

Package `mathfont` v. 1.4 Documentation

Conrad Kosowsky

April 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, including punctuation. Crucially, `mathfont` is compatible with both Lua \LaTeX and Xe \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. Donald Knuth originally designed \TeX to support fonts created with 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 Xe \TeX and Hans Hagen, et al.'s Lua \TeX significantly extend the font-loading capabilities of \TeX .¹ 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.² 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.

The `mathfont` package applies `fontspec`'s advances in font selection to math-mode typesetting, and this documentation explains `mathfont` as follows. Section 1 presents the basic

¹Information on Xe \TeX is available at <https://tug.org/xetex/>, and information on Lua \TeX is available at the official website for Lua \TeX : <http://www.luaotex.org/>.

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

functionality and related packages. Sections 2 and 3 explain how to use the default and local font-change commands, and users in a hurry will find the most important information here. 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 lay out the \LaTeX code for default and local font changes, respectively. Section 10 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 is 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 \LaTeX 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 \LaTeX 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^AT_EX’s `\PassOptionsToPackage` command.

Compatibility with LuaL^AT_EX 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 `"`.³ For example, the code

```
\catcode'\="=\active
\def"#1{\ifmmode
  \kern<dimension>\relax #1\kern<other dimension>\relax
\else
  \char'\#"#1%
\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_EX’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.⁴ 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^AT_EX; it can adjust the font of basic mathematical symbols such as those in tables 5 and 6; 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.⁵

Users who want to stick with pdfL^AT_EX should consider Jean-François Burnol’s `mathastext`

³The problem isn’t one of ability—LuaT_EX’s programming capabilities tend to outweigh those of X_ƎT_EX. Rather, I’m unfamiliar with most of the workings of LuaT_EX. 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.

⁴Andrew Gilbert Moschou, “`mathspec`—Specify arbitrary fonts for mathematics in X_ƎT_EX,” <https://ctan.org/pkg/mathspec>.

⁵Will Robertson, “`unicode-math`—Unicode mathematics support for X_ƎT_EX and LuaT_EX,” <https://ctan.org/pkg/unicode-math>.

as a useful alternative to `mathfont`.⁶ This package allows the user to specify the math-mode 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^AT_EX kernel in conjunction with Plain T_EX. Whereas `mathfont` works exclusively in the context of unicode fonts, `mathastext` was designed for the T1 and related encodings of Plain T_EX and L^AT_EX, 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 for a class of characters, `mathfont` allows for that class only the local font changes outlined in section 3.

2 Setting the Default Font

The `\mathfont` command sets the default font for certain classes of characters. Its structure is given by

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

where the *optional character classes* can be any set of keywords from Tables 1 and 2 separated by commas, and the *font name* can be any OpenType or TrueType font in a directory searchable by T_EX. 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. To change both math and text fonts simultaneously, the package provides the command

$$\backslash\setfont\{\langle font\ name\rangle\},$$

which calls both `\mathfont` and `fontspec`'s `\setmainfont` using the *font name* as arguments. Most users will probably find this command sufficient for most applications. Both `\mathfont` and `\setfont` should appear only in the document preamble.

The user should specify any keyword optional arguments for `\mathfont` as entries in a comma-separated list. The order is irrelevant, and spaces between entries are permitted. For each item in the list, the package will change the math-mode font for that class of characters to the mandatory argument. Leaving out the optional argument will cause the command to revert to its default behavior, and for most applications, that will be sufficient. 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 such as `\log` or `\sin`, and all symbols from Tables 5 and 6 to the font in the mandatory argument.

⁶Jean-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_EX's internal mathematics spacing. In fact the opposite is true: `mathastext` goes to significant lengths to preserve and even extend rules for space between various math-mode characters.

Table 1: Math Alphabet Characters

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

(Specifically, all keywords from Table 1 and the `symbol` keyword from Table 2.) In this example, the new font 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. As of version 1.4, `mathfont` ignores all spaces in its optional argument, so the above code snippet is equivalent to

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

Technical point: `mathfont` expands its optional argument inside an `\edef`, so the argument can contain any number of fully expandable control sequences. Users should also 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

Table 2: Other Math Symbols

Keyword	Meaning
<code>symbols</code>	Basic mathematical symbols (includes punctuation)
<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

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

$$\backslash\math\langle keyword \rangle\{\langle argument \rangle\}.$$

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

$$\backslash\mathfont[bcal]\{\text{STIXGeneral}\}$$

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 \TeX '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 \TeX 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

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

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

$$\backslash\newmathrm\{\backslash\matharial\}\{\text{Arial}\}$$

would create the macro

$$\backslash\matharial\{\langle argument \rangle\},$$

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

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, $\text{\LaTeX} 2_{\epsilon}$ 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, `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

$$\backslash\text{newmathfontcommand}\{\langle\text{control sequence}\rangle\}\{\langle\text{font name}\rangle\}\{\langle\text{series}\rangle\}\{\langle\text{shape}\rangle\},$$

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 $\text{\LaTeX} 2_{\epsilon}$, `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.

Table 4: Greek Character Control Sequences

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

*Starred Modern Greek and all Ancient Greek letters are not provided by default and are defined only when the user calls `\mathfont` for `greekupper` and/or `greeklower` 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 package acts on the integral, summation, and product symbols. As stated previously, users

Table 5: Characters Corresponding to the `symbols` Keyword

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

*If the package option `operators` is enabled.

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

Character	Control Sequence	Class	Character	Control Sequence	Class
\	<code>\mathbackslash</code>	<code>\mathord</code>	{	<code>\leftbrace</code>	<code>\mathopen</code>
°	<code>\degree</code>	<code>\mathord</code>	}	<code>\rightbrace</code>	<code>\mathclose</code>
,	<code>\comma</code>	<code>\mathord</code>			

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 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 five new control sequences listed in Table 6. The `\degree` command functions as the unicode alternative to the `$^\circ$` construction, and the other four 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.⁷ The package also provides an additional comma character to typeset large integers. \TeX users have likely noticed the extra space after commas in math mode, e.g. 10,000 versus 10,000, and `mathfont`’s `\comma` solves this problem. Here the first ten thousand uses a standard `,` while the second uses `\comma`. In other words, the control sequence typesets a comma in math mode but does not insert the extra space following it. I recommend using `\comma` basically only for large integers, but I leave the ultimate decision up to the user.

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 $X_{\text{f}}\TeX$ or $\text{Lua}\TeX$ ” 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 $X_{\text{f}}\TeX$ and $\text{Lua}\TeX$ primitives `\Umathcode` and `\Umathchardef` are defined, so if for

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

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 `XYTeX` or `LuaTeX` distributions may 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 when `XYTeX` and `LuaTeX` introduced these primitives.⁸

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, `LATeX`'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.

The “internal commands restored” error arises when the user calls `\mathfont` in a situation where the package already restored the small portion of the `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 `LATeX` kernel can do so with the `\restoremathinternals` command. 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 `TeX` StackExchange, Share`LATeX`, and Wikibooks's `LATeX` wiki are often invaluable in dealing with `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. I welcome emails about `mathfont` and will make every effort to write back to correspondence 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

⁸However, the manual for a beta version of `LuaTeX`, 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>

`\if@packages` boolean from false to true.

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{mathfont}[2019/04/01 v. 1.4 Package mathfont]
3 \newif\if@packages
4 \newif\if@operators
5 \DeclareOption{packages}{\@packagestrue}
6 \DeclareOption{operators}{\@operatorstrue}
7 \DeclareOption{no-operators}{\@operatorsfalse}
8 \DeclareOption*{\PackageError{mathfont}
9   {Option "\CurrentOption" unknown}
10  {I'm ignoring that option because\MessageBreak
11   I don't know what it means.}}
12 \ProcessOptions*
```

Some error and informational messages.

```
13 \def\M@FontChangeInfo#1{\PackageInfo{mathfont}
14   {Changing #1 characters to\MessageBreak
15   math font \csname M@font\the\M@count\endcsname\MessageBreak
16   with \csname @#1 shape\endcsname\MessageBreak
17   shape}}
18 \def\M@NewFontCommandInfo#1#2#3#4{\PackageInfo{mathfont}
19   {Creating math alphabet\MessageBreak
20   command \string#1 using #2\MessageBreak
21   with series #3 and shape\MessageBreak
22   #4}}
23 \def\M@CommandInitializeInfo#1{\PackageInfo{mathfont}
24   {Initializing #1 font-change\MessageBreak
25   command}}
26 \def\M@SetInternalsInfo{\PackageInfo{mathfont}
27   {Adjusting \string\set@mathchar\space and\MessageBreak
28   \string\set@mathsymbol}}
29 \def\M@RestoreInternalsInfo{\PackageInfo{mathfont}
30   {Restoring \string\set@mathchar\space and\MessageBreak
31   \string\set@mathsymbol}}
32 \def\M@CharsSetWarning#1{\PackageWarning{mathfont}
33   {Font for #1 chars has already\MessageBreak
34   been set}}
35 \def\M@NoMathError{\PackageError{mathfont}
36   {Package fontspec was loaded\MessageBreak
37   without the "no-math" option}
38   {This isn't really an error--\MessageBreak
39   it's fine to load fontspec\MessageBreak
40   without "no-math." However,\MessageBreak
41   strange things could happen,\MessageBreak
42   so beware of any sudden and\MessageBreak
43   unexpected font changes. To\MessageBreak
44   resolve this error message,\MessageBreak
45   load fontspec with the "no-\MessageBreak
```

```

46 math" option. If you haven't\MessageBreak
47 loaded fontspec manually, try\MessageBreak
48 loading mathfont earlier in\MessageBreak
49 your preamble.}}
50 \def\M@InvalidOptionError#1{\PackageError{mathfont}
51 {Invalid option "#1"\MessageBreak for \string\mathfont}
52 {Hm. Check that you spelled the\MessageBreak
53 option correctly. Otherwise, I'm\MessageBreak
54 not sure what's wrong. Is this\MessageBreak
55 option listed in the package\MessageBreak
56 documentation?}}
57 \def\M@MissingOptionError{\PackageError{mathfont}
58 {Missing option for \string\mathfont}
59 {It looks like you included a , or = in\MessageBreak
60 the optional argument of \string\mathfont\space\MessageBreak
61 but didn't put anything before it.}}
62 \def\M@MissingSuboptionError{\PackageError{mathfont}
63 {Missing suboption for \string\mathfont}
64 {It looks like you included an = somewhere\MessageBreak
65 but didn't put the suboption after it. Either\MessageBreak
66 that or you typed == instead of = in the\MessageBreak
67 optional argument of \string\mathfont.}}
68 \def\M@InternalsRestoredError{\PackageError{mathfont}
69 {Internal commands restored}
70 {This package slightly changes two LaTeX\MessageBreak
71 internal commands, and you really shouldn't\MessageBreak
72 be loading new math fonts without those\MessageBreak
73 adjustments. What happened here is that you\MessageBreak
74 used \string\mathfont\space in a situation
75 where those\MessageBreak
76 two commands retain their original defini-\MessageBreak
77 tions. Presumably you used
78 \string\mathfont\space after\MessageBreak
79 loading another package or calling the\MessageBreak
80 \string\restoremathinternals\space
81 command, and in any\MessageBreak
82 event, I'm going to ignore this call to\MessageBreak
83 \string\mathfont. Try retypesetting this document\MessageBreak
84 with all \string\mathfont\space
85 commands immediately\MessageBreak
86 after \string\usepackage{mathfont}.}}
87 \def\M@MissingControlSequenceError#1{\PackageError{mathfont}
88 {Improper argument\MessageBreak
89 for \string#1}
90 {Please use a control sequence as the\MessageBreak
91 first argument of \string#1.}}
92 \def\M@HModeError#1{\PackageError{mathfont}

```

```

93 {Missing \string$ inserted.\MessageBreak
94 Command \string#1 must be used in\MessageBreak
95 math mode}
96 {I generated an error because\MessageBreak
97 you used \string#1 outside of\MessageBreak
98 math mode. I've inserted a \string$\MessageBreak
99 just before your \string#1, so\MessageBreak
100 we should be all good now.}}

```

We want to create a salient error message if the engine doesn't define the required XeTeX or LuaTeX primitives. We change + to active to force TeX to print the required spaces in the message, and we put the entire production inside a group to make this change local. The @gobbletwo eats the extra period and return that L^ATeX adds to the error message.

```

101 \begingroup
102 \catcode'\+=\active
103 \def+{\space}
104 \def\M@XeTeXLuaTeXError{\GenericError{
105   {\MessageBreak\MessageBreak
106   Package mathfont fatal error:
107   \MessageBreak\MessageBreak
108   +*****\MessageBreak
109   +*****\MessageBreak
110   +!!!FATAL ERROR!!!\MessageBreak
111   +*****\MessageBreak
112   +***Missing XeTeX***\MessageBreak
113   +***or LuaTeX***\MessageBreak
114   +*****\MessageBreak
115   +*****\MessageBreak\@gobbletwo}
116   {See the mathfont package documentation for explanation.}
117   {This package requires either\MessageBreak
118   XeLaTeX or LuaLaTeX. Please\MessageBreak
119   retypeset your document with\MessageBreak
120   one of those two engines. This\MessageBreak
121   error is fatal, so I'm going\MessageBreak
122   to stop processing once you\MessageBreak
123   push enter.}\@end}

```

Check that the engine has defined the necessary primitives.

```

124 \ifx\Umathcode\@undefined
125 \else
126   \ifx\Umathchardef\@undefined
127   \else
128     \let\M@XeTeXLuaTeXError\relax
129   \fi
130 \fi
131 \M@XeTeXLuaTeXError
132 \endgroup

```

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.

```

133 \@ifpackageloaded{fontspec}
134   {\ifnum\csname g__fontspec_math_bool\endcsname=1
135     \M@NoMathError
136     \fi}{\RequirePackage[no-math]{fontspec}}
137 \expandafter\let\expandafter\@newfont
138   \csname fontspec_set_family:Nnn\endcsname

```

We save `\set@mathchar` and `\set@mathsymbol` from the \LaTeX 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 \XeTeX and \LuaTeX primitives `\Umathcode` and `\Umathchardef`. The unicode primitives support decimal input using a `+` sign, and we take advantage of that feature to avoid hexadecimal conversions.

```

139 \M@SetInternalsInfo
140 \let\@@set@mathchar\set@mathchar
141 \let\@@set@mathsymbol\set@mathsymbol
142 \def\set@mathchar#1#2#3#4{%
143   \multiply\count\z@ by 16\relax
144   \advance\count\z@\count\tw@
145   \global\Umathcode'#2=\mathchar@type#3+#1+\count\z@}
146 \def\set@mathsymbol#1#2#3#4{%
147   \multiply\count\z@ by 16\relax
148   \advance\count\z@\count\tw@
149   \global\Umathchardef#2\mathchar@type#3+#1+\count\z@}

```

We need to keep track of the number of times we have loaded a font, and the count `\M@count` fulfills this role. We will use the `\M@toks` object to record a message that displays when the user calls `\mathfont`.

```

150 \newcount\M@count
151 \M@count\z@
152 \newtoks\M@toks

```

We create necessary booleans and the default math font shapes.

```

153 \newif\if@upper
154 \newif\if@lower
155 \newif\if@greekupper
156 \newif\if@greeklower
157 \newif\if@operator
158 \newif\if@digits
159 \newif\if@symbols
160 \newif\if@bb
161 \newif\if@cal
162 \newif\if@frak
163 \newif\if@bcal
164 \newif\if@bfrak
165 \newif\if@suboptionpresent

```

```

166 \def\@uppershape{italic}
167 \def\@lowershape{italic}
168 \def\@greekuppershape{roman}
169 \def\@greeklowershape{italic}
170 \def\@digitsshape{roman}
171 \def\@symbolsshape{roman}
172 \def\@operatorshape{roman}
173 \def\@bbshape{roman}
174 \def\@calshape{roman}
175 \def\@frakshape{roman}
176 \def\@bcalshape{roman}
177 \def\@bfrakshape{roman}

```

The command `\check@option@valid` confirms that a user's keyword option is legitimate. The macro defines `\@temperror` to be an invalid option error and loops through all possible options. If the command argument matches one of the correct possibilities, the package changes `\@temperror` to `\relax`. The macro ends this process with a call to `\@temperror`, so the package issues an error if and only if the specified option is invalid. 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.

```

178 \def\check@option@valid#1{%
179   \def\@temperror{\M@InvalidOptionError{#1}}
180   \@for\@j:=upper,lower,greekupper,greeklower,digits,symbols,operator\do{%
181     \ifx\@j#1
182       \let\@temperror\relax
183     \fi}
184   \@for\@j:=bb,cal,frak,bcal,bfrak\do{%
185     \ifx\@j#1
186       \let\@temperror\relax
187       \csname define@\@j\endcsname
188     \fi}
189   \@temperror}

```

Do the same thing for the suboption.

```

190 \def\check@suboption@valid#1{%
191   \def\@temperror{\M@InvalidOptionError{#1}}
192   \@for\@j:=roman,italic\do{%
193     \ifx\@j#1
194       \let\@temperror\relax
195     \fi}
196   \@temperror}

```

We want to allow the user to specify options using an `xkeyval`-type syntax. However, we do not need the full package; a slim 24 lines of code will suffice. We use the next three macros to parse the optional argument of `\mathfont`. The control sequence

$$\@gobblesp \langle text \rangle \@nil \langle macro \rangle$$

takes the first string of consecutive non-space tokens in *text* and sets it to be the contents of

macro. We remove initial spaces with `\@ifnextchar` and use delimiters in the definition of `\g@bblesp@def` for the token selection and final storage.

```
197 \def\@gobble@equals#1=#2\@nil{#1}
198 \def\@gobblesp{\@ifnextchar\@empty\relax\g@bblesp@def}
199 \def\g@bblesp@def#1 #2\@nil#3{\def#3{#1}}
```

Parse the option and suboption. When `\mathfont` expands, it loops through its optional argument with `\@for`, and for each option-suboption token string, it calls `\parse@suboption` on the token string followed by `=\@nil`. Doing so splits the option and suboption, and `\parse@suboption` handles each separately. All tokens before the first `=` become the option, and all tokens after become the suboption. The macro removes any extra spaces with `\@gobblesp` and puts the option in `\@tempa`. We include the extra `\` in the argument of `\@gobblesp` because we need to check that the option portion of the token string contains non-space characters. If not, `\@tempa` will become `\`, and the package issues an error. (If yes, `\g@bblesp@def` will gobble the `\` and following space, so it won't matter.)

```
200 \def\parse@suboption#1=#2\@nil{%
201   \@gobblesp#1 \ \ \@nil\@tempa
202   \def\@tempb{\ \}
203   \ifx\@tempa\@tempb
204     \M@MissingOptionError
205   \else
206     \check@option@valid\@tempa
207   \fi
```

Adding the `=\@nil` to the token string allows us to see immediately whether the user specified a suboption by checking if `#2` is empty. If not, the macro again removes extra spaces with `\@gobblesp`, and we check whether `\@let@token` from the `\@ifnextchar` is the extra `=`. That would mean the user typed `=` but didn't follow it with a suboption, and in that case, `mathfont` issues an error. If the suboption is valid, the macro gobbles the last `=` with `\@gobble@equals` and stores the result in `\@tempb`.

```
208   \def\@tempb{#2}
209   \ifx\@tempb\@empty
210     \@suboptionpresentfalse
211   \else
212     \@gobblesp#2 \@nil\@tempb
213     \if\@let@token=
214       \M@MissingSuboptionError
215     \else
216       \edef\@tempb{\expandafter\@gobble@equals\@tempb=\@nil}
217       \check@suboption@valid\@tempb
218     \fi
219     \@suboptionpresenttrue
220   \fi}
```

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.

```
221 \def\mathfont{\@ifnextchar[{\@mathfont}
222   {\@mathfont [upper,lower,greekupper,greeklower,digits,symbols,operator]}}
223 \def\@mathfont [#1]#2{%
224   \ifx\@@set@mathchar\set@mathchar
225     \M@InternalsRestoredError
226   \else
227     \M@toks{}
228     \@tempswafalse
229     \expandafter\@newfont\csname M@font\the\M@count\endcsname\{#2}
230     \edef\@tempa{#1}
231     \@for\@i:=\@tempa\do{\expandafter\parse@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!)

```
232     \expandafter\ifx\csname if@\@tempa\expandafter\endcsname
233       \csname iftrue\endcsname
234       \M@CharsSetWarning{\@tempa}
235     \else
```

```

236     \if@tempswa
237     \expandafter\M@toks\expandafter{\the\M@toks, }
238     \else
239     \@tempswatrue
240     \fi
241     \edef\@tempc{\the\M@toks\@tempa}
242     \expandafter\M@toks\expandafter{\@tempc}

```

Handle the case with a suboption.

```

243     \if@suboptionpresent
244     \expandafter\edef\csname @\@tempa shape\endcsname{\@tempb}
245     \edef\@tempc{\the\M@toks\space(\@tempb)}
246     \expandafter\M@toks\expandafter{\@tempc}
247     \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.

```

248     \def\@tempb{roman}
249     \expandafter\ifx\csname @\@tempa shape\endcsname\@tempb
250     \@ifundefined{symMroman\the\M@count}{%
251     \DeclareSymbolFont{Mroman\the\M@count}{TU}
252     {\csname M@font\the\M@count\endcsname}{m}{n}}{}
253     \fi
254     \def\@tempb{italic}
255     \expandafter\ifx\csname @\@tempa shape\endcsname\@tempb
256     \@ifundefined{symMitalic\the\M@count}{%
257     \DeclareSymbolFont{Mitalic\the\M@count}{TU}
258     {\csname M@font\the\M@count\endcsname}{m}{it}}{}
259     \fi

```

And now the magic happens.

```

260     \M@FontChangeInfo{\@tempa}
261     \csname M@\@tempa @set\endcsname
262     \csname @\@tempa true\endcsname
263     \fi}
264     \advance\M@count\@ne\relax
265     \message{:: mathfont :: following keyword(s) changed to #2:^^J%
266     --\the\M@toks.^^J}
267     \fi}
268 \@onlypreamble\mathfont

```

We end the section by coding `\setfont`.

```

269 \def\setfont#1{\mathfont{#1}\setmainfont[Ligatures=TeX]{#1}}
270 \@onlypreamble\setfont

```

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.

```

271 \def\check@csarg#1#2#3{%
272   \ifcat\relax\noexpand#2
273     \def\@tempa{#2}
274     \@newfont\@tempb{}{#3}
275     \expandafter\@newmathfontcommand
276   \else
277     \M@MissingControlSequenceError#1
278   \fi}

```

Now declare the math alphabet.

```

279 \def\@newmathfontcommand#1#2{\expandafter
280   \DeclareMathAlphabet\expandafter{\@tempa}{TU}{\@tempb}{#1}{#2}}
281 \def\newmathfontcommand{\check@csarg\newmathfontcommand}
282 \@onlypreamble\newmathfontcommand

```

Then provide the four default versions.

```

283 \def\newmathrm#1#2{\check@csarg\newmathrm{#1}{#2}{m}{n}}
284 \def\newmathit#1#2{\check@csarg\newmathit{#1}{#2}{m}{it}}
285 \def\newmathbold#1#2{\check@csarg\newmathbold{#1}{#2}{b}{n}}
286 \def\newmathboldit#1#2{\check@csarg\newmathboldit{#1}{#2}{b}{it}}
287 \@onlypreamble\newmathrm
288 \@onlypreamble\newmathit
289 \@onlypreamble\newmathbold
290 \@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.

```

291 \def\check@mmode#1#2{%
292   \let\@tempa#2%
293   \ifmmode
294     \expandafter\@tempa
295   \else
296     \M@HModeError#1%
297     \leavevmode\expandafter$\expandafter\@tempa
298   \fi}
299 \def\process@tokens#1#2{\@tfor\@k:=#1\do{\csname M@#2@\@k\endcsname}}
300 \def\define@bb{%
301   \M@CommandInitializeInfo{blackboard bold}
302   \def\mathbb{\check@mmode\mathbb\@mathbb}
303   \def\@mathbb##1{\process@tokens{##1}{bb}}

```

```

304 \def\define@cal{%
305   \M@CommandInitializeInfo{calligraphic}
306   \def\mathcal{\check@mmode\mathcal\@mathcal}
307   \def\@mathcal##1{\process@tokens{##1}{cal}}}
308 \def\define@frak{%
309   \M@CommandInitializeInfo{fraktur}
310   \def\mathfrak{\check@mmode\mathfrak\@mathfrak}
311   \def\@mathfrak##1{\process@tokens{##1}{frak}}}
312 \def\define@bcal{%
313   \M@CommandInitializeInfo{bold calligraphic}
314   \def\mathbcal{\check@mmode\mathbcal\@mathbcal}
315   \def\@mathbcal##1{\process@tokens{##1}{bcal}}}
316 \def\define@bfrak{%
317   \M@CommandInitializeInfo{bold fraktur}
318   \def\mathbfrak{\check@mmode\mathbfrak\@mathbfrak}
319   \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^AT_EX twenty-six expansions later when it evaluates each `\DeclareMathSymbol`.

```

320 \def\M@upper@set{%
321   \edef\M@upper{M@uppershape\the\M@count}
322   \DeclareMathSymbol{A}{\mathalpha}{\M@upper}{‘A}
323   \DeclareMathSymbol{B}{\mathalpha}{\M@upper}{‘B}
324   \DeclareMathSymbol{C}{\mathalpha}{\M@upper}{‘C}
325   \DeclareMathSymbol{D}{\mathalpha}{\M@upper}{‘D}
326   \DeclareMathSymbol{E}{\mathalpha}{\M@upper}{‘E}
327   \DeclareMathSymbol{F}{\mathalpha}{\M@upper}{‘F}
328   \DeclareMathSymbol{G}{\mathalpha}{\M@upper}{‘G}
329   \DeclareMathSymbol{H}{\mathalpha}{\M@upper}{‘H}
330   \DeclareMathSymbol{I}{\mathalpha}{\M@upper}{‘I}
331   \DeclareMathSymbol{J}{\mathalpha}{\M@upper}{‘J}
332   \DeclareMathSymbol{K}{\mathalpha}{\M@upper}{‘K}
333   \DeclareMathSymbol{L}{\mathalpha}{\M@upper}{‘L}
334   \DeclareMathSymbol{M}{\mathalpha}{\M@upper}{‘M}
335   \DeclareMathSymbol{N}{\mathalpha}{\M@upper}{‘N}
336   \DeclareMathSymbol{O}{\mathalpha}{\M@upper}{‘O}
337   \DeclareMathSymbol{P}{\mathalpha}{\M@upper}{‘P}
338   \DeclareMathSymbol{Q}{\mathalpha}{\M@upper}{‘Q}
339   \DeclareMathSymbol{R}{\mathalpha}{\M@upper}{‘R}
340   \DeclareMathSymbol{S}{\mathalpha}{\M@upper}{‘S}
341   \DeclareMathSymbol{T}{\mathalpha}{\M@upper}{‘T}
342   \DeclareMathSymbol{U}{\mathalpha}{\M@upper}{‘U}
343   \DeclareMathSymbol{V}{\mathalpha}{\M@upper}{‘V}
344   \DeclareMathSymbol{W}{\mathalpha}{\M@upper}{‘W}

```

```

345 \DeclareMathSymbol{X}{\mathalpha}{\M@upper}{'X}
346 \DeclareMathSymbol{Y}{\mathalpha}{\M@upper}{'Y}
347 \DeclareMathSymbol{Z}{\mathalpha}{\M@upper}{'Z}

```

Set minuscule Latin characters.

```

348 \def\M@lower@set{%
349 \edef\M@lower{\M@lowershape\the\M@count}
350 \DeclareMathSymbol{a}{\mathalpha}{\M@lower}{'a}
351 \DeclareMathSymbol{b}{\mathalpha}{\M@lower}{'b}
352 \DeclareMathSymbol{c}{\mathalpha}{\M@lower}{'c}
353 \DeclareMathSymbol{d}{\mathalpha}{\M@lower}{'d}
354 \DeclareMathSymbol{e}{\mathalpha}{\M@lower}{'e}
355 \DeclareMathSymbol{f}{\mathalpha}{\M@lower}{'f}
356 \DeclareMathSymbol{g}{\mathalpha}{\M@lower}{'g}
357 \DeclareMathSymbol{h}{\mathalpha}{\M@lower}{'h}
358 \DeclareMathSymbol{i}{\mathalpha}{\M@lower}{'i}
359 \DeclareMathSymbol{j}{\mathalpha}{\M@lower}{'j}
360 \DeclareMathSymbol{k}{\mathalpha}{\M@lower}{'k}
361 \DeclareMathSymbol{l}{\mathalpha}{\M@lower}{'l}
362 \DeclareMathSymbol{m}{\mathalpha}{\M@lower}{'m}
363 \DeclareMathSymbol{n}{\mathalpha}{\M@lower}{'n}
364 \DeclareMathSymbol{o}{\mathalpha}{\M@lower}{'o}
365 \DeclareMathSymbol{p}{\mathalpha}{\M@lower}{'p}
366 \DeclareMathSymbol{q}{\mathalpha}{\M@lower}{'q}
367 \DeclareMathSymbol{r}{\mathalpha}{\M@lower}{'r}
368 \DeclareMathSymbol{s}{\mathalpha}{\M@lower}{'s}
369 \DeclareMathSymbol{t}{\mathalpha}{\M@lower}{'t}
370 \DeclareMathSymbol{u}{\mathalpha}{\M@lower}{'u}
371 \DeclareMathSymbol{v}{\mathalpha}{\M@lower}{'v}
372 \DeclareMathSymbol{w}{\mathalpha}{\M@lower}{'w}
373 \DeclareMathSymbol{x}{\mathalpha}{\M@lower}{'x}
374 \DeclareMathSymbol{y}{\mathalpha}{\M@lower}{'y}
375 \DeclareMathSymbol{z}{\mathalpha}{\M@lower}{'z}

```

Set capital Greek characters.

```

376 \def\M@greekupper@set{%
377 \edef\M@greekupper{\M@greekuppershape\the\M@count}
378 \DeclareMathSymbol{\Alpha}{\mathalpha}{\M@greekupper}{"391}
379 \DeclareMathSymbol{\Beta}{\mathalpha}{\M@greekupper}{"392}
380 \DeclareMathSymbol{\Gamma}{\mathalpha}{\M@greekupper}{"393}
381 \DeclareMathSymbol{\Delta}{\mathalpha}{\M@greekupper}{"394}
382 \DeclareMathSymbol{\Epsilon}{\mathalpha}{\M@greekupper}{"395}
383 \DeclareMathSymbol{\Zeta}{\mathalpha}{\M@greekupper}{"396}
384 \DeclareMathSymbol{\Eta}{\mathalpha}{\M@greekupper}{"397}
385 \DeclareMathSymbol{\Theta}{\mathalpha}{\M@greekupper}{"398}
386 \DeclareMathSymbol{\Iota}{\mathalpha}{\M@greekupper}{"399}
387 \DeclareMathSymbol{\Kappa}{\mathalpha}{\M@greekupper}{"39A}
388 \DeclareMathSymbol{\Lambda}{\mathalpha}{\M@greekupper}{"39B}
389 \DeclareMathSymbol{\Mu}{\mathalpha}{\M@greekupper}{"39C}

```

```

390 \DeclareMathSymbol{\Nu}{\mathalpha}{\M@greekupper}{"39D}
391 \DeclareMathSymbol{\Xi}{\mathalpha}{\M@greekupper}{"39E}
392 \DeclareMathSymbol{\Omicron}{\mathalpha}{\M@greekupper}{"39F}
393 \DeclareMathSymbol{\Pi}{\mathalpha}{\M@greekupper}{"3A0}
394 \DeclareMathSymbol{\Rho}{\mathalpha}{\M@greekupper}{"3A1}
395 \DeclareMathSymbol{\Sigma}{\mathalpha}{\M@greekupper}{"3A3}
396 \DeclareMathSymbol{\Tau}{\mathalpha}{\M@greekupper}{"3A4}
397 \DeclareMathSymbol{\Upsilon}{\mathalpha}{\M@greekupper}{"3A5}
398 \DeclareMathSymbol{\Phi}{\mathalpha}{\M@greekupper}{"3A6}
399 \DeclareMathSymbol{\Chi}{\mathalpha}{\M@greekupper}{"3A7}
400 \DeclareMathSymbol{\Psi}{\mathalpha}{\M@greekupper}{"3A8}
401 \DeclareMathSymbol{\Omega}{\mathalpha}{\M@greekupper}{"3A9}
402 \DeclareMathSymbol{\varTheta}{\mathalpha}{\M@greekupper}{"3F4}
403 \DeclareMathSymbol{\Heta}{\mathalpha}{\M@greekupper}{"370}
404 \DeclareMathSymbol{\Sampi}{\mathalpha}{\M@greekupper}{"3E0}
405 \DeclareMathSymbol{\Diagamma}{\mathalpha}{\M@greekupper}{"3DC}
406 \DeclareMathSymbol{\Koppa}{\mathalpha}{\M@greekupper}{"3D8}
407 \DeclareMathSymbol{\Stigma}{\mathalpha}{\M@greekupper}{"3DA}
408 \DeclareMathSymbol{\Sho}{\mathalpha}{\M@greekupper}{"3F7}
409 \DeclareMathSymbol{\San}{\mathalpha}{\M@greekupper}{"3FA}
410 \DeclareMathSymbol{\varSampi}{\mathalpha}{\M@greekupper}{"372}
411 \DeclareMathSymbol{\varDiagamma}{\mathalpha}{\M@greekupper}{"376}
412 \DeclareMathSymbol{\varKoppa}{\mathalpha}{\M@greekupper}{"3DE}

```

Set minuscule Greek characters.

```

413 \def\M@greeklower@set{%
414 \edef\M@greeklower{\M@greeklowershape\the\M@count}
415 \DeclareMathSymbol{\alpha}{\mathalpha}{\M@greeklower}{"3B1}
416 \DeclareMathSymbol{\beta}{\mathalpha}{\M@greeklower}{"3B2}
417 \DeclareMathSymbol{\gamma}{\mathalpha}{\M@greeklower}{"3B3}
418 \DeclareMathSymbol{\delta}{\mathalpha}{\M@greeklower}{"3B4}
419 \DeclareMathSymbol{\epsilon}{\mathalpha}{\M@greeklower}{"3F5}
420 \DeclareMathSymbol{\zeta}{\mathalpha}{\M@greeklower}{"3B6}
421 \DeclareMathSymbol{\eta}{\mathalpha}{\M@greeklower}{"3B7}
422 \DeclareMathSymbol{\theta}{\mathalpha}{\M@greeklower}{"3B8}
423 \DeclareMathSymbol{\iota}{\mathalpha}{\M@greeklower}{"3B9}
424 \DeclareMathSymbol{\kappa}{\mathalpha}{\M@greeklower}{"3BA}
425 \DeclareMathSymbol{\lambda}{\mathalpha}{\M@greeklower}{"3BB}
426 \DeclareMathSymbol{\mu}{\mathalpha}{\M@greeklower}{"3BC}
427 \DeclareMathSymbol{\nu}{\mathalpha}{\M@greeklower}{"3BD}
428 \DeclareMathSymbol{\xi}{\mathalpha}{\M@greeklower}{"3BE}
429 \DeclareMathSymbol{\omicron}{\mathalpha}{\M@greeklower}{"3BF}
430 \DeclareMathSymbol{\pi}{\mathalpha}{\M@greeklower}{"3C0}
431 \DeclareMathSymbol{\rho}{\mathalpha}{\M@greeklower}{"3C1}
432 \DeclareMathSymbol{\sigma}{\mathalpha}{\M@greeklower}{"3C3}
433 \DeclareMathSymbol{\tau}{\mathalpha}{\M@greeklower}{"3C4}
434 \DeclareMathSymbol{\upsilon}{\mathalpha}{\M@greeklower}{"3C5}
435 \DeclareMathSymbol{\phi}{\mathalpha}{\M@greeklower}{"3D5}

```

```

436 \DeclareMathSymbol{\chi}{\mathalpha}{\M@greeklower}{"3C7}
437 \DeclareMathSymbol{\psi}{\mathalpha}{\M@greeklower}{"3C8}
438 \DeclareMathSymbol{\omega}{\mathalpha}{\M@greeklower}{"3C9}
439 \DeclareMathSymbol{\varbeta}{\mathalpha}{\M@greeklower}{"3D0}
440 \DeclareMathSymbol{\varepsilon}{\mathalpha}{\M@greeklower}{"3B5}
441 \DeclareMathSymbol{\vartheta}{\mathalpha}{\M@greeklower}{"3D1}
442 \DeclareMathSymbol{\varrho}{\mathalpha}{\M@greeklower}{"3F1}
443 \DeclareMathSymbol{\varsigma}{\mathalpha}{\M@greeklower}{"3C2}
444 \DeclareMathSymbol{\varphi}{\mathalpha}{\M@greeklower}{"3C6}
445 \DeclareMathSymbol{\heta}{\mathalpha}{\M@greeklower}{"371}
446 \DeclareMathSymbol{\sampi}{\mathalpha}{\M@greeklower}{"3E1}
447 \DeclareMathSymbol{\diagamma}{\mathalpha}{\M@greeklower}{"3DD}
448 \DeclareMathSymbol{\koppa}{\mathalpha}{\M@greeklower}{"3D9}
449 \DeclareMathSymbol{\stigma}{\mathalpha}{\M@greeklower}{"3DB}
450 \DeclareMathSymbol{\sho}{\mathalpha}{\M@greeklower}{"3F8}
451 \DeclareMathSymbol{\san}{\mathalpha}{\M@greeklower}{"3FB}
452 \DeclareMathSymbol{\varsampi}{\mathalpha}{\M@greeklower}{"373}
453 \DeclareMathSymbol{\vardiagamma}{\mathalpha}{\M@greeklower}{"377}
454 \DeclareMathSymbol{\varkoppa}{\mathalpha}{\M@greeklower}{"3DF}

```

Set digits.

```

455 \def\M@digits@set{%
456 \edef\M@digits{M@digitsshape\the\M@count}
457 \DeclareMathSymbol{0}{\mathalpha}{\M@digits}{'0}
458 \DeclareMathSymbol{1}{\mathalpha}{\M@digits}{'1}
459 \DeclareMathSymbol{2}{\mathalpha}{\M@digits}{'2}
460 \DeclareMathSymbol{3}{\mathalpha}{\M@digits}{'3}
461 \DeclareMathSymbol{4}{\mathalpha}{\M@digits}{'4}
462 \DeclareMathSymbol{5}{\mathalpha}{\M@digits}{'5}
463 \DeclareMathSymbol{6}{\mathalpha}{\M@digits}{'6}
464 \DeclareMathSymbol{7}{\mathalpha}{\M@digits}{'7}
465 \DeclareMathSymbol{8}{\mathalpha}{\M@digits}{'8}
466 \DeclareMathSymbol{9}{\mathalpha}{\M@digits}{'9}}

```

Set symbols.

```

467 \def\M@symbols@set{%
468 \edef\M@symbols{M@symbolsshape\the\M@count}
469 \DeclareMathSymbol{.}{\mathord}{\M@symbols}{"2E}
470 \DeclareMathSymbol{@}{\mathord}{\M@symbols}{"40}
471 \DeclareMathSymbol{\mathhash}{\mathord}{\M@symbols}{"23}
472 \DeclareMathSymbol{\mathdollar}{\mathord}{\M@symbols}{"24}
473 \DeclareMathSymbol{\mathpercent}{\mathord}{\M@symbols}{"25}
474 \DeclareMathSymbol{\mathand}{\mathord}{\M@symbols}{"26}
475 \DeclareMathSymbol{\mathparagraph}{\mathord}{\M@symbols}{"B6}
476 \DeclareMathSymbol{\mathsection}{\mathord}{\M@symbols}{"A7}
477 \DeclareMathSymbol{\mathsterling}{\mathord}{\M@symbols}{"A3}
478 \DeclareMathSymbol{|}{\mathord}{\M@symbols}{"7C}
479 \DeclareMathSymbol{\neg}{\mathord}{\M@symbols}{"AC}
480 \DeclareMathSymbol{\infty}{\mathord}{\M@symbols}{"221E}

```



```

481 \DeclareMathSymbol{\partial}{\mathord}{\M@symbols}{"2202}
482 \DeclareMathSymbol{\mathbackslash}{\mathord}{\M@symbols}{"5C}
483 \DeclareMathSymbol{\degree}{\mathord}{\M@symbols}{"B0}
484 \DeclareMathSymbol{\Delta}{\mathord}{\M@symbols}{"2206}
485 \DeclareMathSymbol{'}{\mathord}{\M@symbols}{"2032}
486 \DeclareMathSymbol{"}{\mathord}{\M@symbols}{"2033}
487 \DeclareMathSymbol{\comma}{\mathord}{\M@symbols}{"2C}
488 \if@operators
489   \DeclareMathSymbol{\sum}{\mathop}{\M@symbols}{"2211}
490   \DeclareMathSymbol{\prod}{\mathop}{\M@symbols}{"220F}
491   \DeclareMathSymbol{\intop}{\mathop}{\M@symbols}{"222B}
492 \fi
493 \DeclareMathSymbol{+}{\mathbin}{\M@symbols}{"2B}
494 \DeclareMathSymbol{-}{\mathbin}{\M@symbols}{"2212}
495 \DeclareMathSymbol{*}{\mathbin}{\M@symbols}{"2A}
496 \DeclareMathSymbol{\times}{\mathbin}{\M@symbols}{"D7}
497 \DeclareMathSymbol{/}{\mathbin}{\M@symbols}{"2215}
498 \DeclareMathSymbol{\div}{\mathbin}{\M@symbols}{"F7}
499 \DeclareMathSymbol{\pm}{\mathbin}{\M@symbols}{"B1}
500 \DeclareMathSymbol{\cdot}{\mathbin}{\M@symbols}{"2219}
501 \DeclareMathSymbol{\bullet}{\mathbin}{\M@symbols}{"2022}
502 \DeclareMathSymbol{\dagger}{\mathbin}{\M@symbols}{"2020}
503 \DeclareMathSymbol{\ddagger}{\mathbin}{\M@symbols}{"2021}
504 \DeclareMathSymbol{\setminus}{\mathbin}{\M@symbols}{"5C}
505 \DeclareMathSymbol{=}{\mathrel}{\M@symbols}{"3D}
506 \DeclareMathSymbol{<}{\mathrel}{\M@symbols}{"3C}
507 \DeclareMathSymbol{>}{\mathrel}{\M@symbols}{"3E}
508 \DeclareMathSymbol{\leq}{\mathrel}{\M@symbols}{"2264}
509 \DeclareMathSymbol{\geq}{\mathrel}{\M@symbols}{"2265}
510 \DeclareMathSymbol{\sim}{\mathrel}{\M@symbols}{"7E}
511 \DeclareMathSymbol{\approx}{\mathrel}{\M@symbols}{"2248}
512 \DeclareMathSymbol{\equiv}{\mathrel}{\M@symbols}{"2261}
513 \DeclareMathSymbol{\mid}{\mathrel}{\M@symbols}{"7C}
514 \DeclareMathSymbol{\parallel}{\mathrel}{\M@symbols}{"2016}
515 \DeclareMathSymbol{:}{\mathrel}{\M@symbols}{"3A}
516 \DeclareMathSymbol{(){\mathopen}{\M@symbols}{"28}
517 \DeclareMathSymbol{)}{\mathclose}{\M@symbols}{"29}
518 \DeclareMathSymbol{[]{\mathopen}{\M@symbols}{"5B}
519 \DeclareMathSymbol{[]}{\mathclose}{\M@symbols}{"5D}
520 \DeclareMathSymbol{\leftbrace}{\mathopen}{\M@symbols}{"7B}
521 \DeclareMathSymbol{\rightbrace}{\mathclose}{\M@symbols}{"7D}
522 \DeclareMathSymbol{?}{\mathclose}{\M@symbols}{"3F}
523 \DeclareMathSymbol{!}{\mathclose}{\M@symbols}{"21}
524 \DeclareMathSymbol{,}{\mathpunct}{\M@symbols}{"2C}
525 \DeclareMathSymbol{;}{\mathpunct}{\M@symbols}{"3B}
526 \DeclareMathSymbol{\colon}{\mathpunct}{\M@symbols}{"3A}
527 \let\mathellipsis\undefined

```

528 `\DeclareMathSymbol{\mathellipsis}{\mathinner}{\M@symbols}{"2026}`

Finally a bit of housekeeping. We redefine `\#`, `\$`, `\%`, and `\&` as robust commands that expand to previously declared `\mathhash`, etc. commands 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. The last three commands defined here preserve the Computer Modern font for characters used in several math-mode symbols.

```
529 \DeclareRobustCommand\#{\ifmmode\mathhash\else\char"23\fi}
530 \DeclareRobustCommand\${\ifmmode\mathdollar\else\char"24\fi}
531 \DeclareRobustCommand\%\{\ifmmode\mathpercent\else\char"25\fi}
532 \DeclareRobustCommand\&\{\ifmmode\mathand\else\char"26\fi}
533 \DeclareMathSymbol{\@relbar}{\mathbin}{symbols}{"00}
534 \DeclareMathSymbol{\@Relbar}{\mathrel}{operators}{"3D}
535 \DeclareMathSymbol{\@verticalbar}{\mathord}{symbols}{"6A}
536 \DeclareRobustCommand\relbar{\mathrel{\smash\@relbar}}
537 \DeclareRobustCommand\Relbar{\mathrel{\@Relbar}}
538 \DeclareRobustCommand\models{\mathrel{\@verticalbar}\joinrel\Relbar}}
```

Set new operator font.

```
539 \def\M@operator@set{%
540 \edef\operator@font{\noexpand\mathgroup
541 \expandafter\noexpand\cname symM\@operatorshape\the\M@count\endcname}}
```

Set blackboard bold characters.

```
542 \def\M@bb@set{%
543 \edef\M@bb{M\@bbshape\the\M@count}
544 \DeclareMathSymbol{\M@bb@A}{\mathord}{\M@bb}{"1D538}
545 \DeclareMathSymbol{\M@bb@B}{\mathord}{\M@bb}{"1D539}
546 \DeclareMathSymbol{\M@bb@C}{\mathord}{\M@bb}{"2102}
547 \DeclareMathSymbol{\M@bb@D}{\mathord}{\M@bb}{"1D53B}
548 \DeclareMathSymbol{\M@bb@E}{\mathord}{\M@bb}{"1D53C}
549 \DeclareMathSymbol{\M@bb@F}{\mathord}{\M@bb}{"1D53D}
550 \DeclareMathSymbol{\M@bb@G}{\mathord}{\M@bb}{"1D53E}
551 \DeclareMathSymbol{\M@bb@H}{\mathord}{\M@bb}{"210D}
552 \DeclareMathSymbol{\M@bb@I}{\mathord}{\M@bb}{"1D540}
553 \DeclareMathSymbol{\M@bb@J}{\mathord}{\M@bb}{"1D541}
554 \DeclareMathSymbol{\M@bb@K}{\mathord}{\M@bb}{"1D542}
555 \DeclareMathSymbol{\M@bb@L}{\mathord}{\M@bb}{"1D543}
556 \DeclareMathSymbol{\M@bb@M}{\mathord}{\M@bb}{"1D544}
557 \DeclareMathSymbol{\M@bb@N}{\mathord}{\M@bb}{"2115}
558 \DeclareMathSymbol{\M@bb@O}{\mathord}{\M@bb}{"1D546}
559 \DeclareMathSymbol{\M@bb@P}{\mathord}{\M@bb}{"2119}
560 \DeclareMathSymbol{\M@bb@Q}{\mathord}{\M@bb}{"211A}
561 \DeclareMathSymbol{\M@bb@R}{\mathord}{\M@bb}{"211D}
562 \DeclareMathSymbol{\M@bb@S}{\mathord}{\M@bb}{"1D54A}
563 \DeclareMathSymbol{\M@bb@T}{\mathord}{\M@bb}{"1D54B}
564 \DeclareMathSymbol{\M@bb@U}{\mathord}{\M@bb}{"1D54C}
565 \DeclareMathSymbol{\M@bb@V}{\mathord}{\M@bb}{"1D54D}}
```

```

566 \DeclareMathSymbol{\M@bb@W}{\mathord}{\M@bb}{"1D54E}
567 \DeclareMathSymbol{\M@bb@X}{\mathord}{\M@bb}{"1D43F}
568 \DeclareMathSymbol{\M@bb@Y}{\mathord}{\M@bb}{"1D550}
569 \DeclareMathSymbol{\M@bb@Z}{\mathord}{\M@bb}{"2124}
570 \DeclareMathSymbol{\M@bb@a}{\mathord}{\M@bb}{"1D552}
571 \DeclareMathSymbol{\M@bb@b}{\mathord}{\M@bb}{"1D553}
572 \DeclareMathSymbol{\M@bb@c}{\mathord}{\M@bb}{"1D554}
573 \DeclareMathSymbol{\M@bb@d}{\mathord}{\M@bb}{"1D555}
574 \DeclareMathSymbol{\M@bb@e}{\mathord}{\M@bb}{"1D556}
575 \DeclareMathSymbol{\M@bb@f}{\mathord}{\M@bb}{"1D557}
576 \DeclareMathSymbol{\M@bb@g}{\mathord}{\M@bb}{"1D558}
577 \DeclareMathSymbol{\M@bb@h}{\mathord}{\M@bb}{"1D559}
578 \DeclareMathSymbol{\M@bb@i}{\mathord}{\M@bb}{"1D55A}
579 \DeclareMathSymbol{\M@bb@j}{\mathord}{\M@bb}{"1D55B}
580 \DeclareMathSymbol{\M@bb@k}{\mathord}{\M@bb}{"1D55C}
581 \DeclareMathSymbol{\M@bb@l}{\mathord}{\M@bb}{"1D55D}
582 \DeclareMathSymbol{\M@bb@m}{\mathord}{\M@bb}{"1D55E}
583 \DeclareMathSymbol{\M@bb@n}{\mathord}{\M@bb}{"1D55F}
584 \DeclareMathSymbol{\M@bb@o}{\mathord}{\M@bb}{"1D560}
585 \DeclareMathSymbol{\M@bb@p}{\mathord}{\M@bb}{"1D561}
586 \DeclareMathSymbol{\M@bb@q}{\mathord}{\M@bb}{"1D562}
587 \DeclareMathSymbol{\M@bb@r}{\mathord}{\M@bb}{"1D563}
588 \DeclareMathSymbol{\M@bb@s}{\mathord}{\M@bb}{"1D564}
589 \DeclareMathSymbol{\M@bb@t}{\mathord}{\M@bb}{"1D565}
590 \DeclareMathSymbol{\M@bb@u}{\mathord}{\M@bb}{"1D566}
591 \DeclareMathSymbol{\M@bb@v}{\mathord}{\M@bb}{"1D567}
592 \DeclareMathSymbol{\M@bb@w}{\mathord}{\M@bb}{"1D568}
593 \DeclareMathSymbol{\M@bb@x}{\mathord}{\M@bb}{"1D569}
594 \DeclareMathSymbol{\M@bb@y}{\mathord}{\M@bb}{"1D56A}
595 \DeclareMathSymbol{\M@bb@z}{\mathord}{\M@bb}{"1D56B}
596 \expandafter\DeclareMathSymbol\expandafter
597   {\csname M@bb@0\endcsname}{\mathord}{\M@bb}{"1D7D8}
598 \expandafter\DeclareMathSymbol\expandafter
599   {\csname M@bb@1\endcsname}{\mathord}{\M@bb}{"1D7D9}
600 \expandafter\DeclareMathSymbol\expandafter
601   {\csname M@bb@2\endcsname}{\mathord}{\M@bb}{"1D7DA}
602 \expandafter\DeclareMathSymbol\expandafter
603   {\csname M@bb@3\endcsname}{\mathord}{\M@bb}{"1D7DB}
604 \expandafter\DeclareMathSymbol\expandafter
605   {\csname M@bb@4\endcsname}{\mathord}{\M@bb}{"1D7DC}
606 \expandafter\DeclareMathSymbol\expandafter
607   {\csname M@bb@5\endcsname}{\mathord}{\M@bb}{"1D7DD}
608 \expandafter\DeclareMathSymbol\expandafter
609   {\csname M@bb@6\endcsname}{\mathord}{\M@bb}{"1D7DE}
610 \expandafter\DeclareMathSymbol\expandafter
611   {\csname M@bb@7\endcsname}{\mathord}{\M@bb}{"1D7DF}
612 \expandafter\DeclareMathSymbol\expandafter

```

```

613   {\csname M@bb@8\endcsname}{\mathord}{\M@bb}{"1D7E0}
614 \expandafter\DeclareMathSymbol\expandafter
615   {\csname M@bb@9\endcsname}{\mathord}{\M@bb}{"1D7E1}}

```

Set caligraphic characters.

```

616 \def\M@cal@set{%
617   \edef\M@cal{M@calshape\the\M@count}
618   \DeclareMathSymbol{\M@cal@A}{\mathord}{\M@cal}{"1D49C}
619   \DeclareMathSymbol{\M@cal@B}{\mathord}{\M@cal}{"212C}
620   \DeclareMathSymbol{\M@cal@C}{\mathord}{\M@cal}{"1D49E}
621   \DeclareMathSymbol{\M@cal@D}{\mathord}{\M@cal}{"1D49F}
622   \DeclareMathSymbol{\M@cal@E}{\mathord}{\M@cal}{"2130}
623   \DeclareMathSymbol{\M@cal@F}{\mathord}{\M@cal}{"2131}
624   \DeclareMathSymbol{\M@cal@G}{\mathord}{\M@cal}{"1D4A2}
625   \DeclareMathSymbol{\M@cal@H}{\mathord}{\M@cal}{"210B}
626   \DeclareMathSymbol{\M@cal@I}{\mathord}{\M@cal}{"2110}
627   \DeclareMathSymbol{\M@cal@J}{\mathord}{\M@cal}{"1D4A5}
628   \DeclareMathSymbol{\M@cal@K}{\mathord}{\M@cal}{"1D4A6}
629   \DeclareMathSymbol{\M@cal@L}{\mathord}{\M@cal}{"2112}
630   \DeclareMathSymbol{\M@cal@M}{\mathord}{\M@cal}{"2133}
631   \DeclareMathSymbol{\M@cal@N}{\mathord}{\M@cal}{"1D4A9}
632   \DeclareMathSymbol{\M@cal@O}{\mathord}{\M@cal}{"1D4AA}
633   \DeclareMathSymbol{\M@cal@P}{\mathord}{\M@cal}{"1D4AB}
634   \DeclareMathSymbol{\M@cal@Q}{\mathord}{\M@cal}{"1D4AC}
635   \DeclareMathSymbol{\M@cal@R}{\mathord}{\M@cal}{"211B}
636   \DeclareMathSymbol{\M@cal@S}{\mathord}{\M@cal}{"1D4AE}
637   \DeclareMathSymbol{\M@cal@T}{\mathord}{\M@cal}{"1D4AF}
638   \DeclareMathSymbol{\M@cal@U}{\mathord}{\M@cal}{"1D4B0}
639   \DeclareMathSymbol{\M@cal@V}{\mathord}{\M@cal}{"1D4B1}
640   \DeclareMathSymbol{\M@cal@W}{\mathord}{\M@cal}{"1D4B2}
641   \DeclareMathSymbol{\M@cal@X}{\mathord}{\M@cal}{"1D4B3}
642   \DeclareMathSymbol{\M@cal@Y}{\mathord}{\M@cal}{"1D4B4}
643   \DeclareMathSymbol{\M@cal@Z}{\mathord}{\M@cal}{"1D4B5}
644   \DeclareMathSymbol{\M@cal@a}{\mathord}{\M@cal}{"1D4B6}
645   \DeclareMathSymbol{\M@cal@b}{\mathord}{\M@cal}{"1D4B7}
646   \DeclareMathSymbol{\M@cal@c}{\mathord}{\M@cal}{"1D4B8}
647   \DeclareMathSymbol{\M@cal@d}{\mathord}{\M@cal}{"1D4B9}
648   \DeclareMathSymbol{\M@cal@e}{\mathord}{\M@cal}{"212F}
649   \DeclareMathSymbol{\M@cal@f}{\mathord}{\M@cal}{"1D4BB}
650   \DeclareMathSymbol{\M@cal@g}{\mathord}{\M@cal}{"210A}
651   \DeclareMathSymbol{\M@cal@h}{\mathord}{\M@cal}{"1D4BD}
652   \DeclareMathSymbol{\M@cal@i}{\mathord}{\M@cal}{"1D4BE}
653   \DeclareMathSymbol{\M@cal@j}{\mathord}{\M@cal}{"1D4BF}
654   \DeclareMathSymbol{\M@cal@k}{\mathord}{\M@cal}{"1D4C0}
655   \DeclareMathSymbol{\M@cal@l}{\mathord}{\M@cal}{"1D4C1}
656   \DeclareMathSymbol{\M@cal@m}{\mathord}{\M@cal}{"1D4C2}
657   \DeclareMathSymbol{\M@cal@n}{\mathord}{\M@cal}{"1D4C3}
658   \DeclareMathSymbol{\M@cal@o}{\mathord}{\M@cal}{"2134}

```

```

659 \DeclareMathSymbol{\M@cal@p}{\mathord}{\M@cal}{"1D4C5}
660 \DeclareMathSymbol{\M@cal@q}{\mathord}{\M@cal}{"1D4C6}
661 \DeclareMathSymbol{\M@cal@r}{\mathord}{\M@cal}{"1D4C7}
662 \DeclareMathSymbol{\M@cal@s}{\mathord}{\M@cal}{"1D4C8}
663 \DeclareMathSymbol{\M@cal@t}{\mathord}{\M@cal}{"1D4C9}
664 \DeclareMathSymbol{\M@cal@u}{\mathord}{\M@cal}{"1D4CA}
665 \DeclareMathSymbol{\M@cal@v}{\mathord}{\M@cal}{"1D4CB}
666 \DeclareMathSymbol{\M@cal@w}{\mathord}{\M@cal}{"1D4CC}
667 \DeclareMathSymbol{\M@cal@x}{\mathord}{\M@cal}{"1D4CD}
668 \DeclareMathSymbol{\M@cal@y}{\mathord}{\M@cal}{"1D4CE}
669 \DeclareMathSymbol{\M@cal@z}{\mathord}{\M@cal}{"1D4CF}}

```

Set fraktur characters.

```

670 \def\M@frak@set{%
671 \edef\M@frak{M\@frakshape\the\M@count}
672 \DeclareMathSymbol{\M@frak@A}{\mathord}{\M@frak}{"1D504}
673 \DeclareMathSymbol{\M@frak@B}{\mathord}{\M@frak}{"1D505}
674 \DeclareMathSymbol{\M@frak@C}{\mathord}{\M@frak}{"212D}
675 \DeclareMathSymbol{\M@frak@D}{\mathord}{\M@frak}{"1D507}
676 \DeclareMathSymbol{\M@frak@E}{\mathord}{\M@frak}{"1D508}
677 \DeclareMathSymbol{\M@frak@F}{\mathord}{\M@frak}{"1D509}
678 \DeclareMathSymbol{\M@frak@G}{\mathord}{\M@frak}{"1D50A}
679 \DeclareMathSymbol{\M@frak@H}{\mathord}{\M@frak}{"210C}
680 \DeclareMathSymbol{\M@frak@I}{\mathord}{\M@frak}{"2111}
681 \DeclareMathSymbol{\M@frak@J}{\mathord}{\M@frak}{"1D50D}
682 \DeclareMathSymbol{\M@frak@K}{\mathord}{\M@frak}{"1D50E}
683 \DeclareMathSymbol{\M@frak@L}{\mathord}{\M@frak}{"1D50F}
684 \DeclareMathSymbol{\M@frak@M}{\mathord}{\M@frak}{"1D510}
685 \DeclareMathSymbol{\M@frak@N}{\mathord}{\M@frak}{"1D511}
686 \DeclareMathSymbol{\M@frak@O}{\mathord}{\M@frak}{"1D512}
687 \DeclareMathSymbol{\M@frak@P}{\mathord}{\M@frak}{"1D513}
688 \DeclareMathSymbol{\M@frak@Q}{\mathord}{\M@frak}{"1D514}
689 \DeclareMathSymbol{\M@frak@R}{\mathord}{\M@frak}{"212C}
690 \DeclareMathSymbol{\M@frak@S}{\mathord}{\M@frak}{"1D516}
691 \DeclareMathSymbol{\M@frak@T}{\mathord}{\M@frak}{"1D517}
692 \DeclareMathSymbol{\M@frak@U}{\mathord}{\M@frak}{"1D518}
693 \DeclareMathSymbol{\M@frak@V}{\mathord}{\M@frak}{"1D519}
694 \DeclareMathSymbol{\M@frak@W}{\mathord}{\M@frak}{"1D51A}
695 \DeclareMathSymbol{\M@frak@X}{\mathord}{\M@frak}{"1D51B}
696 \DeclareMathSymbol{\M@frak@Y}{\mathord}{\M@frak}{"1D51C}
697 \DeclareMathSymbol{\M@frak@Z}{\mathord}{\M@frak}{"2128}
698 \DeclareMathSymbol{\M@frak@a}{\mathord}{\M@frak}{"1D51E}
699 \DeclareMathSymbol{\M@frak@b}{\mathord}{\M@frak}{"1D51F}
700 \DeclareMathSymbol{\M@frak@c}{\mathord}{\M@frak}{"1D520}
701 \DeclareMathSymbol{\M@frak@d}{\mathord}{\M@frak}{"1D521}
702 \DeclareMathSymbol{\M@frak@e}{\mathord}{\M@frak}{"1D522}
703 \DeclareMathSymbol{\M@frak@f}{\mathord}{\M@frak}{"1D523}
704 \DeclareMathSymbol{\M@frak@g}{\mathord}{\M@frak}{"1D524}

```

```

705 \DeclareMathSymbol{\M@frak@h}{\mathord}{\M@frak}{"1D525}
706 \DeclareMathSymbol{\M@frak@i}{\mathord}{\M@frak}{"1D526}
707 \DeclareMathSymbol{\M@frak@j}{\mathord}{\M@frak}{"1D527}
708 \DeclareMathSymbol{\M@frak@k}{\mathord}{\M@frak}{"1D528}
709 \DeclareMathSymbol{\M@frak@l}{\mathord}{\M@frak}{"1D529}
710 \DeclareMathSymbol{\M@frak@m}{\mathord}{\M@frak}{"1D52A}
711 \DeclareMathSymbol{\M@frak@n}{\mathord}{\M@frak}{"1D52B}
712 \DeclareMathSymbol{\M@frak@o}{\mathord}{\M@frak}{"1D52C}
713 \DeclareMathSymbol{\M@frak@p}{\mathord}{\M@frak}{"1D52D}
714 \DeclareMathSymbol{\M@frak@q}{\mathord}{\M@frak}{"1D52E}
715 \DeclareMathSymbol{\M@frak@r}{\mathord}{\M@frak}{"1D52F}
716 \DeclareMathSymbol{\M@frak@s}{\mathord}{\M@frak}{"1D530}
717 \DeclareMathSymbol{\M@frak@t}{\mathord}{\M@frak}{"1D531}
718 \DeclareMathSymbol{\M@frak@u}{\mathord}{\M@frak}{"1D532}
719 \DeclareMathSymbol{\M@frak@v}{\mathord}{\M@frak}{"1D533}
720 \DeclareMathSymbol{\M@frak@w}{\mathord}{\M@frak}{"1D534}
721 \DeclareMathSymbol{\M@frak@x}{\mathord}{\M@frak}{"1D535}
722 \DeclareMathSymbol{\M@frak@y}{\mathord}{\M@frak}{"1D536}
723 \DeclareMathSymbol{\M@frak@z}{\mathord}{\M@frak}{"1D537}}

```

Set bold caligraphic characters.

```

724 \def\M@bcal@set{%
725   \edef\M@bcal{\M@bcalshape\the\M@count}
726   \DeclareMathSymbol{\M@bcal@A}{\mathord}{\M@bcal}{"1D4D0}
727   \DeclareMathSymbol{\M@bcal@B}{\mathord}{\M@bcal}{"1D4D1}
728   \DeclareMathSymbol{\M@bcal@C}{\mathord}{\M@bcal}{"1D4D2}
729   \DeclareMathSymbol{\M@bcal@D}{\mathord}{\M@bcal}{"1D4D3}
730   \DeclareMathSymbol{\M@bcal@E}{\mathord}{\M@bcal}{"1D4D4}
731   \DeclareMathSymbol{\M@bcal@F}{\mathord}{\M@bcal}{"1D4D5}
732   \DeclareMathSymbol{\M@bcal@G}{\mathord}{\M@bcal}{"1D4D6}
733   \DeclareMathSymbol{\M@bcal@H}{\mathord}{\M@bcal}{"1D4D7}
734   \DeclareMathSymbol{\M@bcal@I}{\mathord}{\M@bcal}{"1D4D8}
735   \DeclareMathSymbol{\M@bcal@J}{\mathord}{\M@bcal}{"1D4D9}
736   \DeclareMathSymbol{\M@bcal@K}{\mathord}{\M@bcal}{"1D4DA}
737   \DeclareMathSymbol{\M@bcal@L}{\mathord}{\M@bcal}{"1D4DB}
738   \DeclareMathSymbol{\M@bcal@M}{\mathord}{\M@bcal}{"1D4DC}
739   \DeclareMathSymbol{\M@bcal@N}{\mathord}{\M@bcal}{"1D4DD}
740   \DeclareMathSymbol{\M@bcal@O}{\mathord}{\M@bcal}{"1D4DE}
741   \DeclareMathSymbol{\M@bcal@P}{\mathord}{\M@bcal}{"1D4DF}
742   \DeclareMathSymbol{\M@bcal@Q}{\mathord}{\M@bcal}{"1D4E0}
743   \DeclareMathSymbol{\M@bcal@R}{\mathord}{\M@bcal}{"1D4E1}
744   \DeclareMathSymbol{\M@bcal@S}{\mathord}{\M@bcal}{"1D4E2}
745   \DeclareMathSymbol{\M@bcal@T}{\mathord}{\M@bcal}{"1D4E3}
746   \DeclareMathSymbol{\M@bcal@U}{\mathord}{\M@bcal}{"1D4E4}
747   \DeclareMathSymbol{\M@bcal@V}{\mathord}{\M@bcal}{"1D4E5}
748   \DeclareMathSymbol{\M@bcal@W}{\mathord}{\M@bcal}{"1D4E6}
749   \DeclareMathSymbol{\M@bcal@X}{\mathord}{\M@bcal}{"1D4E7}
750   \DeclareMathSymbol{\M@bcal@Y}{\mathord}{\M@bcal}{"1D4E8}

```

```

751 \DeclareMathSymbol{\M@bcal@Z}{\mathord}{\M@bcal}{"1D4E9}
752 \DeclareMathSymbol{\M@bcal@a}{\mathord}{\M@bcal}{"1D4EA}
753 \DeclareMathSymbol{\M@bcal@b}{\mathord}{\M@bcal}{"1D4EB}
754 \DeclareMathSymbol{\M@bcal@c}{\mathord}{\M@bcal}{"1D4EC}
755 \DeclareMathSymbol{\M@bcal@d}{\mathord}{\M@bcal}{"1D4ED}
756 \DeclareMathSymbol{\M@bcal@e}{\mathord}{\M@bcal}{"1D4EE}
757 \DeclareMathSymbol{\M@bcal@f}{\mathord}{\M@bcal}{"1D4EF}
758 \DeclareMathSymbol{\M@bcal@g}{\mathord}{\M@bcal}{"1D4F0}
759 \DeclareMathSymbol{\M@bcal@h}{\mathord}{\M@bcal}{"1D4F1}
760 \DeclareMathSymbol{\M@bcal@i}{\mathord}{\M@bcal}{"1D4F2}
761 \DeclareMathSymbol{\M@bcal@j}{\mathord}{\M@bcal}{"1D4F3}
762 \DeclareMathSymbol{\M@bcal@k}{\mathord}{\M@bcal}{"1D4F4}
763 \DeclareMathSymbol{\M@bcal@l}{\mathord}{\M@bcal}{"1D4F5}
764 \DeclareMathSymbol{\M@bcal@m}{\mathord}{\M@bcal}{"1D4F6}
765 \DeclareMathSymbol{\M@bcal@n}{\mathord}{\M@bcal}{"1D4F7}
766 \DeclareMathSymbol{\M@bcal@o}{\mathord}{\M@bcal}{"1D4F8}
767 \DeclareMathSymbol{\M@bcal@p}{\mathord}{\M@bcal}{"1D4F9}
768 \DeclareMathSymbol{\M@bcal@q}{\mathord}{\M@bcal}{"1D4FA}
769 \DeclareMathSymbol{\M@bcal@r}{\mathord}{\M@bcal}{"1D4FB}
770 \DeclareMathSymbol{\M@bcal@s}{\mathord}{\M@bcal}{"1D4FC}
771 \DeclareMathSymbol{\M@bcal@t}{\mathord}{\M@bcal}{"1D4FD}
772 \DeclareMathSymbol{\M@bcal@u}{\mathord}{\M@bcal}{"1D4FE}
773 \DeclareMathSymbol{\M@bcal@v}{\mathord}{\M@bcal}{"1D4FF}
774 \DeclareMathSymbol{\M@bcal@w}{\mathord}{\M@bcal}{"1D500}
775 \DeclareMathSymbol{\M@bcal@x}{\mathord}{\M@bcal}{"1D501}
776 \DeclareMathSymbol{\M@bcal@y}{\mathord}{\M@bcal}{"1D502}
777 \DeclareMathSymbol{\M@bcal@z}{\mathord}{\M@bcal}{"1D503}

```

Set bold fraktur characters.

```

778 \def\M@bfrac@set{%
779 \edef\M@bfrac{M\M@bfracshape\the\M@count}
780 \DeclareMathSymbol{\M@bfrac@A}{\mathord}{\M@bfrac}{"1D56C}
781 \DeclareMathSymbol{\M@bfrac@B}{\mathord}{\M@bfrac}{"1D56D}
782 \DeclareMathSymbol{\M@bfrac@C}{\mathord}{\M@bfrac}{"1D56E}
783 \DeclareMathSymbol{\M@bfrac@D}{\mathord}{\M@bfrac}{"1D56F}
784 \DeclareMathSymbol{\M@bfrac@E}{\mathord}{\M@bfrac}{"1D570}
785 \DeclareMathSymbol{\M@bfrac@F}{\mathord}{\M@bfrac}{"1D571}
786 \DeclareMathSymbol{\M@bfrac@G}{\mathord}{\M@bfrac}{"1D572}
787 \DeclareMathSymbol{\M@bfrac@H}{\mathord}{\M@bfrac}{"1D573}
788 \DeclareMathSymbol{\M@bfrac@I}{\mathord}{\M@bfrac}{"1D574}
789 \DeclareMathSymbol{\M@bfrac@J}{\mathord}{\M@bfrac}{"1D575}
790 \DeclareMathSymbol{\M@bfrac@K}{\mathord}{\M@bfrac}{"1D576}
791 \DeclareMathSymbol{\M@bfrac@L}{\mathord}{\M@bfrac}{"1D577}
792 \DeclareMathSymbol{\M@bfrac@M}{\mathord}{\M@bfrac}{"1D578}
793 \DeclareMathSymbol{\M@bfrac@N}{\mathord}{\M@bfrac}{"1D579}
794 \DeclareMathSymbol{\M@bfrac@O}{\mathord}{\M@bfrac}{"1D57A}
795 \DeclareMathSymbol{\M@bfrac@P}{\mathord}{\M@bfrac}{"1D57B}
796 \DeclareMathSymbol{\M@bfrac@Q}{\mathord}{\M@bfrac}{"1D57C}

```

```

797 \DeclareMathSymbol{\M@bfrac@R}{\mathord}{\M@bfrac}{"1D57D}
798 \DeclareMathSymbol{\M@bfrac@S}{\mathord}{\M@bfrac}{"1D57E}
799 \DeclareMathSymbol{\M@bfrac@T}{\mathord}{\M@bfrac}{"1D57F}
800 \DeclareMathSymbol{\M@bfrac@U}{\mathord}{\M@bfrac}{"1D580}
801 \DeclareMathSymbol{\M@bfrac@V}{\mathord}{\M@bfrac}{"1D581}
802 \DeclareMathSymbol{\M@bfrac@W}{\mathord}{\M@bfrac}{"1D582}
803 \DeclareMathSymbol{\M@bfrac@X}{\mathord}{\M@bfrac}{"1D583}
804 \DeclareMathSymbol{\M@bfrac@Y}{\mathord}{\M@bfrac}{"1D584}
805 \DeclareMathSymbol{\M@bfrac@Z}{\mathord}{\M@bfrac}{"1D585}
806 \DeclareMathSymbol{\M@bfrac@a}{\mathord}{\M@bfrac}{"1D586}
807 \DeclareMathSymbol{\M@bfrac@b}{\mathord}{\M@bfrac}{"1D587}
808 \DeclareMathSymbol{\M@bfrac@c}{\mathord}{\M@bfrac}{"1D588}
809 \DeclareMathSymbol{\M@bfrac@d}{\mathord}{\M@bfrac}{"1D589}
810 \DeclareMathSymbol{\M@bfrac@e}{\mathord}{\M@bfrac}{"1D58A}
811 \DeclareMathSymbol{\M@bfrac@f}{\mathord}{\M@bfrac}{"1D58B}
812 \DeclareMathSymbol{\M@bfrac@g}{\mathord}{\M@bfrac}{"1D58C}
813 \DeclareMathSymbol{\M@bfrac@h}{\mathord}{\M@bfrac}{"1D58D}
814 \DeclareMathSymbol{\M@bfrac@i}{\mathord}{\M@bfrac}{"1D58E}
815 \DeclareMathSymbol{\M@bfrac@j}{\mathord}{\M@bfrac}{"1D58F}
816 \DeclareMathSymbol{\M@bfrac@k}{\mathord}{\M@bfrac}{"1D590}
817 \DeclareMathSymbol{\M@bfrac@l}{\mathord}{\M@bfrac}{"1D591}
818 \DeclareMathSymbol{\M@bfrac@m}{\mathord}{\M@bfrac}{"1D592}
819 \DeclareMathSymbol{\M@bfrac@n}{\mathord}{\M@bfrac}{"1D593}
820 \DeclareMathSymbol{\M@bfrac@o}{\mathord}{\M@bfrac}{"1D594}
821 \DeclareMathSymbol{\M@bfrac@p}{\mathord}{\M@bfrac}{"1D595}
822 \DeclareMathSymbol{\M@bfrac@q}{\mathord}{\M@bfrac}{"1D596}
823 \DeclareMathSymbol{\M@bfrac@r}{\mathord}{\M@bfrac}{"1D597}
824 \DeclareMathSymbol{\M@bfrac@s}{\mathord}{\M@bfrac}{"1D598}
825 \DeclareMathSymbol{\M@bfrac@t}{\mathord}{\M@bfrac}{"1D599}
826 \DeclareMathSymbol{\M@bfrac@u}{\mathord}{\M@bfrac}{"1D59A}
827 \DeclareMathSymbol{\M@bfrac@v}{\mathord}{\M@bfrac}{"1D59B}
828 \DeclareMathSymbol{\M@bfrac@w}{\mathord}{\M@bfrac}{"1D59C}
829 \DeclareMathSymbol{\M@bfrac@x}{\mathord}{\M@bfrac}{"1D59D}
830 \DeclareMathSymbol{\M@bfrac@y}{\mathord}{\M@bfrac}{"1D59E}
831 \DeclareMathSymbol{\M@bfrac@z}{\mathord}{\M@bfrac}{"1D59F}}

```

Provide the command to reset macros and conclude.

```

832 \def\restoremathinternals{%
833   \M@RestoreInternalsInfo
834   \let\set@mathchar\@@set@mathchar
835   \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.

```

836 \if@packages
837 \else
838   \let\@@usepackage\usepackage
839   \def\usepackage{\restoremathinternals
840     \let\usepackage\@@usepackage

```



```
841 \usepackage}
842 \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	
<code>\#</code>	529
<code>\\$</code>	530
<code>\%</code>	531
<code>\&</code>	532
<code>\@@end</code>	123
<code>\@@set@mathchar</code>	140, 224, 834
<code>\@@set@mathsymbol</code>	141, 835
<code>\@@usepackage</code>	838, 840
<code>\@Relbar</code>	534, 537
<code>\@bbshape</code>	173, 543
<code>\@bcalshape</code>	176, 725
<code>\@bfrakshape</code>	177, 779
<code>\@calshape</code>	174, 617
<code>\@digitssshape</code>	170, 456
<code>\@for</code>	180, 184, 192, 231
<code>\@frakshape</code>	175, 671
<code>\@gobble@equals</code>	197, 216
<code>\@gobblesp</code>	198, 201, 212
<code>\@greeklowershape</code>	169, 414
<code>\@greekuppershape</code>	168, 377
<code>\@ifnextchar</code>	198, 221
<code>\@ifpackageloaded</code>	133
<code>\@ifundefined</code>	250, 256
<code>\@let@token</code>	213
<code>\@lowershape</code>	167, 349
<code>\@mathbb</code>	302, 303
<code>\@mathbcal</code>	314, 315
<code>\@mathbfrak</code>	318, 319
<code>\@mathcal</code>	306, 307
<code>\@mathfont</code>	221–223
<code>\@mathfrak</code>	310, 311
<code>\@newfont</code>	137, 229, 274
<code>\@newmathfontcommand</code>	275, 279
<code>\@onlypreamble</code>	268, 270, 282, 287–290
<code>\@operatorsfalse</code>	7
<code>\@operatorshape</code>	172, 541
<code>\@operatorstrue</code>	6
<code>\@packagestrue</code>	5
<code>\@relbar</code>	533, 536
<code>\@suboptionpresentfalse</code>	210
<code>\@suboptionpresenttrue</code>	219
<code>\@symbolsshape</code>	171, 468
<code>\@temperror</code>	179, 182, 186, 189, 191, 194, 196
<code>\@tempswafalse</code>	228
<code>\@tempswatrue</code>	239
<code>\@tfor</code>	299
<code>\@uppershape</code>	166, 321
<code>\@verticalbar</code>	535, 538
A	
<code>\Alpha</code>	378
<code>\alpha</code>	415
<code>\approx</code>	511
B	
<code>\Beta</code>	379
<code>\beta</code>	416
<code>\bullet</code>	501
C	
<code>\catcode</code>	102
<code>\cdot</code>	500
<code>\char</code>	529–532
<code>\check@csarg</code>	271, 281, 283–286
<code>\check@mmode</code>	291, 302, 306, 310, 314, 318
<code>\check@option@valid</code>	178, 206
<code>\check@suboption@valid</code>	190, 217
<code>\Chi</code>	399
<code>\chi</code>	436
<code>\colon</code>	526
<code>\comma</code>	487
<code>\CurrentOption</code>	9
D	
<code>\dagger</code>	502
<code>\ddagger</code>	503
<code>\DeclareMathAlphabet</code>	280
<code>\DeclareSymbolFont</code>	251, 257
<code>\define@bb</code>	300
<code>\define@bcal</code>	312
<code>\define@bfrak</code>	316
<code>\define@cal</code>	304
<code>\define@frak</code>	308

<code>\degree</code>	483	<code>\Koppa</code>	406
<code>\Delta</code>	381, 484	<code>\koppa</code>	448
<code>\delta</code>	418		
<code>\Diagamma</code>	405		L
<code>\diagamma</code>	447	<code>\Lambda</code>	388
<code>\div</code>	498	<code>\lambda</code>	425
		<code>\leftbrace</code>	520
		<code>\leq</code>	508
			M
E			
<code>\Epsilon</code>	382	<code>\M@bb@set</code>	542
<code>\epsilon</code>	419	<code>\M@bcal@set</code>	724
<code>\equiv</code>	512	<code>\M@bfrac@set</code>	778
<code>\Eta</code>	384	<code>\M@cal@set</code>	616
<code>\eta</code>	421	<code>\M@CharsSetWarning</code>	32, 234
		<code>\M@CommandInitializeInfo</code>	
			23, 301, 305, 309, 313, 317
G		<code>\M@digits@set</code>	455
<code>\g@bblesp@def</code>	198, 199	<code>\M@FontChangeInfo</code>	13, 260
<code>\Gamma</code>	380	<code>\M@frac@set</code>	670
<code>\gamma</code>	417	<code>\M@greeklower@set</code>	413
<code>\geq</code>	509	<code>\M@greekupper@set</code>	376
		<code>\M@HModeError</code>	92, 296
		<code>\M@InternalsRestoredError</code>	68, 225
H		<code>\M@InvalidOptionError</code>	50, 179, 191
<code>\Heta</code>	403	<code>\M@lower@set</code>	348
<code>\heta</code>	445	<code>\M@MissingControlSequenceError</code>	87, 277
		<code>\M@MissingOptionError</code>	57, 204
		<code>\M@MissingSuboptionError</code>	62, 214
		<code>\M@NewFontCommandInfo</code>	18
I		<code>\M@NoMathError</code>	35, 135
<code>\if@bb</code>	160	<code>\M@operator@set</code>	539
<code>\if@bcal</code>	163	<code>\M@RestoreInternalsInfo</code>	29, 833
<code>\if@bfrac</code>	164	<code>\M@SetInternalsInfo</code>	26, 139
<code>\if@cal</code>	161	<code>\M@symbols@set</code>	467
<code>\if@digits</code>	158	<code>\M@toks</code>	152, 227, 237, 241, 242, 245, 246, 266
<code>\if@frac</code>	162	<code>\M@upper@set</code>	320
<code>\if@greeklower</code>	156	<code>\M@XeTeXLuaTeXError</code>	104, 128, 131
<code>\if@greekupper</code>	155	<code>\mathand</code>	474, 532
<code>\if@lower</code>	154	<code>\mathbackslash</code>	482
<code>\if@operator</code>	157	<code>\mathbb</code>	302
<code>\if@operators</code>	4, 488	<code>\mathbcal</code>	314
<code>\if@packages</code>	3, 836	<code>\mathbfrac</code>	318
<code>\if@suboptionpresent</code>	165, 243	<code>\mathcal</code>	306
<code>\if@symbols</code>	159	<code>\mathchar@type</code>	145, 149
<code>\if@tempswa</code>	236	<code>\mathdollar</code>	472, 530
<code>\if@upper</code>	153	<code>\mathellipsis</code>	527, 528
<code>\infty</code>	480		
<code>\intop</code>	491		
<code>\Iota</code>	386		
<code>\iota</code>	423		
		K	
<code>\Kappa</code>	387		
<code>\kappa</code>	424		

\varsigma	443	\xi	428
\vartheta	402		
\vartheta	441	Z	
	X	\Zeta	383
\Xi	391	\zeta	420