, 132, 146, 166, 180, 261, 276, 277, 279, 539

\um@usv@scrLatin 40, 481

\um@usv@scrlatin 49, 482

\um@usv@sfitLatin 64, 488

\um@usv@sfitlatin 65, 489

\um@usv@sfLatin 61, 486

\um@usv@sflatin 62, 487

\um@usv@sfnum . . . . . . . . . . . . 60, 63, 485
\um@usv@ttLatin . . . . . . . . . . . . 67, 491
\um@usv@ttlatin . . . . . . . . . . . . 68, 492
\um@usv@ttnum . . . . . . . . . . . . . . 66, 490
\um@usv@upGreek . . . . . 4, 19, 77, 107, 110, 138, 162, 176, 255, 474, 632, 637
\um@usv@upgreek . . . . . 5, 20, 79, 114, 122, 139, 163, 177, 258, 475, 641, 650
\um@usv@upLatin . . . . . 2, 15, 31, 40, 53, 61, 62, 64, 67, 73, 95, 97, 136, 154, 157, 160, 174, 250, 470, 619, 621
\um@usv@uplatin . . . . . . . . . . 3, 16, 39, 49, 59, 65, 68, 75, 100, 103, 137, 155, 158, 161, 175, 254, 471, 623, 628
\um@usv@varepsilon . . . . . 9, 24, 86, 115, 123, 147, 167, 181, 518, 642, 651
\um@usv@varkappa . . . . . . 11, 26, 88, 117, 125, 149, 169, 183, 520, 644, 653
\um@usv@varphi . . . . . . . . 12, 27, 89, 118, 126, 150, 170, 184, 521, 645, 654
\um@usv@varpi . . . . . . . . . 14, 29, 91, 120, 128, 152, 172, 186, 523, 647, 656
\um@usv@varrho . . . . . . . . 13, 28, 90, 119, 127, 151, 171, 185, 522, 646, 655
\um@usv@varTheta 8, 23, 82, 108, 111, 144, 164, 178, 256, 516, 633, 634, 638
\um@usv@vartheta . . . . . . 10, 25, 87, 116, 124, 148, 168, 182, 519, 643, 652
\um@zf@feature . . . . . . . . . 400, 410, 412
\UnicodeMathSymbol . . . . . . . . 209, 213
\unless . . . . . . . . . . . . . . . . . . 222, 425
\updefault . . . . . . . . . . . . . . . . . . . 217
\upint . . . . . . . . . . . . . . . . . . . . . . 384
\uppartial . . . . . . . . . . . . . . . . . . . 608
\Upsilon . . . . . . . . . . . . . . . . . . . . 578
\upsilon . . . . . . . . . . . . . . . . . . . . 603

V

\varepsilon . . . . . . . . . . . . . . 587, 609
\varkappa . . . . . . . . . . . . . . . . . . . . 611
\varointclockwise . . . . . . . . . . . . . 381
\varphi . . . . . . . . . . . . . . . . . . . . . . 612
\varpi . . . . . . . . . . . . . . . . . . . . . . . 615
\varrho . . . . . . . . . . . . . . . . . . 613, 614
\varsigma . . . . . . . . . . . . . . . . . . . . 600
\varTheta . . . . . . . . . . . . . . . . . . . . 575
\vartheta . . . . . . . . . . . . . . . . . . . . 610
\version@elt . . . . . . . . . . . . . 208, 222
\version@list . . . . . . . . . . 213, 228, 244

X

\xdef . . . . . . . . . . 182, 207, 276, 395, 422
\XeTeXdelcode . . . . . . . . . . . . . 161, 166
\XeTeXdelimiter . . . . . . . . . . . 160, 165
\XeTeXmathchar . . . . . . . . . . . . . . . . 114
\XeTeXmathchardef . . . . . . . . . 147, 170
\XeTeXmathcode . . 162, 167, 169, 176, 178
\XeTeXradical . . . . . . . . . . . . . . . . . 157
\Xi . . . . . . . . . . . . . . . . . . . . . . . . . 571
\xi . . . . . . . . . . . . . . . . . . . . . . . . . 596
\XKV@rm . . . . . . . . . . . . . . . . . . . . . 204

Z

\Z . . . . . . . . . . . . . . . . . . . . . . . 38, 58
\z@ . . . . . . . . . . . . . . . . . . . . 34–37, 46, 54, 62, 101–104, 107, 109–112, 115, 154–157, 160, 162–165, 168
\zap@space . . . . . . . . . . . . . . . . . . . 422
\Zeta . . . . . . . . . . . . . . . . . . . . . . . 563
\zeta . . . . . . . . . . . . . . . . . . . . . . . 588
\zf@family . . . . . . . . . . . . . . . . . . . 217
\zf@fontspec . . . . . . . . . . . . . . . . . 195
\zf@update@ff . . . . . . . . . . . . . 411, 413