

# Package **mathfont** v. 1.3 Documentation

Conrad Kosowsky

January 2019

`kosowsky.latex@gmail.com`

## Abstract

The **mathfont** package provides a flexible interface for changing the font of math mode characters. The package allows the user to specify a default unicode font for each of six basic classes of Latin and Greek characters, and it provides additional support for unicode math and alphanumeric symbols. Crucially, **mathfont** is compatible with both Lua $\TeX$  and X $\LaTeX$ , and it provides several font-loading commands that allow the user to change fonts locally or for individual characters within math mode.

Handling fonts in  $\TeX$  and  $\LaTeX$  is a notoriously difficult task. The  $\TeX$  engine originally supported only fonts designed with Donald Knuth's Metafont, and while subsequent versions of  $\TeX$  extended this functionality to postscript fonts, Plain  $\TeX$ 's font-loading capabilities remain limited. Many, if not most,  $\LaTeX$  users are unfamiliar with the `.fd` files that must be used in font declaration, and the minutiae of  $\TeX$ 's `\font` primitive can be esoteric and confusing.  $\LaTeX$  2 $\epsilon$ 's New Font Selection System (NFSS) implemented a straightforward syntax for loading and managing fonts, but  $\LaTeX$  macros overlaying a  $\TeX$  core face the same versatility issues as Plain  $\TeX$  itself. Fonts in math mode present a double challenge: even if the user successfully loads a font either in Plain  $\TeX$  or through the NFSS, defining math symbols can be unintuitive for users who are unfamiliar with  $\TeX$ 's `\mathcode` primitive. Again,  $\LaTeX$  2 $\epsilon$  simplifies much of the coding challenges with its `\DeclareMathSymbol` macro, but from the computer's perspective, the fundamental aspects of the task remain the same.

More recent engines such as Jonathan Kew's X $\LaTeX$  and Hans Hagen, et al.'s Lua $\TeX$  significantly extend the font-loading capabilities of  $\TeX$ .<sup>1</sup> Both support TrueType and OpenType font formats and provide many additional primitives for managing fonts. The **fontspec** package by Will Robertson and Khaled Hosny acts as a front-end for the font management built into these two engines.<sup>2</sup> It allows users to easily load new fonts and transition between those ones already in memory, and users who are interested in changing text fonts should consult the documentation for that package. Notably, **fontspec** provides users with the tools to manually adjust OpenType features of the fonts they load, so the package gives equivalent or in some cases superior font management capabilities to a modern WYSIWYG word processor. (Of course, the efficiency depends on the skill of the  $\LaTeX$  user.) In situations where a font's OpenType features are unclear or difficult to access, **fontspec**'s typographical advantages can significantly outweigh the accessibility of a word processor.

This document is organized as follows. Section 1 presents the basic functionality and related packages. The following two sections explain how to use the default and local font-change commands, and users in a hurry will find the most important information here.

---

<sup>1</sup>Information on X $\LaTeX$  is available at <https://tug.org/xetex/>, and information on Lua $\TeX$  is available at the official website for Lua $\TeX$ : <http://www.luatex.org/>.

<sup>2</sup>Will Robertson and Khaled Hosny, "fontspec—Advanced font selection in X $\LaTeX$  and Lua $\LaTeX$ ," <https://ctan.org/pkg/fontspec>.

Sections 4 and 5 discuss the Greek characters and mathematical symbols that **mathfont** provides, and section 6 describes the more complicated error messages that **mathfont** may throw. The remaining five sections detail the implementation. Section 7 lays out the initial implementation for the package, and sections 8 and 9 deal again with default and local font changes, respectively. The following section covers the code for the unicode alphanumeric letterlike symbols, and the final section 11 contains the unicode hex values for every character supported by **mathfont**. For an example of the package in action, see **mathfont\_example.pdf**, which is included with the **mathfont** installation and available on CTAN.

## 1 Basic Functionality

The **mathfont** package uses **fontspec** as a back end to load fonts for use in math mode, and it provides two ways to do this: (1) changing the default font for certain classes of math mode characters; and (2) defining new commands that change the font locally for the so-called “math alphabet” characters. (The math alphabet characters are all Latin letters, all Arabic numerals, and any Greek letters that **mathfont** has set the font for.) The package can change the default math font for Latin and Greek letters, Arabic numerals, and unicode alphanumeric symbols to any unicode OpenType or TrueType font. Tables 1 and 2 display the specific classes of characters that **mathfont**’s default font-change command acts on.

The package can be loaded with the standard **\usepackage** syntax, and it has three optional arguments: **packages**, **operators**, and **no-operators**. During loading, **mathfont** redefines two L<sup>A</sup>T<sub>E</sub>X internal macros to make symbol declaration compatible with unicode fonts, and the **packages** option determines when **mathfont** resets the two internal commands. If the user loads the package without the **packages** option, **mathfont** will reset the internal commands at the next **\usepackage** declaration, but with the argument, **mathfont** does not reset them at all. In practice, using this option should be harmless in most situations, and the issue is primarily one of convenience. The **\mathfont** command works only with these redefinitions, so once the package has reset the L<sup>A</sup>T<sub>E</sub>X kernel, it will not adjust the default font for any class of characters. Thus the **packages** option allows users to sprinkle **\mathfont** declarations throughout their preamble, and doing so may be advantageous depending on the situation. However, leaving the kernel modified could induce clashes with other packages, so proceed at your own risk! For users who provided the **packages** argument or are loading packages before the **\documentclass** declaration, the control sequence **\restoremathinternals** manually returns the internal commands to their default definitions. It should be considered best practice to load **mathfont** without **packages** and put all **\mathfont** commands as soon after loading the package as possible.

The package arguments **operators** and **no-operators** determine **mathfont**’s behavior regarding the integral, summation, and product signs. The **operators** argument tells the package to adjust the font for these three symbols, and the **no-operators** argument does the opposite. By default, the package executes the **operators** option, but users may want to disable this behavior depending on the desired math font and other symbols being used. See sections 2 and 5 for more information.

The package loads **fontspec** with the “no-math” option if and only if the user has not already loaded **fontspec**. Users who want **fontspec** without **no-math** or with other options

in place should manually load it before requiring `mathfont`, and I strongly recommend that users who do so load `fontspec` with the `no-math` option. Alternatively, users who want to pass options to `fontspec` without loading it themselves can use L<sup>A</sup>T<sub>E</sub>X’s `\PassOptionsToPackage` command.

Compatibility with LuaL<sup>A</sup>T<sub>E</sub>X comes at the expense of `mathspec`’s convenient space-adjustment character `"`, and spacing-conscientious users can either manually add `\kern` or `\muskip` to their equations or redefine an active version of `"`.<sup>3</sup> For example, the code

```
\catcode'\ "="\active
\def"#1{\ifmmode
  \kern<dimension>\relax #1\kern<other dimension>\relax
\else
  \char'\ "%
\fi}
```

will serve as a hack that very roughly approximates `mathspec`’s `"`. This code will redefine `"` to typeset a right double quotation mark in horizontal mode, but in math mode, the character will insert *dimension* and *other dimension* of white space on either side respectively of the following character. More advanced users can automate the dimensions by using T<sub>E</sub>X’s `\if` conditional to test whether the following character needs a particular spacing adjustment. (For example, Computer Modern’s math-mode *f* is notorious for being significantly wider than its italic counterpart.) Future versions of `mathfont` may provide an automated space-adjustment command, but it is not one of my priorities.

The functionality of `mathfont` is most closely related to that of the `mathspec` package by Andrew Gilbert Moschou.<sup>4</sup> These two packages incorporate the use of individual unicode characters into math mode, and their symbol declaration process is similar. Both use `fontspec` as a back end, and both create font-changing commands for math mode characters. However, the functionality differs in three crucial respects: `mathfont` is compatible with LuaL<sup>A</sup>T<sub>E</sub>X; it can adjust the font of basic mathematical symbols; and its local math-font-changing mechanisms provide a slightly greater level of flexibility than what users can achieve with `mathspec`. Further, as far as I am aware, this package is the first to provide support for the unicode alphanumeric symbols listed in Table 2, even in the context of fonts without built-in math support. (Please let me know if this is incorrect!) In this way `mathfont`, like `mathspec`, is more versatile than the `unicode-math` package, although potentially less far-reaching.<sup>5</sup>

Users who want to stick with pdfL<sup>A</sup>T<sub>E</sub>X should consider Jean-François Burnol’s `mathastext` as a useful alternative to `mathfont`.<sup>6</sup> This package allows the user to specify the math-mode

---

<sup>3</sup>The problem isn’t one of ability—LuaT<sub>E</sub>X’s programming capabilities tend to outweigh those of X<sub>Ǝ</sub>T<sub>E</sub>X. Rather, I’m unfamiliar with most of the workings of LuaT<sub>E</sub>X. One of my eventual goals for this package is a Lua script that could automatically adjust spacing in math mode for any combination of two characters, but I don’t know if or when I will be able to do that.

<sup>4</sup>Andrew Gilbert Moschou, “`mathspec`—Specify arbitrary fonts for mathematics in X<sub>Ǝ</sub>T<sub>E</sub>X,” <https://ctan.org/pkg/mathspec>.

<sup>5</sup>Will Robertson, “`unicode-math`—Unicode mathematics support for X<sub>Ǝ</sub>T<sub>E</sub>X and LuaT<sub>E</sub>X,” <https://ctan.org/pkg/unicode-math>.

<sup>6</sup>Jean-François Burnol, “`mathastext`—Use the text font in maths mode,” <https://ctan.org/pkg/mathastext>. In several previous versions of this documentation, I mistakenly stated that `mathastext` distorts T<sub>E</sub>X’s internal mathematics spacing. In fact the opposite is true: `mathastext` goes to

font for a large subset of the ASCII characters and is the most closely related package to **mathfont** among those packages designed specifically for the L<sup>A</sup>T<sub>E</sub>X kernel in conjunction with Plain T<sub>E</sub>X. Whereas **mathfont** works exclusively in the context of unicode fonts, **mathastext** was designed for the T1 and related encodings of Plain T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, and **mathfont** provides support for a larger class of symbols. However, the **mathastext** functionality extends beyond that of **mathfont** in two notable aspects: (1) **mathastext** makes use of math versions, extra spacing, and italic corrections; and (2) **mathastext** allows users to change the font for the twenty-five non-alphanumeric characters supported by that package multiple times. After setting the default font, **mathfont** allows users to change it for only Latin, Greek, and digit characters.

## 2 Setting the Default Font

The `\mathfont` command sets the default font for certain classes of characters, and it can be called any number of times in the document preamble. Its structure is

$$\backslash\mathfont[\langle optional\ character\ classes\rangle]\{\langle font\ name\rangle\},$$

where the *optional character classes* can be any set of keywords from Tables 1 and 2, and the *font name* can be any OpenType or TrueType font in a directory searchable by T<sub>E</sub>X. If the user does not specify any optional argument for `\mathfont`, the macro will take all keywords from Table 1 and the **symbols** keyword from Table 2 as the optional argument. When specifying the *font name*, users need to input a name that **fontspec** will recognize and be able to load. Advanced users will note that `\mathfont` uses `\fontspec_set_family:Nnn` and therefore loads fonts in the same way as **fontspec** and related macros from that package. Currently, **mathfont** does not support OpenType features in math mode.

Table 1: Math Alphabet Characters

Keyword	Meaning	Default shape
<b>upper</b>	Capital Latin letters	Italic
<b>lower</b>	Minuscule Latin letters	Italic
<b>greekupper</b>	Capital Greek letters	Upright
<b>greeklower</b>	Minuscule Greek letters	Italic
<b>digits</b>	Arabic numerals	Upright
<b>operator</b>	Operator font	Upright

The user should specify any keyword optional arguments for `\mathfont` as entries in a comma-separated list. The order is irrelevant, and for each item in the list, the package will change the math font for that class of characters to the font specified in the mandatory argument. If the user does not specify an optional argument for `\mathfont`, e.g. writes

$$\backslash\mathfont\{\text{Helvetica}\},$$

the package will change all math-mode Latin characters, Greek characters, digits, operators

---

significant lengths to preserve and even extend rules for space between various math-mode characters.

Table 2: Other Math Symbols

Keyword	Meaning
<code>symbols</code>	Basic mathematical symbols
<code>bb</code>	Blackboard Bold (Double-struck)
<code>cal</code>	Caligraphic
<code>frak</code>	Fraktur
<code>bcal</code>	Bold Caligraphic
<code>bfrak</code>	Bold Fraktur

such as `log` or `sin`, and all symbols from Tables 5 and 6 to the font in the mandatory argument. In this example, that would be Helvetica. By default, `mathfont` will use either an upright or italic shape corresponding to each character-class keyword, but the user can override this setting by writing an equals sign next to the keyword and a shape suboption—either “`roman`” or “`italic`”—following that. Table 1 contains the default shapes for the keywords listed there, while the default shape for all classes of characters in Table 2 is upright. The package interprets the `roman` suboption as specifying an upright shape—normal shape in the language of the NFSS—and the `italic` suboption as specifying an italic shape. Currently `mathfont` does not support default font shapes beyond these two. For example, the command

```
\mathfont[upper=roman,lower=roman]{Times New Roman}
```

would change all math-mode Latin letters to Times New Roman with upright shape. The optional argument for `\mathfont` should not contain any spaces because `mathfont` will not be able to parse the text correctly, and if the user mistakenly includes a space after a comma or around an equals sign, the package will issue an “Invalid option” error. Future versions of `mathfont` may address this limitation, but it is not a priority. Users should be aware that, in order to limit the number of symbol fonts in memory, `\mathfont` will not change the default font for a class of symbols once it has already done so.

The Unicode Consortium designates the Mathematical Alphanumeric Symbols (U+1D400–U+1D7FF) code block for several classes of letters and letterlike symbols that appear frequently in mathematical writing. The `mathfont` package provides easy access to these symbols through the `\mathfont` command, and Table 2 lists the corresponding keywords that should be used with `\mathfont` to do so. When the user calls `\mathfont` with any of the last five keywords from Table 2 in its optional argument, `mathfont` both declares the appropriate unicode characters as math symbols and defines the macro

```
\math<keyword>{\argument}.
```

This control sequence takes one argument, and it converts the characters in its argument into the corresponding style. For example,

```
\mathfont[bcal]{STIXGeneral}
```

will set `STIXGeneral` as the font for bold caligraphic characters and define the command `\mathbcal` to access them in math mode. The argument of the macro produced this way must be a single string of Latin letters without braces or control sequences, although in somewhat idiosyncratic fashion, the `\mathbb` command can also accept Arabic numerals in its argument. Spaces and any character of catcode 12 will be ignored.

Because these commands can handle capital and minuscule Latin letters, the `mathfont` package significantly extends the functionality of T<sub>E</sub>X’s standard `\mathcal` command and the `\mathbb` command from the `amssymb` package. Users should note that typesetting symbols from a letterlike class of characters in Table 2 will require a unicode font that actually contains those characters, such as Symbola or STIXGeneral. That being said, users who need only a few alphanumeric symbols may be able to use a less specialized font that contains for example the characters from the Letterlike Symbols (U+2100–U+214F) code block. This block contains the blackboard bold versions of C, N, Q, R, and Z, and the fonts providing these characters are significantly more common than those that contain the full blackboard bold alphabet.

### 3 Local Font Changes

With `mathfont`, users can locally change the font in math mode by creating a new control sequence for each new font desired. The commands created this way function analogously to the standard math font macros from the L<sup>A</sup>T<sub>E</sub>X kernel such as `\mathrm`, `\mathit`, and `\mathnormal`, and the package includes four basic commands to produce new control sequences. Table 3 lists these commands, and they all have the same argument structure: a control sequence as the first mandatory argument and a font name as the second. For example, the macro `\newmathrm` looks like

$$\backslash\mathrm{\langle control\ sequence\rangle}\{\langle font\ name\rangle\}.$$

It defines the *control sequence* in its first argument to accept a string of characters that it then converts to the *font name* in the second argument with upright shape and medium weight. Writing

$$\backslash\mathrm{\backslash\mathrm{arial}}\{\mathrm{Arial}\}$$

would create the macro

$$\backslash\mathrm{arial}\{\langle argument\rangle\},$$

which can be used only in math mode and which converts the math alphabet characters in its *argument* into the Arial font with upright shape and medium weight. The other three commands in Table 3 function in the same way except that they select different series or shape values for the font in question, and Table 3 lists this information.

Unlike the traditional `\mathrm` and company, `mathfont`’s local font change commands create macros that can act on Greek characters. If the user specifies the font for Greek letters using `\mathfont`, macros created with the commands from Table 3 will affect those characters; otherwise, they will not. Specifically, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> defines Greek letters as `\mathord` characters, and `mathfont` changes this classification to `\mathalpha` type when it declares them as symbols. The local font change commands here act only on characters of class `\mathalpha`, so if the user doesn’t set the font for `greekupper` or `greeklower` characters in a `\mathfont` command, any macro from this section will not be able to act on any Greek letters. Instead, they will still be typeset as `\mathord` characters in Computer Modern.

Together these four commands will provide users with the tools for almost all desired local font changes, but they inevitably will be insufficient for some particular case. Accordingly,

Table 3: Font-changing Commands

Command	Font Characteristics
<code>\newmathrm</code>	Upright shape; medium weight
<code>\newmathit</code>	Italic shape; medium weight
<code>\newmathbold</code>	Upright shape; bold weight
<code>\newmathboldit</code>	Italic shape; bold weight

`mathfont` provides the more general `\newmathfontcommand` macro that functions similarly to the commands from Table 3 but allows the user to specify certain font characteristics. Its argument structure is

`\newmathfontcommand{⟨control sequence⟩}{⟨font name⟩}{⟨series⟩}{⟨shape⟩},`

where the font name again means any OpenType or TrueType font in a directory searchable by `TEX`. The series and shape information refers to the NFSS codes for these attributes, and the control sequence in the first argument becomes the command that allows the user to access the specified font. Advanced users will note that these five macros are essentially wrapped versions of `LATEX`'s `\DeclareMathAlphabet`, and like `\mathfont`, they must be used in the document preamble.

## 4 Greek Characters

Unlike Plain `TEX` and `LATEX 2ε`, `mathfont` defines control sequences for all characters in the Greek alphabet. (`TEX`'s OT encodings do not include the capital Greek characters that resemble Latin letters. Presumably Donald Knuth made this decision to conserve encoding slots.) All such control sequences instruct `LATEX` to typeset the corresponding unicode Greek character, and `mathfont` also provides control sequences for several ancient Greek letters. This package's additional Greek character control sequences will become available once the user calls `\mathfont` for the corresponding class of Greek characters. Table 4 lists all such control sequences. I do not include typeset versions of these characters in this documentation because I wanted to avoid potential compilation problems regarding `mathfont.dtx`. However, see the file `mathfont_greek.pdf`, which is included with the `mathfont` installation and is available on CTAN. As with the unicode letterlike symbols, not all fonts contain ancient Greek or even modern Greek characters, and setting `greekupper` and `greeklower` to a font without these symbols will produce a document with missing characters.

## 5 Math Symbols

With the `symbols` keyword, the `\mathfont` command will change the default math-mode font for all characters in Table 5. For the control sequences on the left half of the table that function in both math and horizontal modes, `mathfont` acts on the character only when it appears in math mode, so the character will be unaffected when used in regular text. The same is true for the punctuation marks on the right side. Notice that by default the

Table 4: Greek Character Control Sequences

Modern Greek Characters		Ancient Greek Characters	
Capital	Minuscule	Capital	Minuscule
\Alpha	\alpha	\Heta	\heta
\Beta	\beta	\Sampi	\sampi
\Gamma	\gamma	\Diagamma	\diagama
\Delta	\delta	\Koppa	\koppa
\Epsilon	\epsilon	\Stigma	\stigma
\Zeta	\zeta	\Sho	\sho
\Eta	\eta	\San	\san
\Theta	\theta	\varSampi	\varsampi
\Iota	\iota	\varDiagamma	\vardiagamma
\Kappa	\kappa	\varKoppa	\varkoppa
\Lambda	\lambda		
\Mu	\mu		
\Nu	\nu		
\Xi	\xi		
\Omicron	\omicron		
\Pi	\pi		
\Rho	\rho		
\Sigma	\sigma		
\Tau	\tau		
\Upsilon	\upsilon		
\Phi	\phi		
\Chi	\chi		
\Psi	\psi		
\Omega	\omega		
\varTheta	\varbeta		
	\varepsilon		
	\vartheta		
	\varrho		
	\varsigma		
	\varphi		

package acts on the integral, summation, and product symbols. As stated previously, users can enable or disable this behavior with the optional package arguments **operators** and **no-operators** respectively, and doing so may be desirable depending on which math symbols the user wants and the math font in use. Finally, a technical point: when **mathfont** acts on the **symbols** characters, it redefines the four macros `\#`, `\$`, `\%`, and `\&` as robust commands more complicated than a single `\char`, so constructs like `\number\#` will need to be rewritten as `\number'\#`.

Choosing which unicode characters to recode is something of a delicate task because so few unicode fonts contain more than the most basic mathematics symbols. In designing this



Table 5: Characters Corresponding to the `symbols` Keyword

Character	Control Sequence	Class	Punctuation	Class
#	\#	\mathord	.	\mathord
\$	\\$	\mathord	@	\mathord
%	\%	\mathord		\mathord
&	\&	\mathord	'	\mathord
¶	\P	\mathord	"	\mathord
§	\S	\mathord	+	\mathbin
£	\pounds	\mathord	-	\mathbin
¬	\neg	\mathord	*	\mathbin
∞	\infty	\mathord	/	\mathbin
∂	\partial	\mathord	=	\mathrel
Δ	\Delta	\mathord	<	\mathrel
Σ*	\sum	\mathop	>	\mathrel
Π*	\prod	\mathop	:	\mathrel
∫*	\int	\mathop	(	\mathopen
×	\times	\mathbin	)	\mathclose
÷	\div	\mathbin	[	\mathopen
±	\pm	\mathbin	]	\mathclose
·	\cdot	\mathbin	!	\mathclose
•	\bullet	\mathbin	?	\mathclose
†	\dag	\mathbin	,	\mathpunct
‡	\ddag	\mathbin	;	\mathpunct
\	\setminus	\mathbin		
≤	\leq	\mathrel		
≥	\geq	\mathrel		
≈	\sim	\mathrel		
≈	\approx	\mathrel		
≡	\equiv	\mathrel		
	\mid	\mathrel		
	\parallel	\mathrel		
:	\colon	\mathpunct		
...	\dots	\mathinner		

\*If the package option `operators` is enabled.

Table 6: Extra Macros Created with the `symbols` Keyword

Character	Control Sequence	Class	Character	Control Sequence	Class
\	\mathbackslash	\mathord	{	\leftbrace	\mathopen
°	\degree	\mathord	}	\rightbrace	\mathclose

portion of `mathfont`, I attempted to find the largest set of characters that reliably appears in every or nearly every major unicode font, and that’s why Table 5 contains no arrows, subset and element symbols, quantifiers, and complicated operation and grouping symbols. Most standard unicode fonts don’t contain those glyphs, and I wanted `mathfont` to be useful for as many fonts as possible. That being said, the package provides the four new control sequences listed in Table 6. The `\degree` command functions as the unicode alternative to the `$^\circ$` construction, and the other three control sequences print the associated unicode character. None of these characters are coded as delimiters, so they will not respond to  $\TeX$ ’s `\left` and `\right` sizing commands.<sup>7</sup>

## 6 Handling Errors

Most of `mathfont`’s error messages are self-explanatory, and the help text will contain instructions for how to resolve the message. Nevertheless, some of the possible error messages warrant additional explanation.

The most salient error is the “Missing  $\TeX$  or Lua $\TeX$ ” message. If a user tries to load `mathfont` without one of these two engines,  $\TeX$  will crash, either through `mathfont`’s fatal error message or when `fontspec` performs its own engine checks. Advanced users should note that `mathfont` doesn’t actually determine the typesetting engine. Rather, it checks whether the  $\TeX$  and Lua $\TeX$  primitives `\Umathcode` and `\Umathchardef` are defined, so if for some reason these control sequences have definitions when the user loads `mathfont`, `fontspec`’s more robust engine checks will cause  $\TeX$  to abort. The reasoning here is straightforward: `mathfont` verifies only that the current typesetting engine provides the commands that it directly needs, so its potential functionality remains as broad as possible. If `fontspec` becomes compatible with a third engine that also provides (analogues of) `\Umathcode` and `\Umathchardef`, there is no reason to prevent `mathfont` from working with that engine as well. The obvious corollary to this setup is that users with very old  $\TeX$  or Lua $\TeX$  distributions could see this fatal error even when running one of these engines, and the solution is probably to upgrade to a more recent version of the engine in question. Unfortunately, I do not know exactly when  $\TeX$  and Lua $\TeX$  introduced these primitives.<sup>8</sup>

The `fontspec` package includes a “no-math” option, and `mathfont` expects `fontspec` to be loaded with this option. As mentioned previously, `mathfont` loads `fontspec` by default, but users can load `fontspec` before `mathfont` if they want to manually specify the package options. Alternatively,  $\TeX$ ’s `\PassOptionsToPackage` may be an even better way to proceed. If `mathfont` detects that `fontspec` was loaded without the no-math option, it will issue an error message saying so. This error is not paramount in the sense that the document will compile almost normally if a user ignores it, but `mathfont` will probably have trouble changing the font of certain math-mode characters in this situation. During development, Arabic numerals posed a particular challenge in this regard.

---

<sup>7</sup>Because of time constraints, I couldn’t figure out how to use Lua $\TeX$ ’s `\Udelimiter` primitive to code the backslash and braces properly. Any suggestions in this area are welcome.

<sup>8</sup>However, the manual for a beta version of Lua $\TeX$ , v. 0.70.1, includes both primitives, so they are at least as old as May 2011. See

<https://os1.ugr.es/CTAN/obsolete/systems/luatex/base/manual/luatexref-t.pdf>

The “internal commands restored” error arises when the user calls `\mathfont` in a situation where the package already restored the small portion of the  $\text{\LaTeX}$  kernel that it adjusts when loaded. Unless the user loads `mathfont` with the `packages` option, this process happens when the user next calls `\usepackage`, and `mathfont` will not carry out any calls to `\mathfont` at that point. As mentioned previously, users who want to manually restore the  $\text{\LaTeX}$  kernel can do so with the `\restoremathinternals` command.

Users who receive an “Invalid option” error for `\mathfont` should make sure they removed any extra spaces from their list of keywords in the command’s optional argument. It is also worth pointing out if the user tries to reset the default font for some characters with `\mathfont`, the package will issue a warning but will not halt the compilation process.

What should you do if you can’t resolve an error? First, always, always make sure that you spelled all of your commands correctly and closed all braces and brackets. Then check the `mathfont` documentation—you may be trying to do something outside the scope of the package, or you may be dealing with a special case. In particular, read all the sections of the documentation that relate to the command or commands you are having trouble with. The internet is a great resource, and websites such as the  $\text{\TeX}$  StackExchange, Share $\text{\LaTeX}$ , and Wikibooks’s  $\text{\LaTeX}$  wiki are often invaluable in dealing with  $\text{\TeX}$ -related issues. Definitely ask another human as well! At that point you should email the author about your code—you might have identified a bug. Please include a minimal working example that demonstrates the problem in any such correspondence. I welcome emails about `mathfont` and will make every effort to write back to emails about the package, but I cannot guarantee a timely response.

## 7 Implementation

First and foremost, the package needs to declare itself. The `packages` option changes the `\if@packages` boolean from false to true.

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{mathfont}[2019/01/19 v. 1.3 Package mathfont]
3 \newif\if@packages
4 \newif\if@operators
5 \@operatorstrue
6 \DeclareOption{packages}{\@packagestrue}
7 \DeclareOption{operators}{\@operatorstrue}
8 \DeclareOption{no-operators}{\@operatorsfalse}
9 \DeclareOption*{\PackageError{mathfont}
10   {Option "\CurrentOption" unknown}
11   {I'm ignoring that option because\MessageBreak
12     I don't know what it means.}}
13 \ProcessOptions*
```

Some error and informational messages.

```

14 \def\M@FontChangeInfo#1{\PackageInfo{mathfont}
15   {Changing #1 characters to\MessageBreak
16   math font \csname M@font\the\M@count\endcsname\MessageBreak
```

```

17 with \csname @#1 shape\endcsname\MessageBreak
18 shape}}
19 \def\M@NewFontCommandInfo#1#2#3#4{\PackageInfo{mathfont}
20 {Creating math alphabet\MessageBreak
21 command \string#1 using #2\MessageBreak
22 with series #3 and shape\MessageBreak
23 #4}}
24 \def\M@CommandInitializeInfo#1{\PackageInfo{mathfont}
25 {Initializing #1 font-change\MessageBreak
26 command}}
27 \def\M@SetInternalsInfo{\PackageInfo{mathfont}
28 {Adjusting \string\set@mathchar\space and\MessageBreak
29 \string\set@mathsymbol}}
30 \def\M@RestoreInternalsInfo{\PackageInfo{mathfont}
31 {Restoring \string\set@mathchar\space and\MessageBreak
32 \string\set@mathsymbol}}
33 \def\M@CharsSetWarning#1{\PackageWarning{mathfont}
34 {Font for #1 chars has already\MessageBreak
35 been set}}
36 \def\M@NoMathError{\PackageError{mathfont}
37 {Package fontspec was loaded\MessageBreak
38 without the "no-math" option}
39 {This isn't really an error--\MessageBreak
40 it's fine to load fontspec\MessageBreak
41 without "no-math." However,\MessageBreak
42 strange things could happen,\MessageBreak
43 so beware of any sudden and\MessageBreak
44 unexpected font changes. To\MessageBreak
45 resolve this error message,\MessageBreak
46 load fontspec with the "no-\MessageBreak
47 math" option. If you haven't\MessageBreak
48 loaded fontspec manually, try\MessageBreak
49 loading mathfont earlier in\MessageBreak
50 your preamble.}}
51 \def\M@InvalidOptionError#1{\PackageError{mathfont}
52 {Invalid option "#1"\MessageBreak for \string\mathfont}
53 {Hm. Check that you spelled the\MessageBreak
54 option correctly and didn't include\MessageBreak
55 any extra spaces. Otherwise, I'm\MessageBreak
56 not sure what's wrong. Is this\MessageBreak
57 option listed in the package\MessageBreak
58 documentation?}}
59 \def\M@InternalsRestoredError{\PackageError{mathfont}
60 {Internal commands restored}
61 {This package slightly changes two LaTeX\MessageBreak
62 internal commands, and you really shouldn't\MessageBreak
63 be loading new math fonts without those\MessageBreak

```

```

64 adjustments. What happened here is that you\MessageBreak
65 used \string\mathfont\space in a situation
66 where those\MessageBreak
67 two commands retain their original defini-\MessageBreak
68 tions. Presumably you used
69 \string\mathfont\space after\MessageBreak
70 loading another package or calling the\MessageBreak
71 \string\restoremathinternals\space
72 command, and in any\MessageBreak
73 event, I'm going to ignore this call to\MessageBreak
74 \string\mathfont. Try retypesetting this document\MessageBreak
75 with all \string\mathfont\space
76 commands immediately\MessageBreak
77 after \string\usepackage{mathfont}.}}
78 \def\M@MissingControlSequenceError#1{\PackageError{mathfont}
79 {Improper argument\MessageBreak
80 for \string#1}
81 {Please use a control sequence as the\MessageBreak
82 first argument of \string#1.}}
83 \def\M@HModeError#1{\PackageError{mathfont}
84 {Missing \string$ inserted.\MessageBreak
85 Command \string#1 must be used in\MessageBreak
86 math mode}
87 {I generated an error because\MessageBreak
88 you used \string#1 outside of\MessageBreak
89 math mode. I've inserted a \string$\MessageBreak
90 just before your \string#1, so\MessageBreak
91 we should be all good now.}}
92 \def\M@XeTeXLuaTeXError{\PackageError{mathfont}
93 {!!FATAL ERROR!! : Missing\MessageBreak XeTeX or LuaTeX}
94 {This package requires either\MessageBreak
95 XeLaTeX or LuaLaTeX. Please\MessageBreak
96 retypeset this document with\MessageBreak
97 one of those two engines. This\MessageBreak
98 error is fatal, so I'm going\MessageBreak
99 to stop processing once you\MessageBreak
100 push enter.}\@@end}

```

Check if the current engine has defined the necessary primitives.

```

101 \ifx\Umathcode\@undefined
102 \else
103   \ifx\Umathchardef\@undefined
104   \else
105     \let\M@XeTeXLuaTeXError\relax
106   \fi
107 \fi
108 \M@XeTeXLuaTeXError

```

We load the fontspec package in order to use its main font loading mechanism, and we \let

the macro `\@newfont` take on this function. We also make sure that `fontspec` was loaded with the `no-math` option because without it, `fontspec` may cause trouble with some of the math characters. If `\g__fontspec_math_bool` is equal to 1, `mathfont` will issue an error message.

```

109 \@ifpackageloaded{fontspec}
110   {\ifnum\csname g__fontspec_math_bool\endcsname=1
111     \M@NoMathError
112     \fi}{\RequirePackage[no-math]{fontspec}}
113 \expandafter\let\expandafter\@newfont
114   \csname fontspec_set_family:Nnn\endcsname

```

We save `\set@mathchar` and `\set@mathsymbol` from the L<sup>A</sup>T<sub>E</sub>X kernel so we can change their definitions. We need to adapt these macros for use with unicode fonts, and we replace `\mathcode` and `\mathchardef` respectively with the X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X and Lua<sub>T</sub>E<sub>X</sub>primitives `\Umathcode` and `\Umathchardef`. The unicode primitives support decimal input using a + sign, and we take advantage of that feature to avoid hexadecimal conversions.

```

115 \M@SetInternalsInfo
116 \let\@@set@mathchar\set@mathchar
117 \let\@@set@mathsymbol\set@mathsymbol
118 \def\set@mathchar#1#2#3#4{%
119   \multiply\count\z@ by 16\relax
120   \advance\count\z@\count\tw@
121   \global\Umathcode'#2=\mathchar@type#3+#1+\count\z@}
122 \def\set@mathsymbol#1#2#3#4{%
123   \multiply\count\z@ by 16\relax
124   \advance\count\z@\count\tw@
125   \global\Umathchardef#2\mathchar@type#3+#1+\count\z@}

```

We need to keep track of the number of times we have loaded a font. The count `\M@count` fulfills this role.

```

126 \newcount\M@count
127 \M@count\z@

```

We create necessary booleans and the default math font shapes.

```

128 \newif\if@upper
129 \newif\if@lower
130 \newif\if@greekupper
131 \newif\if@greeklower
132 \newif\if@operator
133 \newif\if@digits
134 \newif\if@symbols
135 \newif\if@bb
136 \newif\if@cal
137 \newif\if@frak
138 \newif\if@bcal
139 \newif\if@bfrak
140 \newif\if@suboptionpresent
141 \def\@uppershape{italic}
142 \def\@lowershape{italic}
143 \def\@greekuppershape{roman}

```

```

144 \def\@greeklowershape{italic}
145 \def\@digitssshape{roman}
146 \def\@symbolssshape{roman}
147 \def\@operatorshape{roman}
148 \def\@bbshape{roman}
149 \def\@calshape{roman}
150 \def\@frakshape{roman}
151 \def\@bcshape{roman}
152 \def\@bfrakshape{roman}

```

We want to allow the user to specify options using an xkeyval-type syntax. However, we do not need the full package; a slim 31 lines of code will suffice. The macro `\check@suboption` determines whether the user specified a keyword option for `\mathfont` using a `key=value` input and stores both the option and corresponding suboption, if applicable, in `\@tempa` and `\@tempb` respectively.

```

153 \def\strip@equals#1={#1}
154 \def\check@suboption#1=#2\@nil{%
155   \def\@tempa{#1}
156   \def\@tempb{#2}
157   \check@option@valid\@tempa
158   \ifx\@tempb\@empty
159     \@suboptionpresentfalse
160   \else
161     \edef\@tempb{\expandafter\strip@equals\@tempb}
162     \check@suboption@valid\@tempb
163     \@suboptionpresenttrue
164   \fi}

```

The commands `\check@option@valid` and `\check@suboption@valid` confirm that the keyword option and suboption are legitimate. If not, `mathfont` issues an error.

```

165 \def\check@option@valid#1{%
166   \def\@temperror{\M@InvalidOptionError{#1}}
167   \@for\@j:=upper,lower,greekupper,greeklower,digits,symbols,operator\do{%
168     \ifx\@j#1
169       \let\@temperror\relax
170     \fi}

```

We have to initialize the blackboard, calligraphic, and fraktur commands separately because they don't use the same encoding-alphabet system as the regular letters and digits. The `\define@<keyword>` macro does this.

```

171 \@for\@j:=bb,cal,frak,bcal,bfrak\do{%
172   \ifx\@j#1
173     \let\@temperror\relax
174     \csname define@\@j\endcsname
175   \fi}
176 \@temperror}
177 \def\check@suboption@valid#1{%
178   \def\@temperror{\M@InvalidOptionError{#1}}
179   \@for\@j:=roman,italic\do{%

```

```

180   \ifx\@j#1
181     \let\@temperror\relax
182   \fi}
183 \@temperror}

```

## 8 Implementation of Default Font Changes

The user-level command `\mathfont` passes its mandatory argument to `\@newfont` through the `\@mathfont` command. If the user includes `\mathfont` with an optional argument, `\@mathfont` will loop through each item in the optional argument using `\@for` and set the math font for each of those classes of characters. If the user does not specify an optional argument, `\mathfont` calls `\@mathfont` with the list of entries from Table 1 in brackets. The `\@mathfont` macro proceeds in several steps. First, it checks if `\set@mathchar` has been reset, and if so, the current call to `\mathfont` will do nothing. If not, the macro loads the user's font with `\@newfont` and checks its keyword optional arguments. The command declares the appropriate symbol fonts if necessary, actually sets the math font by calling `\M@<keyword>\set` command, and records that the font change occurred by switching the appropriate boolean to true. Finally, `\@mathfont` increments `\M@count` by one.

The package stores each new font in a macro of the form `\M@font<number>`, where *number* is given by the current value of `\M@count`. The name of the corresponding symbol fonts is `M<shape><number>`, where *shape* is either *roman* or *italic* and *number* is again the value of `\M@count`. For each keyword from Tables 1 and 2, the package originally defines `\@<keyword>shape` as the default shape for that collection of characters. If the user specifies a suboption for any keyword in the optional argument of `\mathfont`, the package overrides the default shape by redefining `\@<keyword>shape` to the value of the suboption. For example, suppose the user loads `mathfont` and then immediately writes

```
\mathfont[upper=roman]{Helvetica}.
```

In this situation, `mathfont` would define `\M@font0` to be the internal name of Helvetica, and the corresponding symbol font names would be `Mroman0` and `Mitalic0`. Because the user specified a suboption, `\@mathfont` would redefine `\M@uppershape` to the token string “*roman*,” and the package will use `\M@uppershape` later to specify the symbol font for capital Latin characters.

```

184 \def\gobble@equals#1=#2\@nil{#1}
185 \def\mathfont{\@ifnextchar[{\@mathfont}
186   {\@mathfont[upper,lower,greekupper,greeklower,digits,symbols,operator]}}
187 \def\@mathfont[#1]#2{%
188   \ifx\@@set@mathchar\set@mathchar
189     \M@InternalsRestoredError
190   \else
191     \expandafter\@newfont\csname M@font\the\M@count\endcsname{}\{#2}
192     \@for\@i:=#1\do{\expandafter\check@suboption\@i=\@nil

```

If the user calls `\mathfont` and tries to reset the font for a certain class of characters, `mathfont` will issue a warning, and the package will not adjust the font for those characters. Notice



the particularly awkward syntax with the `\csname-\endcsname` pairs. Without this construct,  $\TeX$  wouldn't realize that `\csname if@ \@tempa\endcsname` matches the eventual `\fi`, and the `\@for` loop would break. ( $\TeX$  does not have a smart if-parser!)

```

193     \expandafter\ifx\csname if@ \@tempa\endcsname
194         \csname iftrue\endcsname
195     \M@CharsSetWarning{\@tempa}
196     \else
    Handle the case with a suboption.
197     \if@suboptionpresent
198         \expandafter\edef\csname @ \@tempa shape\endcsname{\@tempb}
199     \fi

```

For both possible shapes, check whether the package has already declared that version of the current font for use in math mode using `\@ifundefined`. If not, do so.

```

200     \def\@tempb{roman}
201     \expandafter\ifx\csname @ \@tempa shape\endcsname\@tempb
202         \@ifundefined{symMroman\the\M@count}{%
203             \DeclareSymbolFont{Mroman\the\M@count}{TU}
204             {\csname M@font\the\M@count\endcsname}{m}{n}}{}
205     \fi
206     \def\@tempb{italic}
207     \expandafter\ifx\csname @ \@tempa shape\endcsname\@tempb
208         \@ifundefined{symMitalic\the\M@count}{%
209             \DeclareSymbolFont{Mitalic\the\M@count}{TU}
210             {\csname M@font\the\M@count\endcsname}{m}{it}}{}
211     \fi

```

And now the magic happens.

```

212     \M@FontChangeInfo{\@tempa}
213     \csname M@ \@tempa @set\endcsname
214     \csname @ \@tempa true\endcsname
215     \fi}
216     \advance\M@count\@ne\relax
217     \message{^^J:: mathfont :: math font for character keyword classes^^J#1^^J%
218         changed to #2^^J^^J}
219     \fi}
220 \@onlypreamble\mathfont

```

## 9 Implementation of Local Font Changes

The general `\newmathfontcommand` macro creates commands that locally change the math font. This macro creates a new math alphabet, and it stores that alphabet in the user-provided control sequence. First, we check to make sure that the user provided a control sequence for the first argument of `\newmathfontcommand`, and if the user did not, `\check@csarg` issues an error.

```

221 \def\check@csarg#1#2#3{%

```

```

222 \ifcat\relax\noexpand#2
223   \def\@tempa{#2}
224   \@newfont\@tempb{}{#3}
225   \expandafter\@newmathfontcommand
226 \else
227   \M@MissingControlSequenceError#1
228 \fi}

```

Now declare the math alphabet.

```

229 \def\@newmathfontcommand#1#2{\expandafter
230   \DeclareMathAlphabet\expandafter{\@tempa}{TU}{\@tempb}{#1}{#2}}
231 \def\newmathfontcommand{\check@csarg\newmathfontcommand}
232 \@onlypreamble\newmathfontcommand

```

Then provide the four default versions.

```

233 \def\newmathrm#1#2{\check@csarg\newmathrm{#1}{#2}{m}{n}}
234 \def\newmathit#1#2{\check@csarg\newmathit{#1}{#2}{m}{it}}
235 \def\newmathbold#1#2{\check@csarg\newmathbold{#1}{#2}{b}{n}}
236 \def\newmathboldit#1#2{\check@csarg\newmathboldit{#1}{#2}{b}{it}}
237 \@onlypreamble\newmathrm
238 \@onlypreamble\newmathit
239 \@onlypreamble\newmathbold
240 \@onlypreamble\newmathboldit

```

## 10 Implementation of Alphanumeric Symbols

Each `\define@<keyword>` macro redefines one of the standard `\math<keyword>` commands. The new versions first check if they are in math mode using `\check@mmode` and then scan all tokens of their argument using `\@tfor`. For each token, the macro calls `\M@<keyword>@<token>`, which evaluates to a `\mathord` symbol in the desired style.

```

241 \def\check@mmode#1#2{%
242   \let\@tempa#2%
243   \ifmmode
244     \expandafter\@tempa
245   \else
246     \leavevmode\M@HModeError#1%
247     \expandafter$\expandafter\@tempa
248   \fi}
249 \def\process@tokens#1#2{\@tfor\@k:=#1\do{\csname M@#2@\@k\endcsname}}
250 \def\define@bb{%
251   \M@CommandInitializeInfo{blackboard bold}
252   \def\mathbb{\check@mmode\mathbb\@mathbb}
253   \def\@mathbb##1{\process@tokens{##1}{bb}}
254 \def\define@cal{%
255   \M@CommandInitializeInfo{calligraphic}
256   \def\mathcal{\check@mmode\mathcal\@mathcal}
257   \def\@mathcal##1{\process@tokens{##1}{cal}}

```

```

258 \def\define@frak{%
259   \M@CommandInitializeInfo{fraktur}
260   \def\mathfrak{\check@mmode\mathfrak\@mathfrak}
261   \def\@mathfrak##1{\process@tokens{##1}{frak}}}
262 \def\define@bcal{%
263   \M@CommandInitializeInfo{bold calligraphic}
264   \def\mathbcal{\check@mmode\mathbcal\@mathbcal}
265   \def\@mathbcal##1{\process@tokens{##1}{bcal}}}
266 \def\define@bfrak{%
267   \M@CommandInitializeInfo{bold fraktur}
268   \def\mathbfrak{\check@mmode\mathbfrak\@mathbfrak}
269   \def\@mathbfrak##1{\process@tokens{##1}{bfrak}}}

```

## 11 Unicode Hex Values

Set capital Latin characters. We use an `\edef` for `\M@upper` because every expansion now will save L<sup>A</sup>T<sub>E</sub>X twenty-six expansions later when it evaluates each `\DeclareMathSymbol`.

```

270 \def\M@upper@set{%
271   \edef\M@upper{M@uppershape\the\M@count}
272   \DeclareMathSymbol{A}{\mathalpha}{\M@upper}{‘A}
273   \DeclareMathSymbol{B}{\mathalpha}{\M@upper}{‘B}
274   \DeclareMathSymbol{C}{\mathalpha}{\M@upper}{‘C}
275   \DeclareMathSymbol{D}{\mathalpha}{\M@upper}{‘D}
276   \DeclareMathSymbol{E}{\mathalpha}{\M@upper}{‘E}
277   \DeclareMathSymbol{F}{\mathalpha}{\M@upper}{‘F}
278   \DeclareMathSymbol{G}{\mathalpha}{\M@upper}{‘G}
279   \DeclareMathSymbol{H}{\mathalpha}{\M@upper}{‘H}
280   \DeclareMathSymbol{I}{\mathalpha}{\M@upper}{‘I}
281   \DeclareMathSymbol{J}{\mathalpha}{\M@upper}{‘J}
282   \DeclareMathSymbol{K}{\mathalpha}{\M@upper}{‘K}
283   \DeclareMathSymbol{L}{\mathalpha}{\M@upper}{‘L}
284   \DeclareMathSymbol{M}{\mathalpha}{\M@upper}{‘M}
285   \DeclareMathSymbol{N}{\mathalpha}{\M@upper}{‘N}
286   \DeclareMathSymbol{O}{\mathalpha}{\M@upper}{‘O}
287   \DeclareMathSymbol{P}{\mathalpha}{\M@upper}{‘P}
288   \DeclareMathSymbol{Q}{\mathalpha}{\M@upper}{‘Q}
289   \DeclareMathSymbol{R}{\mathalpha}{\M@upper}{‘R}
290   \DeclareMathSymbol{S}{\mathalpha}{\M@upper}{‘S}
291   \DeclareMathSymbol{T}{\mathalpha}{\M@upper}{‘T}
292   \DeclareMathSymbol{U}{\mathalpha}{\M@upper}{‘U}
293   \DeclareMathSymbol{V}{\mathalpha}{\M@upper}{‘V}
294   \DeclareMathSymbol{W}{\mathalpha}{\M@upper}{‘W}
295   \DeclareMathSymbol{X}{\mathalpha}{\M@upper}{‘X}
296   \DeclareMathSymbol{Y}{\mathalpha}{\M@upper}{‘Y}
297   \DeclareMathSymbol{Z}{\mathalpha}{\M@upper}{‘Z}}

```

Set minuscule Latin characters.

```

298 \def\M@lower@set{%
299   \edef\M@lower{M\@lowershape\the\M@count}
300   \DeclareMathSymbol{a}{\mathalpha}{\M@lower}{‘a}
301   \DeclareMathSymbol{b}{\mathalpha}{\M@lower}{‘b}
302   \DeclareMathSymbol{c}{\mathalpha}{\M@lower}{‘c}
303   \DeclareMathSymbol{d}{\mathalpha}{\M@lower}{‘d}
304   \DeclareMathSymbol{e}{\mathalpha}{\M@lower}{‘e}
305   \DeclareMathSymbol{f}{\mathalpha}{\M@lower}{‘f}
306   \DeclareMathSymbol{g}{\mathalpha}{\M@lower}{‘g}
307   \DeclareMathSymbol{h}{\mathalpha}{\M@lower}{‘h}
308   \DeclareMathSymbol{i}{\mathalpha}{\M@lower}{‘i}
309   \DeclareMathSymbol{j}{\mathalpha}{\M@lower}{‘j}
310   \DeclareMathSymbol{k}{\mathalpha}{\M@lower}{‘k}
311   \DeclareMathSymbol{l}{\mathalpha}{\M@lower}{‘l}
312   \DeclareMathSymbol{m}{\mathalpha}{\M@lower}{‘m}
313   \DeclareMathSymbol{n}{\mathalpha}{\M@lower}{‘n}
314   \DeclareMathSymbol{o}{\mathalpha}{\M@lower}{‘o}
315   \DeclareMathSymbol{p}{\mathalpha}{\M@lower}{‘p}
316   \DeclareMathSymbol{q}{\mathalpha}{\M@lower}{‘q}
317   \DeclareMathSymbol{r}{\mathalpha}{\M@lower}{‘r}
318   \DeclareMathSymbol{s}{\mathalpha}{\M@lower}{‘s}
319   \DeclareMathSymbol{t}{\mathalpha}{\M@lower}{‘t}
320   \DeclareMathSymbol{u}{\mathalpha}{\M@lower}{‘u}
321   \DeclareMathSymbol{v}{\mathalpha}{\M@lower}{‘v}
322   \DeclareMathSymbol{w}{\mathalpha}{\M@lower}{‘w}
323   \DeclareMathSymbol{x}{\mathalpha}{\M@lower}{‘x}
324   \DeclareMathSymbol{y}{\mathalpha}{\M@lower}{‘y}
325   \DeclareMathSymbol{z}{\mathalpha}{\M@lower}{‘z}}

```

Set capital Greek characters.

```

326 \def\M@greekupper@set{%
327   \edef\M@greekupper{M@greekuppershape\the\M@count}
328   \DeclareMathSymbol{\Alpha}{\mathalpha}{\M@greekupper}{"391}
329   \DeclareMathSymbol{\Beta}{\mathalpha}{\M@greekupper}{"392}
330   \DeclareMathSymbol{\Gamma}{\mathalpha}{\M@greekupper}{"393}
331   \DeclareMathSymbol{\Delta}{\mathalpha}{\M@greekupper}{"394}
332   \DeclareMathSymbol{\Epsilon}{\mathalpha}{\M@greekupper}{"395}
333   \DeclareMathSymbol{\Zeta}{\mathalpha}{\M@greekupper}{"396}
334   \DeclareMathSymbol{\Eta}{\mathalpha}{\M@greekupper}{"397}
335   \DeclareMathSymbol{\Theta}{\mathalpha}{\M@greekupper}{"398}
336   \DeclareMathSymbol{\Iota}{\mathalpha}{\M@greekupper}{"399}
337   \DeclareMathSymbol{\Kappa}{\mathalpha}{\M@greekupper}{"39A}
338   \DeclareMathSymbol{\Lambda}{\mathalpha}{\M@greekupper}{"39B}
339   \DeclareMathSymbol{\Mu}{\mathalpha}{\M@greekupper}{"39C}
340   \DeclareMathSymbol{\Nu}{\mathalpha}{\M@greekupper}{"39D}
341   \DeclareMathSymbol{\Xi}{\mathalpha}{\M@greekupper}{"39E}
342   \DeclareMathSymbol{\Omicron}{\mathalpha}{\M@greekupper}{"39F}
343   \DeclareMathSymbol{\Pi}{\mathalpha}{\M@greekupper}{"3A0}

```

```

344 \DeclareMathSymbol{\Rho}{\mathalpha}{\M@greekupper}{3A1}
345 \DeclareMathSymbol{\Sigma}{\mathalpha}{\M@greekupper}{3A3}
346 \DeclareMathSymbol{\Tau}{\mathalpha}{\M@greekupper}{3A4}
347 \DeclareMathSymbol{\Upsilon}{\mathalpha}{\M@greekupper}{3A5}
348 \DeclareMathSymbol{\Phi}{\mathalpha}{\M@greekupper}{3A6}
349 \DeclareMathSymbol{\Chi}{\mathalpha}{\M@greekupper}{3A7}
350 \DeclareMathSymbol{\Psi}{\mathalpha}{\M@greekupper}{3A8}
351 \DeclareMathSymbol{\Omega}{\mathalpha}{\M@greekupper}{3A9}
352 \DeclareMathSymbol{\varTheta}{\mathalpha}{\M@greekupper}{3F4}
353 \DeclareMathSymbol{\Heta}{\mathalpha}{\M@greekupper}{370}
354 \DeclareMathSymbol{\Sampi}{\mathalpha}{\M@greekupper}{3E0}
355 \DeclareMathSymbol{\Diagamma}{\mathalpha}{\M@greekupper}{3DC}
356 \DeclareMathSymbol{\Koppa}{\mathalpha}{\M@greekupper}{3D8}
357 \DeclareMathSymbol{\Stigma}{\mathalpha}{\M@greekupper}{3DA}
358 \DeclareMathSymbol{\Sho}{\mathalpha}{\M@greekupper}{3F7}
359 \DeclareMathSymbol{\San}{\mathalpha}{\M@greekupper}{3FA}
360 \DeclareMathSymbol{\varSampi}{\mathalpha}{\M@greekupper}{372}
361 \DeclareMathSymbol{\varDiagamma}{\mathalpha}{\M@greekupper}{376}
362 \DeclareMathSymbol{\varKoppa}{\mathalpha}{\M@greekupper}{3DE}

```

Set minuscule Greek characters.

```

363 \def\M@greeklower@set{%
364   \edef\M@greeklower{\M@greeklowershape\the\M@count}
365   \DeclareMathSymbol{\alpha}{\mathalpha}{\M@greeklower}{3B1}
366   \DeclareMathSymbol{\beta}{\mathalpha}{\M@greeklower}{3B2}
367   \DeclareMathSymbol{\gamma}{\mathalpha}{\M@greeklower}{3B3}
368   \DeclareMathSymbol{\delta}{\mathalpha}{\M@greeklower}{3B4}
369   \DeclareMathSymbol{\epsilon}{\mathalpha}{\M@greeklower}{3F5}
370   \DeclareMathSymbol{\zeta}{\mathalpha}{\M@greeklower}{3B6}
371   \DeclareMathSymbol{\eta}{\mathalpha}{\M@greeklower}{3B7}
372   \DeclareMathSymbol{\theta}{\mathalpha}{\M@greeklower}{3B8}
373   \DeclareMathSymbol{\iota}{\mathalpha}{\M@greeklower}{3B9}
374   \DeclareMathSymbol{\kappa}{\mathalpha}{\M@greeklower}{3BA}
375   \DeclareMathSymbol{\lambda}{\mathalpha}{\M@greeklower}{3BB}
376   \DeclareMathSymbol{\mu}{\mathalpha}{\M@greeklower}{3BC}
377   \DeclareMathSymbol{\nu}{\mathalpha}{\M@greeklower}{3BD}
378   \DeclareMathSymbol{\xi}{\mathalpha}{\M@greeklower}{3BE}
379   \DeclareMathSymbol{\omicron}{\mathalpha}{\M@greeklower}{3BF}
380   \DeclareMathSymbol{\pi}{\mathalpha}{\M@greeklower}{3C0}
381   \DeclareMathSymbol{\rho}{\mathalpha}{\M@greeklower}{3C1}
382   \DeclareMathSymbol{\sigma}{\mathalpha}{\M@greeklower}{3C3}
383   \DeclareMathSymbol{\tau}{\mathalpha}{\M@greeklower}{3C4}
384   \DeclareMathSymbol{\upsilon}{\mathalpha}{\M@greeklower}{3C5}
385   \DeclareMathSymbol{\phi}{\mathalpha}{\M@greeklower}{3D5}
386   \DeclareMathSymbol{\chi}{\mathalpha}{\M@greeklower}{3C7}
387   \DeclareMathSymbol{\psi}{\mathalpha}{\M@greeklower}{3C8}
388   \DeclareMathSymbol{\omega}{\mathalpha}{\M@greeklower}{3C9}
389   \DeclareMathSymbol{\varbeta}{\mathalpha}{\M@greeklower}{3D0}

```

```

390 \DeclareMathSymbol{\varepsilon}{\mathalpha}{\M@greeklower}{3B5}
391 \DeclareMathSymbol{\vartheta}{\mathalpha}{\M@greeklower}{3D1}
392 \DeclareMathSymbol{\varrho}{\mathalpha}{\M@greeklower}{3F1}
393 \DeclareMathSymbol{\varsigma}{\mathalpha}{\M@greeklower}{3C2}
394 \DeclareMathSymbol{\varphi}{\mathalpha}{\M@greeklower}{3C6}
395 \DeclareMathSymbol{\heta}{\mathalpha}{\M@greeklower}{371}
396 \DeclareMathSymbol{\sampi}{\mathalpha}{\M@greeklower}{3E1}
397 \DeclareMathSymbol{\diagamma}{\mathalpha}{\M@greeklower}{3DD}
398 \DeclareMathSymbol{\koppa}{\mathalpha}{\M@greeklower}{3D9}
399 \DeclareMathSymbol{\stigma}{\mathalpha}{\M@greeklower}{3DB}
400 \DeclareMathSymbol{\sho}{\mathalpha}{\M@greeklower}{3F8}
401 \DeclareMathSymbol{\san}{\mathalpha}{\M@greeklower}{3FB}
402 \DeclareMathSymbol{\varsampi}{\mathalpha}{\M@greeklower}{373}
403 \DeclareMathSymbol{\vardiagamma}{\mathalpha}{\M@greeklower}{377}
404 \DeclareMathSymbol{\varkoppa}{\mathalpha}{\M@greeklower}{3DF}

```

Set digits.

```

405 \def\M@digits@set{%
406   \edef\M@digits{M@digitsshape\the\M@count}
407   \DeclareMathSymbol{0}{\mathalpha}{\M@digits}{0}
408   \DeclareMathSymbol{1}{\mathalpha}{\M@digits}{1}
409   \DeclareMathSymbol{2}{\mathalpha}{\M@digits}{2}
410   \DeclareMathSymbol{3}{\mathalpha}{\M@digits}{3}
411   \DeclareMathSymbol{4}{\mathalpha}{\M@digits}{4}
412   \DeclareMathSymbol{5}{\mathalpha}{\M@digits}{5}
413   \DeclareMathSymbol{6}{\mathalpha}{\M@digits}{6}
414   \DeclareMathSymbol{7}{\mathalpha}{\M@digits}{7}
415   \DeclareMathSymbol{8}{\mathalpha}{\M@digits}{8}
416   \DeclareMathSymbol{9}{\mathalpha}{\M@digits}{9}}

```

Set symbols.

```

417 \def\M@symbols@set{%
418   \edef\M@symbols{M@symbolsshape\the\M@count}
419   \DeclareMathSymbol{.}{\mathord}{\M@symbols}{2E}
420   \DeclareMathSymbol{@}{\mathord}{\M@symbols}{40}
421   \DeclareMathSymbol{\mathpound}{\mathord}{\M@symbols}{23}
422   \DeclareMathSymbol{\mathdollar}{\mathord}{\M@symbols}{24}
423   \DeclareMathSymbol{\mathpercent}{\mathord}{\M@symbols}{25}
424   \DeclareMathSymbol{\mathand}{\mathord}{\M@symbols}{26}
425   \DeclareMathSymbol{\mathparagraph}{\mathord}{\M@symbols}{B6}
426   \DeclareMathSymbol{\mathsection}{\mathord}{\M@symbols}{A7}
427   \DeclareMathSymbol{\mathsterling}{\mathord}{\M@symbols}{A3}
428   \DeclareMathSymbol{|}{\mathord}{\M@symbols}{7C}
429   \DeclareMathSymbol{\neg}{\mathord}{\M@symbols}{AC}
430   \DeclareMathSymbol{\infty}{\mathord}{\M@symbols}{221E}
431   \DeclareMathSymbol{\partial}{\mathord}{\M@symbols}{2202}
432   \DeclareMathSymbol{\mathbackslash}{\mathord}{\M@symbols}{5C}
433   \DeclareMathSymbol{\degree}{\mathord}{\M@symbols}{B0}
434   \DeclareMathSymbol{\Delta}{\mathord}{\M@symbols}{2206}

```

```

435 \DeclareMathSymbol{'}\mathord{\M@symbols}{2032}
436 \DeclareMathSymbol{"}\mathord{\M@symbols}{2033}
437 \if@operators
438   \DeclareMathSymbol{\sum}\mathop{\M@symbols}{2211}
439   \DeclareMathSymbol{\prod}\mathop{\M@symbols}{220F}
440   \DeclareMathSymbol{\intop}\mathop{\M@symbols}{222B}
441 \fi
442 \DeclareMathSymbol{+}\mathbin{\M@symbols}{2B}
443 \DeclareMathSymbol{-}\mathbin{\M@symbols}{2212}
444 \DeclareMathSymbol{*}\mathbin{\M@symbols}{2A}
445 \DeclareMathSymbol{\times}\mathbin{\M@symbols}{D7}
446 \DeclareMathSymbol{/}\mathbin{\M@symbols}{2215}
447 \DeclareMathSymbol{\div}\mathbin{\M@symbols}{F7}
448 \DeclareMathSymbol{\pm}\mathbin{\M@symbols}{B1}
449 \DeclareMathSymbol{\cdot}\mathbin{\M@symbols}{2219}
450 \DeclareMathSymbol{\bullet}\mathbin{\M@symbols}{2022}
451 \DeclareMathSymbol{\dagger}\mathbin{\M@symbols}{2020}
452 \DeclareMathSymbol{\ddagger}\mathbin{\M@symbols}{2021}
453 \DeclareMathSymbol{\setminus}\mathbin{\M@symbols}{5C}
454 \DeclareMathSymbol{=}\mathrel{\M@symbols}{3D}
455 \DeclareMathSymbol{<}\mathrel{\M@symbols}{3C}
456 \DeclareMathSymbol{>}\mathrel{\M@symbols}{3E}
457 \DeclareMathSymbol{\leq}\mathrel{\M@symbols}{2264}
458 \DeclareMathSymbol{\geq}\mathrel{\M@symbols}{2265}
459 \DeclareMathSymbol{\sim}\mathrel{\M@symbols}{7E}
460 \DeclareMathSymbol{\approx}\mathrel{\M@symbols}{2248}
461 \DeclareMathSymbol{\equiv}\mathrel{\M@symbols}{2261}
462 \DeclareMathSymbol{\mid}\mathrel{\M@symbols}{7C}
463 \DeclareMathSymbol{\parallel}\mathrel{\M@symbols}{2016}
464 \DeclareMathSymbol{:}\mathrel{\M@symbols}{3A}
465 \DeclareMathSymbol{()}\mathopen{\M@symbols}{28}
466 \DeclareMathSymbol{)}\mathclose{\M@symbols}{29}
467 \DeclareMathSymbol{[]}\mathopen{\M@symbols}{5B}
468 \DeclareMathSymbol{[]}\mathclose{\M@symbols}{5D}
469 \DeclareMathSymbol{\leftbrace}\mathopen{\M@symbols}{7B}
470 \DeclareMathSymbol{\rightbrace}\mathclose{\M@symbols}{7D}
471 \DeclareMathSymbol{?}\mathclose{\M@symbols}{3F}
472 \DeclareMathSymbol{!}\mathclose{\M@symbols}{21}
473 \DeclareMathSymbol{,}\mathpunct{\M@symbols}{2C}
474 \DeclareMathSymbol{;}\mathpunct{\M@symbols}{3B}
475 \DeclareMathSymbol{\colon}\mathpunct{\M@symbols}{3A}
476 \let\mathellipsis@undefined
477 \DeclareMathSymbol{\mathellipsis}\mathinner{\M@symbols}{2026}

```

Redefining  $\#$ ,  $\$$ ,  $\%$ , and  $\&$  is tricky. We use `\string` to declare robust versions of these commands that expand to previously declared `\mathpound`, etc. symbols in math mode and retain their standard `\char` definitions otherwise. Other commands that function in both math and horizontal modes such as `\S` or `\dag` also use this technique.

```

478 \count@\escapechar
479 \escapechar\m@ne
480 \edef\#{\noexpand\protect
481   \expandafter\noexpand\csname\string\# \endcsname}
482 \expandafter\def\csname\string\# \endcsname{\ifmmode
483   \mathpound\else\char"23\fi}
484 \edef\${\noexpand\protect
485   \expandafter\noexpand\csname\string\$ \endcsname}
486 \expandafter\def\csname\string\$ \endcsname{\ifmmode
487   \mathdollar\else\char"24\fi}
488 \edef\%{\noexpand\protect
489   \expandafter\noexpand\csname\string\% \endcsname}
490 \expandafter\def\csname\string\% \endcsname{\ifmmode
491   \mathpercent\else\char"25\fi}
492 \edef\&{\noexpand\protect
493   \expandafter\noexpand\csname\string\& \endcsname}
494 \expandafter\def\csname\string\& \endcsname{\ifmmode
495   \mathand\else\char"26\fi}
496 \escapechar\count@

```

Finally a bit of housekeeping.

```

497 \DeclareMathSymbol{\@relbar}{\mathbin}{symbols}{"00}
498 \DeclareMathSymbol{\@Relbar}{\mathrel}{operators}{"3D}
499 \DeclareMathSymbol{\@verticalbar}{\mathord}{symbols}{"6A}
500 \DeclareRobustCommand\relbar{\mathrel{\smash{\@relbar}}}
501 \DeclareRobustCommand\Relbar{\mathrel{\@Relbar}}
502 \DeclareRobustCommand\models{\mathrel{\@verticalbar}\joinrel\Relbar}}

```

Set new operator font.

```

503 \def\M@operator@set{%
504   \edef\operator@font{\noexpand\mathgroup
505     \expandafter\noexpand
506     \csname symM\@operatorshape\the\M@count\endcsname}}

```

Set blackboard bold characters.

```

507 \def\M@bb@set{%
508   \edef\M@bb{M\@bbshape\the\M@count}
509   \DeclareMathSymbol{\M@bb@A}{\mathord}{\M@bb}{"1D538}
510   \DeclareMathSymbol{\M@bb@B}{\mathord}{\M@bb}{"1D539}
511   \DeclareMathSymbol{\M@bb@C}{\mathord}{\M@bb}{"2102}
512   \DeclareMathSymbol{\M@bb@D}{\mathord}{\M@bb}{"1D53B}
513   \DeclareMathSymbol{\M@bb@E}{\mathord}{\M@bb}{"1D53C}
514   \DeclareMathSymbol{\M@bb@F}{\mathord}{\M@bb}{"1D53D}
515   \DeclareMathSymbol{\M@bb@G}{\mathord}{\M@bb}{"1D53E}
516   \DeclareMathSymbol{\M@bb@H}{\mathord}{\M@bb}{"210D}
517   \DeclareMathSymbol{\M@bb@I}{\mathord}{\M@bb}{"1D540}
518   \DeclareMathSymbol{\M@bb@J}{\mathord}{\M@bb}{"1D541}
519   \DeclareMathSymbol{\M@bb@K}{\mathord}{\M@bb}{"1D542}
520   \DeclareMathSymbol{\M@bb@L}{\mathord}{\M@bb}{"1D543}

```



```

521 \DeclareMathSymbol{\M@bb@M}{\mathord}{\M@bb}{1D544}
522 \DeclareMathSymbol{\M@bb@N}{\mathord}{\M@bb}{2115}
523 \DeclareMathSymbol{\M@bb@O}{\mathord}{\M@bb}{1D546}
524 \DeclareMathSymbol{\M@bb@P}{\mathord}{\M@bb}{2119}
525 \DeclareMathSymbol{\M@bb@Q}{\mathord}{\M@bb}{211A}
526 \DeclareMathSymbol{\M@bb@R}{\mathord}{\M@bb}{211D}
527 \DeclareMathSymbol{\M@bb@S}{\mathord}{\M@bb}{1D54A}
528 \DeclareMathSymbol{\M@bb@T}{\mathord}{\M@bb}{1D54B}
529 \DeclareMathSymbol{\M@bb@U}{\mathord}{\M@bb}{1D54C}
530 \DeclareMathSymbol{\M@bb@V}{\mathord}{\M@bb}{1D54D}
531 \DeclareMathSymbol{\M@bb@W}{\mathord}{\M@bb}{1D54E}
532 \DeclareMathSymbol{\M@bb@X}{\mathord}{\M@bb}{1D43F}
533 \DeclareMathSymbol{\M@bb@Y}{\mathord}{\M@bb}{1D550}
534 \DeclareMathSymbol{\M@bb@Z}{\mathord}{\M@bb}{2124}
535 \DeclareMathSymbol{\M@bb@a}{\mathord}{\M@bb}{1D552}
536 \DeclareMathSymbol{\M@bb@b}{\mathord}{\M@bb}{1D553}
537 \DeclareMathSymbol{\M@bb@c}{\mathord}{\M@bb}{1D554}
538 \DeclareMathSymbol{\M@bb@d}{\mathord}{\M@bb}{1D555}
539 \DeclareMathSymbol{\M@bb@e}{\mathord}{\M@bb}{1D556}
540 \DeclareMathSymbol{\M@bb@f}{\mathord}{\M@bb}{1D557}
541 \DeclareMathSymbol{\M@bb@g}{\mathord}{\M@bb}{1D558}
542 \DeclareMathSymbol{\M@bb@h}{\mathord}{\M@bb}{1D559}
543 \DeclareMathSymbol{\M@bb@i}{\mathord}{\M@bb}{1D55A}
544 \DeclareMathSymbol{\M@bb@j}{\mathord}{\M@bb}{1D55B}
545 \DeclareMathSymbol{\M@bb@k}{\mathord}{\M@bb}{1D55C}
546 \DeclareMathSymbol{\M@bb@l}{\mathord}{\M@bb}{1D55D}
547 \DeclareMathSymbol{\M@bb@m}{\mathord}{\M@bb}{1D55E}
548 \DeclareMathSymbol{\M@bb@n}{\mathord}{\M@bb}{1D55F}
549 \DeclareMathSymbol{\M@bb@o}{\mathord}{\M@bb}{1D560}
550 \DeclareMathSymbol{\M@bb@p}{\mathord}{\M@bb}{1D561}
551 \DeclareMathSymbol{\M@bb@q}{\mathord}{\M@bb}{1D562}
552 \DeclareMathSymbol{\M@bb@r}{\mathord}{\M@bb}{1D563}
553 \DeclareMathSymbol{\M@bb@s}{\mathord}{\M@bb}{1D564}
554 \DeclareMathSymbol{\M@bb@t}{\mathord}{\M@bb}{1D565}
555 \DeclareMathSymbol{\M@bb@u}{\mathord}{\M@bb}{1D566}
556 \DeclareMathSymbol{\M@bb@v}{\mathord}{\M@bb}{1D567}
557 \DeclareMathSymbol{\M@bb@w}{\mathord}{\M@bb}{1D568}
558 \DeclareMathSymbol{\M@bb@x}{\mathord}{\M@bb}{1D569}
559 \DeclareMathSymbol{\M@bb@y}{\mathord}{\M@bb}{1D56A}
560 \DeclareMathSymbol{\M@bb@z}{\mathord}{\M@bb}{1D56B}
561 \expandafter\DeclareMathSymbol\expandafter
562   {\csname M@bb@0\endcsname}{\mathord}{\M@bb}{1D7D8}
563 \expandafter\DeclareMathSymbol\expandafter
564   {\csname M@bb@1\endcsname}{\mathord}{\M@bb}{1D7D9}
565 \expandafter\DeclareMathSymbol\expandafter
566   {\csname M@bb@2\endcsname}{\mathord}{\M@bb}{1D7DA}
567 \expandafter\DeclareMathSymbol\expandafter

```

```

568   {\csname M@bb@3\endcsname}{\mathord}{\M@bb}{1D7DB}
569 \expandafter\DeclareMathSymbol\expandafter
570   {\csname M@bb@4\endcsname}{\mathord}{\M@bb}{1D7DC}
571 \expandafter\DeclareMathSymbol\expandafter
572   {\csname M@bb@5\endcsname}{\mathord}{\M@bb}{1D7DD}
573 \expandafter\DeclareMathSymbol\expandafter
574   {\csname M@bb@6\endcsname}{\mathord}{\M@bb}{1D7DE}
575 \expandafter\DeclareMathSymbol\expandafter
576   {\csname M@bb@7\endcsname}{\mathord}{\M@bb}{1D7DF}
577 \expandafter\DeclareMathSymbol\expandafter
578   {\csname M@bb@8\endcsname}{\mathord}{\M@bb}{1D7E0}
579 \expandafter\DeclareMathSymbol\expandafter
580   {\csname M@bb@9\endcsname}{\mathord}{\M@bb}{1D7E1}}

```

Set caligraphic characters.

```

581 \def\M@cal@set{%
582   \edef\M@cal{M@calshape\the\M@count}
583   \DeclareMathSymbol{\M@cal@A}{\mathord}{\M@cal}{1D49C}
584   \DeclareMathSymbol{\M@cal@B}{\mathord}{\M@cal}{212C}
585   \DeclareMathSymbol{\M@cal@C}{\mathord}{\M@cal}{1D49E}
586   \DeclareMathSymbol{\M@cal@D}{\mathord}{\M@cal}{1D49F}
587   \DeclareMathSymbol{\M@cal@E}{\mathord}{\M@cal}{2130}
588   \DeclareMathSymbol{\M@cal@F}{\mathord}{\M@cal}{2131}
589   \DeclareMathSymbol{\M@cal@G}{\mathord}{\M@cal}{1D4A2}
590   \DeclareMathSymbol{\M@cal@H}{\mathord}{\M@cal}{210B}
591   \DeclareMathSymbol{\M@cal@I}{\mathord}{\M@cal}{2110}
592   \DeclareMathSymbol{\M@cal@J}{\mathord}{\M@cal}{1D4A5}
593   \DeclareMathSymbol{\M@cal@K}{\mathord}{\M@cal}{1D4A6}
594   \DeclareMathSymbol{\M@cal@L}{\mathord}{\M@cal}{2112}
595   \DeclareMathSymbol{\M@cal@M}{\mathord}{\M@cal}{2133}
596   \DeclareMathSymbol{\M@cal@N}{\mathord}{\M@cal}{1D4A9}
597   \DeclareMathSymbol{\M@cal@O}{\mathord}{\M@cal}{1D4AA}
598   \DeclareMathSymbol{\M@cal@P}{\mathord}{\M@cal}{1D4AB}
599   \DeclareMathSymbol{\M@cal@Q}{\mathord}{\M@cal}{1D4AC}
600   \DeclareMathSymbol{\M@cal@R}{\mathord}{\M@cal}{211B}
601   \DeclareMathSymbol{\M@cal@S}{\mathord}{\M@cal}{1D4AE}
602   \DeclareMathSymbol{\M@cal@T}{\mathord}{\M@cal}{1D4AF}
603   \DeclareMathSymbol{\M@cal@U}{\mathord}{\M@cal}{1D4B0}
604   \DeclareMathSymbol{\M@cal@V}{\mathord}{\M@cal}{1D4B1}
605   \DeclareMathSymbol{\M@cal@W}{\mathord}{\M@cal}{1D4B2}
606   \DeclareMathSymbol{\M@cal@X}{\mathord}{\M@cal}{1D4B3}
607   \DeclareMathSymbol{\M@cal@Y}{\mathord}{\M@cal}{1D4B4}
608   \DeclareMathSymbol{\M@cal@Z}{\mathord}{\M@cal}{1D4B5}
609   \DeclareMathSymbol{\M@cal@a}{\mathord}{\M@cal}{1D4B6}
610   \DeclareMathSymbol{\M@cal@b}{\mathord}{\M@cal}{1D4B7}
611   \DeclareMathSymbol{\M@cal@c}{\mathord}{\M@cal}{1D4B8}
612   \DeclareMathSymbol{\M@cal@d}{\mathord}{\M@cal}{1D4B9}
613   \DeclareMathSymbol{\M@cal@e}{\mathord}{\M@cal}{212F}

```

```

614 \DeclareMathSymbol{\M@cal@f}{\mathord}{\M@cal}{"1D4BB}
615 \DeclareMathSymbol{\M@cal@g}{\mathord}{\M@cal}{"210A}
616 \DeclareMathSymbol{\M@cal@h}{\mathord}{\M@cal}{"1D4BD}
617 \DeclareMathSymbol{\M@cal@i}{\mathord}{\M@cal}{"1D4BE}
618 \DeclareMathSymbol{\M@cal@j}{\mathord}{\M@cal}{"1D4BF}
619 \DeclareMathSymbol{\M@cal@k}{\mathord}{\M@cal}{"1D4C0}
620 \DeclareMathSymbol{\M@cal@l}{\mathord}{\M@cal}{"1D4C1}
621 \DeclareMathSymbol{\M@cal@m}{\mathord}{\M@cal}{"1D4C2}
622 \DeclareMathSymbol{\M@cal@n}{\mathord}{\M@cal}{"1D4C3}
623 \DeclareMathSymbol{\M@cal@o}{\mathord}{\M@cal}{"2134}
624 \DeclareMathSymbol{\M@cal@p}{\mathord}{\M@cal}{"1D4C5}
625 \DeclareMathSymbol{\M@cal@q}{\mathord}{\M@cal}{"1D4C6}
626 \DeclareMathSymbol{\M@cal@r}{\mathord}{\M@cal}{"1D4C7}
627 \DeclareMathSymbol{\M@cal@s}{\mathord}{\M@cal}{"1D4C8}
628 \DeclareMathSymbol{\M@cal@t}{\mathord}{\M@cal}{"1D4C9}
629 \DeclareMathSymbol{\M@cal@u}{\mathord}{\M@cal}{"1D4CA}
630 \DeclareMathSymbol{\M@cal@v}{\mathord}{\M@cal}{"1D4CB}
631 \DeclareMathSymbol{\M@cal@w}{\mathord}{\M@cal}{"1D4CC}
632 \DeclareMathSymbol{\M@cal@x}{\mathord}{\M@cal}{"1D4CD}
633 \DeclareMathSymbol{\M@cal@y}{\mathord}{\M@cal}{"1D4CE}
634 \DeclareMathSymbol{\M@cal@z}{\mathord}{\M@cal}{"1D4CF}}

```

Set fraktur characters.

```

635 \def\M@frak@set{%
636   \edef\M@frak{M\@frakshape\the\M@count}
637   \DeclareMathSymbol{\M@frak@A}{\mathord}{\M@frak}{"1D504}
638   \DeclareMathSymbol{\M@frak@B}{\mathord}{\M@frak}{"1D505}
639   \DeclareMathSymbol{\M@frak@C}{\mathord}{\M@frak}{"212D}
640   \DeclareMathSymbol{\M@frak@D}{\mathord}{\M@frak}{"1D507}
641   \DeclareMathSymbol{\M@frak@E}{\mathord}{\M@frak}{"1D508}
642   \DeclareMathSymbol{\M@frak@F}{\mathord}{\M@frak}{"1D509}
643   \DeclareMathSymbol{\M@frak@G}{\mathord}{\M@frak}{"1D50A}
644   \DeclareMathSymbol{\M@frak@H}{\mathord}{\M@frak}{"210C}
645   \DeclareMathSymbol{\M@frak@I}{\mathord}{\M@frak}{"2111}
646   \DeclareMathSymbol{\M@frak@J}{\mathord}{\M@frak}{"1D50D}
647   \DeclareMathSymbol{\M@frak@K}{\mathord}{\M@frak}{"1D50E}
648   \DeclareMathSymbol{\M@frak@L}{\mathord}{\M@frak}{"1D50F}
649   \DeclareMathSymbol{\M@frak@M}{\mathord}{\M@frak}{"1D510}
650   \DeclareMathSymbol{\M@frak@N}{\mathord}{\M@frak}{"1D511}
651   \DeclareMathSymbol{\M@frak@O}{\mathord}{\M@frak}{"1D512}
652   \DeclareMathSymbol{\M@frak@P}{\mathord}{\M@frak}{"1D513}
653   \DeclareMathSymbol{\M@frak@Q}{\mathord}{\M@frak}{"1D514}
654   \DeclareMathSymbol{\M@frak@R}{\mathord}{\M@frak}{"212C}
655   \DeclareMathSymbol{\M@frak@S}{\mathord}{\M@frak}{"1D516}
656   \DeclareMathSymbol{\M@frak@T}{\mathord}{\M@frak}{"1D517}
657   \DeclareMathSymbol{\M@frak@U}{\mathord}{\M@frak}{"1D518}
658   \DeclareMathSymbol{\M@frak@V}{\mathord}{\M@frak}{"1D519}
659   \DeclareMathSymbol{\M@frak@W}{\mathord}{\M@frak}{"1D51A}

```

```

660 \DeclareMathSymbol{\M@frak@X}{\mathord}{\M@frak}{1D51B}
661 \DeclareMathSymbol{\M@frak@Y}{\mathord}{\M@frak}{1D51C}
662 \DeclareMathSymbol{\M@frak@Z}{\mathord}{\M@frak}{2128}
663 \DeclareMathSymbol{\M@frak@a}{\mathord}{\M@frak}{1D51E}
664 \DeclareMathSymbol{\M@frak@b}{\mathord}{\M@frak}{1D51F}
665 \DeclareMathSymbol{\M@frak@c}{\mathord}{\M@frak}{1D520}
666 \DeclareMathSymbol{\M@frak@d}{\mathord}{\M@frak}{1D521}
667 \DeclareMathSymbol{\M@frak@e}{\mathord}{\M@frak}{1D522}
668 \DeclareMathSymbol{\M@frak@f}{\mathord}{\M@frak}{1D523}
669 \DeclareMathSymbol{\M@frak@g}{\mathord}{\M@frak}{1D524}
670 \DeclareMathSymbol{\M@frak@h}{\mathord}{\M@frak}{1D525}
671 \DeclareMathSymbol{\M@frak@i}{\mathord}{\M@frak}{1D526}
672 \DeclareMathSymbol{\M@frak@j}{\mathord}{\M@frak}{1D527}
673 \DeclareMathSymbol{\M@frak@k}{\mathord}{\M@frak}{1D528}
674 \DeclareMathSymbol{\M@frak@l}{\mathord}{\M@frak}{1D529}
675 \DeclareMathSymbol{\M@frak@m}{\mathord}{\M@frak}{1D52A}
676 \DeclareMathSymbol{\M@frak@n}{\mathord}{\M@frak}{1D52B}
677 \DeclareMathSymbol{\M@frak@o}{\mathord}{\M@frak}{1D52C}
678 \DeclareMathSymbol{\M@frak@p}{\mathord}{\M@frak}{1D52D}
679 \DeclareMathSymbol{\M@frak@q}{\mathord}{\M@frak}{1D52E}
680 \DeclareMathSymbol{\M@frak@r}{\mathord}{\M@frak}{1D52F}
681 \DeclareMathSymbol{\M@frak@s}{\mathord}{\M@frak}{1D530}
682 \DeclareMathSymbol{\M@frak@t}{\mathord}{\M@frak}{1D531}
683 \DeclareMathSymbol{\M@frak@u}{\mathord}{\M@frak}{1D532}
684 \DeclareMathSymbol{\M@frak@v}{\mathord}{\M@frak}{1D533}
685 \DeclareMathSymbol{\M@frak@w}{\mathord}{\M@frak}{1D534}
686 \DeclareMathSymbol{\M@frak@x}{\mathord}{\M@frak}{1D535}
687 \DeclareMathSymbol{\M@frak@y}{\mathord}{\M@frak}{1D536}
688 \DeclareMathSymbol{\M@frak@z}{\mathord}{\M@frak}{1D537}

```

Set bold caligraphic characters.

```

689 \def\M@bcal@set{%
690   \edef\M@bcal{M\@bcalshape\the\M@count}
691   \DeclareMathSymbol{\M@bcal@A}{\mathord}{\M@bcal}{1D4D0}
692   \DeclareMathSymbol{\M@bcal@B}{\mathord}{\M@bcal}{1D4D1}
693   \DeclareMathSymbol{\M@bcal@C}{\mathord}{\M@bcal}{1D4D2}
694   \DeclareMathSymbol{\M@bcal@D}{\mathord}{\M@bcal}{1D4D3}
695   \DeclareMathSymbol{\M@bcal@E}{\mathord}{\M@bcal}{1D4D4}
696   \DeclareMathSymbol{\M@bcal@F}{\mathord}{\M@bcal}{1D4D5}
697   \DeclareMathSymbol{\M@bcal@G}{\mathord}{\M@bcal}{1D4D6}
698   \DeclareMathSymbol{\M@bcal@H}{\mathord}{\M@bcal}{1D4D7}
699   \DeclareMathSymbol{\M@bcal@I}{\mathord}{\M@bcal}{1D4D8}
700   \DeclareMathSymbol{\M@bcal@J}{\mathord}{\M@bcal}{1D4D9}
701   \DeclareMathSymbol{\M@bcal@K}{\mathord}{\M@bcal}{1D4DA}
702   \DeclareMathSymbol{\M@bcal@L}{\mathord}{\M@bcal}{1D4DB}
703   \DeclareMathSymbol{\M@bcal@M}{\mathord}{\M@bcal}{1D4DC}
704   \DeclareMathSymbol{\M@bcal@N}{\mathord}{\M@bcal}{1D4DD}
705   \DeclareMathSymbol{\M@bcal@O}{\mathord}{\M@bcal}{1D4DE}

```

```

706 \DeclareMathSymbol{\M@bcal@P}{\mathord}{\M@bcal}{1D4DF}
707 \DeclareMathSymbol{\M@bcal@Q}{\mathord}{\M@bcal}{1D4E0}
708 \DeclareMathSymbol{\M@bcal@R}{\mathord}{\M@bcal}{1D4E1}
709 \DeclareMathSymbol{\M@bcal@S}{\mathord}{\M@bcal}{1D4E2}
710 \DeclareMathSymbol{\M@bcal@T}{\mathord}{\M@bcal}{1D4E3}
711 \DeclareMathSymbol{\M@bcal@U}{\mathord}{\M@bcal}{1D4E4}
712 \DeclareMathSymbol{\M@bcal@V}{\mathord}{\M@bcal}{1D4E5}
713 \DeclareMathSymbol{\M@bcal@W}{\mathord}{\M@bcal}{1D4E6}
714 \DeclareMathSymbol{\M@bcal@X}{\mathord}{\M@bcal}{1D4E7}
715 \DeclareMathSymbol{\M@bcal@Y}{\mathord}{\M@bcal}{1D4E8}
716 \DeclareMathSymbol{\M@bcal@Z}{\mathord}{\M@bcal}{1D4E9}
717 \DeclareMathSymbol{\M@bcal@a}{\mathord}{\M@bcal}{1D4EA}
718 \DeclareMathSymbol{\M@bcal@b}{\mathord}{\M@bcal}{1D4EB}
719 \DeclareMathSymbol{\M@bcal@c}{\mathord}{\M@bcal}{1D4EC}
720 \DeclareMathSymbol{\M@bcal@d}{\mathord}{\M@bcal}{1D4ED}
721 \DeclareMathSymbol{\M@bcal@e}{\mathord}{\M@bcal}{1D4EE}
722 \DeclareMathSymbol{\M@bcal@f}{\mathord}{\M@bcal}{1D4EF}
723 \DeclareMathSymbol{\M@bcal@g}{\mathord}{\M@bcal}{1D4F0}
724 \DeclareMathSymbol{\M@bcal@h}{\mathord}{\M@bcal}{1D4F1}
725 \DeclareMathSymbol{\M@bcal@i}{\mathord}{\M@bcal}{1D4F2}
726 \DeclareMathSymbol{\M@bcal@j}{\mathord}{\M@bcal}{1D4F3}
727 \DeclareMathSymbol{\M@bcal@k}{\mathord}{\M@bcal}{1D4F4}
728 \DeclareMathSymbol{\M@bcal@l}{\mathord}{\M@bcal}{1D4F5}
729 \DeclareMathSymbol{\M@bcal@m}{\mathord}{\M@bcal}{1D4F6}
730 \DeclareMathSymbol{\M@bcal@n}{\mathord}{\M@bcal}{1D4F7}
731 \DeclareMathSymbol{\M@bcal@o}{\mathord}{\M@bcal}{1D4F8}
732 \DeclareMathSymbol{\M@bcal@p}{\mathord}{\M@bcal}{1D4F9}
733 \DeclareMathSymbol{\M@bcal@q}{\mathord}{\M@bcal}{1D4FA}
734 \DeclareMathSymbol{\M@bcal@r}{\mathord}{\M@bcal}{1D4FB}
735 \DeclareMathSymbol{\M@bcal@s}{\mathord}{\M@bcal}{1D4FC}
736 \DeclareMathSymbol{\M@bcal@t}{\mathord}{\M@bcal}{1D4FD}
737 \DeclareMathSymbol{\M@bcal@u}{\mathord}{\M@bcal}{1D4FE}
738 \DeclareMathSymbol{\M@bcal@v}{\mathord}{\M@bcal}{1D4FF}
739 \DeclareMathSymbol{\M@bcal@w}{\mathord}{\M@bcal}{1D500}
740 \DeclareMathSymbol{\M@bcal@x}{\mathord}{\M@bcal}{1D501}
741 \DeclareMathSymbol{\M@bcal@y}{\mathord}{\M@bcal}{1D502}
742 \DeclareMathSymbol{\M@bcal@z}{\mathord}{\M@bcal}{1D503}

```

Set bold fraktur characters.

```

743 \def\M@bfrak@set{%
744   \edef\M@bfrak{\M@bfrakshape\the\M@count}
745   \DeclareMathSymbol{\M@bfrak@A}{\mathord}{\M@bfrak}{1D56C}
746   \DeclareMathSymbol{\M@bfrak@B}{\mathord}{\M@bfrak}{1D56D}
747   \DeclareMathSymbol{\M@bfrak@C}{\mathord}{\M@bfrak}{1D56E}
748   \DeclareMathSymbol{\M@bfrak@D}{\mathord}{\M@bfrak}{1D56F}
749   \DeclareMathSymbol{\M@bfrak@E}{\mathord}{\M@bfrak}{1D570}
750   \DeclareMathSymbol{\M@bfrak@F}{\mathord}{\M@bfrak}{1D571}
751   \DeclareMathSymbol{\M@bfrak@G}{\mathord}{\M@bfrak}{1D572}

```

```

752 \DeclareMathSymbol{\M@bfrac@H}{\mathord}{\M@bfrac}{"1D573}
753 \DeclareMathSymbol{\M@bfrac@I}{\mathord}{\M@bfrac}{"1D574}
754 \DeclareMathSymbol{\M@bfrac@J}{\mathord}{\M@bfrac}{"1D575}
755 \DeclareMathSymbol{\M@bfrac@K}{\mathord}{\M@bfrac}{"1D576}
756 \DeclareMathSymbol{\M@bfrac@L}{\mathord}{\M@bfrac}{"1D577}
757 \DeclareMathSymbol{\M@bfrac@M}{\mathord}{\M@bfrac}{"1D578}
758 \DeclareMathSymbol{\M@bfrac@N}{\mathord}{\M@bfrac}{"1D579}
759 \DeclareMathSymbol{\M@bfrac@O}{\mathord}{\M@bfrac}{"1D57A}
760 \DeclareMathSymbol{\M@bfrac@P}{\mathord}{\M@bfrac}{"1D57B}
761 \DeclareMathSymbol{\M@bfrac@Q}{\mathord}{\M@bfrac}{"1D57C}
762 \DeclareMathSymbol{\M@bfrac@R}{\mathord}{\M@bfrac}{"1D57D}
763 \DeclareMathSymbol{\M@bfrac@S}{\mathord}{\M@bfrac}{"1D57E}
764 \DeclareMathSymbol{\M@bfrac@T}{\mathord}{\M@bfrac}{"1D57F}
765 \DeclareMathSymbol{\M@bfrac@U}{\mathord}{\M@bfrac}{"1D580}
766 \DeclareMathSymbol{\M@bfrac@V}{\mathord}{\M@bfrac}{"1D581}
767 \DeclareMathSymbol{\M@bfrac@W}{\mathord}{\M@bfrac}{"1D582}
768 \DeclareMathSymbol{\M@bfrac@X}{\mathord}{\M@bfrac}{"1D583}
769 \DeclareMathSymbol{\M@bfrac@Y}{\mathord}{\M@bfrac}{"1D584}
770 \DeclareMathSymbol{\M@bfrac@Z}{\mathord}{\M@bfrac}{"1D585}
771 \DeclareMathSymbol{\M@bfrac@a}{\mathord}{\M@bfrac}{"1D586}
772 \DeclareMathSymbol{\M@bfrac@b}{\mathord}{\M@bfrac}{"1D587}
773 \DeclareMathSymbol{\M@bfrac@c}{\mathord}{\M@bfrac}{"1D588}
774 \DeclareMathSymbol{\M@bfrac@d}{\mathord}{\M@bfrac}{"1D589}
775 \DeclareMathSymbol{\M@bfrac@e}{\mathord}{\M@bfrac}{"1D58A}
776 \DeclareMathSymbol{\M@bfrac@f}{\mathord}{\M@bfrac}{"1D58B}
777 \DeclareMathSymbol{\M@bfrac@g}{\mathord}{\M@bfrac}{"1D58C}
778 \DeclareMathSymbol{\M@bfrac@h}{\mathord}{\M@bfrac}{"1D58D}
779 \DeclareMathSymbol{\M@bfrac@i}{\mathord}{\M@bfrac}{"1D58E}
780 \DeclareMathSymbol{\M@bfrac@j}{\mathord}{\M@bfrac}{"1D58F}
781 \DeclareMathSymbol{\M@bfrac@k}{\mathord}{\M@bfrac}{"1D590}
782 \DeclareMathSymbol{\M@bfrac@l}{\mathord}{\M@bfrac}{"1D591}
783 \DeclareMathSymbol{\M@bfrac@m}{\mathord}{\M@bfrac}{"1D592}
784 \DeclareMathSymbol{\M@bfrac@n}{\mathord}{\M@bfrac}{"1D593}
785 \DeclareMathSymbol{\M@bfrac@o}{\mathord}{\M@bfrac}{"1D594}
786 \DeclareMathSymbol{\M@bfrac@p}{\mathord}{\M@bfrac}{"1D595}
787 \DeclareMathSymbol{\M@bfrac@q}{\mathord}{\M@bfrac}{"1D596}
788 \DeclareMathSymbol{\M@bfrac@r}{\mathord}{\M@bfrac}{"1D597}
789 \DeclareMathSymbol{\M@bfrac@s}{\mathord}{\M@bfrac}{"1D598}
790 \DeclareMathSymbol{\M@bfrac@t}{\mathord}{\M@bfrac}{"1D599}
791 \DeclareMathSymbol{\M@bfrac@u}{\mathord}{\M@bfrac}{"1D59A}
792 \DeclareMathSymbol{\M@bfrac@v}{\mathord}{\M@bfrac}{"1D59B}
793 \DeclareMathSymbol{\M@bfrac@w}{\mathord}{\M@bfrac}{"1D59C}
794 \DeclareMathSymbol{\M@bfrac@x}{\mathord}{\M@bfrac}{"1D59D}
795 \DeclareMathSymbol{\M@bfrac@y}{\mathord}{\M@bfrac}{"1D59E}
796 \DeclareMathSymbol{\M@bfrac@z}{\mathord}{\M@bfrac}{"1D59F}}

```

Provide the command to reset macros and conclude.

```

797 \def\restoremathinternals{%

```

```

798 \M@RestoreInternalsInfo
799 \let\set@mathchar\@set@mathchar
800 \let\set@mathsymbol\@set@mathsymbol}

```

Unless the user specified the `packages` option, `mathfont` will reset the internal math symbol commands when the user loads another package.

```

801 \if@packages
802 \else
803   \let\@usepackage\usepackage
804   \def\usepackage{\restoremathinternals
805     \let\usepackage\@usepackage
806     \usepackage}
807 \fi

```

# 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	
\#	480–482
\\$	484–486
\%	488–490
\&	492–494
\@@end	100
\@@set@mathchar	116, 188, 799
\@@set@mathsymbol	117, 800
\@usepackage	803, 805
\@Relbar	498, 501
\@bbshape	148, 508
\@bcalshape	151, 690
\@bfrakshape	152, 744
\@calshape	149, 582
\@digitssshape	145, 406
\@for	167, 171, 179, 192
\@frakshape	150, 636
\@greek lowershape	144, 364
\@greek uppershape	143, 327
\@ifnextchar	185
\@ifpackageloaded	109
\@ifundefined	202, 208
\@ lowershape	142, 299
\@mathbb	252, 253
\@mathbcal	264, 265
\@mathbfrak	268, 269
\@mathcal	256, 257
\@mathfont	185–187
\@mathfrak	260, 261
\@newfont	113, 191, 224
\@newmathfontcommand	225, 229
\@nil	154, 184, 192
\@onlypreamble	220, 232, 237–240
\@operatorsfalse	8
\@operatorshape	147, 506
\@operatorstrue	5, 7
\@packagestrue	6
\@relbar	497, 500
\@suboptionpresentfalse	159
\@suboptionpresenttrue	163
\@symbolssshape	146, 418
\@temperror	166, 169, 173, 176, 178, 181, 183
\@tfor	249
\@uppershape	141, 271
\@verticalbar	499, 502
<b>A</b>	
\Alpha	328
\alpha	365
\approx	460
<b>B</b>	
\Beta	329
\beta	366
\bullet	450
<b>C</b>	
\cdot	449
\char	483, 487, 491, 495
\check@csarg	221, 231, 233–236
\check@mmode	241, 252, 256, 260, 264, 268
\check@option@valid	157, 165
\check@suboption	154, 192
\check@suboption@valid	162, 177
\Chi	349
\chi	386
\colon	475
\CurrentOption	10
<b>D</b>	
\dagger	451
\ddagger	452
\DeclareMathAlphabet	230
\DeclareSymbolFont	203, 209
\define@bb	250
\define@bcal	262
\define@bfrak	266
\define@cal	254
\define@frak	258
\degree	433
\Delta	331, 434
\delta	368
\Diagamma	355
\diagamma	397



\div .....	447	\lambda .....	375
<b>E</b>		\leavevmode .....	246
\Epsilon .....	332	\leftbrace .....	469
\epsilon .....	369	\leq .....	457
\equiv .....	461	<b>M</b>	
\escapechar .....	478, 479, 496	\M@bb@set .....	507
\Eta .....	334	\M@bcal@set .....	689
\eta .....	371	\M@bfrak@set .....	743
<b>G</b>		\M@cal@set .....	581
\Gamma .....	330	\M@CharsSetWarning .....	33, 195
\gamma .....	367	\M@CommandInitializeInfo .....	24, 251, 255, 259, 263, 267
\geq .....	458	\M@digits@set .....	405
\gobble@equals .....	184	\M@FontChangeInfo .....	14, 212
<b>H</b>		\M@frak@set .....	635
\Heta .....	353	\M@greeklower@set .....	363
\heta .....	395	\M@greekupper@set .....	326
<b>I</b>		\M@HModeError .....	83, 246
\if@bb .....	135	\M@InternalsRestoredError .....	59, 189
\if@bcal .....	138	\M@InvalidOptionError .....	51, 166, 178
\if@bfrak .....	139	\M@lower@set .....	298
\if@cal .....	136	\M@MissingControlSequenceError ..	78, 227
\if@digits .....	133	\M@NewFontCommandInfo .....	19
\if@frak .....	137	\M@NoMathError .....	36, 111
\if@greeklower .....	131	\M@operator@set .....	503
\if@greekupper .....	130	\M@RestoreInternalsInfo .....	30, 798
\if@lower .....	129	\M@SetInternalsInfo .....	27, 115
\if@operator .....	132	\M@symbols@set .....	417
\if@operators .....	4, 437	\M@upper@set .....	270
\if@packages .....	3, 801	\M@XeTeXLuaTeXError .....	92, 105, 108
\if@suboptionpresent .....	140, 197	\mathand .....	424, 495
\if@symbols .....	134	\mathbackslash .....	432
\if@upper .....	128	\mathbb .....	252
\infty .....	430	\mathbcal .....	264
\intop .....	440	\mathbfrak .....	268
\Iota .....	336	\mathcal .....	256
\iota .....	373	\mathchar@type .....	121, 125
<b>K</b>		\mathdollar .....	422, 487
\Kappa .....	337	\mathellipsis .....	476, 477
\kappa .....	374	\mathfont .....	52, 65, 69, 74, 75, 185, 220
\Koppa .....	356	\mathfrak .....	260
\koppa .....	398	\mathgroup .....	504
<b>L</b>		\mathparagraph .....	425
\Lambda .....	338	\mathpercent .....	423, 491
		\mathpound .....	421, 483
		\mathsection .....	426
		\mathsterling .....	427

<code>\mid</code> .....	462	<code>\set@mathchar</code> ....	28, 31, 116, 118, 188, 799
<code>\models</code> .....	502	<code>\set@mathsymbol</code> .....	29, 32, 117, 122, 800
<code>\Mu</code> .....	339	<code>\setminus</code> .....	453
<code>\mu</code> .....	376	<code>\Sho</code> .....	358
<b>N</b>		<code>\sho</code> .....	400
<code>\neg</code> .....	429	<code>\Sigma</code> .....	345
<code>\newmathbold</code> .....	235, 239	<code>\sigma</code> .....	382
<code>\newmathboldit</code> .....	236, 240	<code>\sim</code> .....	459
<code>\newmathfontcommand</code> .....	231, 232	<code>\Stigma</code> .....	357
<code>\newmathit</code> .....	234, 238	<code>\stigma</code> .....	399
<code>\newmathrm</code> .....	233, 237	<code>\strip@equals</code> .....	153, 161
<code>\Nu</code> .....	340	<code>\sum</code> .....	438
<code>\nu</code> .....	377	<b>T</b>	
<b>O</b>		<code>\Tau</code> .....	346
<code>\Omega</code> .....	351	<code>\tau</code> .....	383
<code>\omega</code> .....	388	<code>\Theta</code> .....	335
<code>\Omicron</code> .....	342	<code>\theta</code> .....	372
<code>\omicron</code> .....	379	<code>\times</code> .....	445
<code>\operator@font</code> .....	504	<b>U</b>	
<b>P</b>		<code>\Umathchardef</code> .....	103, 125
<code>\parallel</code> .....	463	<code>\Umathcode</code> .....	101, 121
<code>\partial</code> .....	431	<code>\Upsilon</code> .....	347
<code>\Phi</code> .....	348	<code>\upsilon</code> .....	384
<code>\phi</code> .....	385	<code>\usepackage</code> .....	77, 803–806
<code>\Pi</code> .....	343	<b>V</b>	
<code>\pi</code> .....	380	<code>\varbeta</code> .....	389
<code>\pm</code> .....	448	<code>\varDiagamma</code> .....	361
<code>\process@tokens</code> 249, 253, 257, 261, 265, 269		<code>\vardiagamma</code> .....	403
<code>\prod</code> .....	439	<code>\varepsilon</code> .....	390
<code>\Psi</code> .....	350	<code>\varKoppa</code> .....	362
<code>\psi</code> .....	387	<code>\varkoppa</code> .....	404
<b>R</b>		<code>\varphi</code> .....	394
<code>\Relbar</code> .....	501, 502	<code>\varrho</code> .....	392
<code>\relbar</code> .....	500	<code>\varSampi</code> .....	360
<code>\RequirePackage</code> .....	112	<code>\varsampi</code> .....	402
<code>\restoremathinternals</code> .....	71, 797, 804	<code>\varsigma</code> .....	393
<code>\Rho</code> .....	344	<code>\varTheta</code> .....	352
<code>\rho</code> .....	381	<code>\vartheta</code> .....	391
<code>\rightbrace</code> .....	470	<b>X</b>	
<b>S</b>		<code>\Xi</code> .....	341
<code>\Sampi</code> .....	354	<code>\xi</code> .....	378
<code>\sampi</code> .....	396	<b>Z</b>	
<code>\San</code> .....	359	<code>\Zeta</code> .....	333
<code>\san</code> .....	401	<code>\zeta</code> .....	370