The \textbf{mathastext} package

\textbf{Jean-François Burnol}
\texttt{jfbu (at) free (dot) fr}
Package version: 1.3zb (2023/12/29)

The \texttt{mathastext} package changes the fonts which are used in math mode for letters, digits and a few other punctuation and symbol signs to replace them with the font as used for the document text. Thus, the package makes it possible to use a quite arbitrary font without worrying too much that it does not have specially designed accompanying math fonts. Also, \texttt{mathastext} provides a simple mechanism in order to use more than one math-as-text font in the same document.

`\texttt{mathastext}' is a LaTeX package
\begin{verbatim}
\usepackage{mathastext}
\end{verbatim}

The document will use in math mode the text font as configured at package loading time, for these characters:

\begin{verbatim}
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
!?,.;+-=() []$/#%&<>|{}
\end{verbatim}

The command \texttt{\MTsetmathskips} allows to set up extra spacings around each given letter.

Use multiple \texttt{\MTastext[name]}'s to define in the preamble various math versions using each a given text font, to be later activated in the document body via the command \texttt{\MTversion[name]}.

With the subdued option, mathastext will be active only inside such math versions distinct from the normal and bold.

Main options: italic, frenchmath, defaultmathsizes, subdued, asterisk, LGRgreek.

Documentation generated from the source file with Time-stamp: <29-12-2023 at 10:54:52 CET>. 
1 What \texttt{mathastext} does

If you have used the package before please make sure to check first section 4 where all changes across releases are recorded.

All blue colored words, such as \texttt{Mathastext} or \texttt{italic}, are hyperlinked to their official descriptions located either in the section 2 (Package commands) or subsection 3.2 (Complete list of options).

1.1 Aim of this package and basic usage

The initial ideology of \texttt{mathastext} was to produce mathematical texts with a very uniform look, not separating math from text as strongly as is usually done.

\texttt{mathastext}'s basic aim is thus to have the same font for text and mathematics. With hundreds of free text fonts packaged for \LaTeX{} and only a handful of math ones, chances are your favorite text font does not mix so well with the available math ones; \texttt{mathastext} may then help. Note that \texttt{mathastext} was initially developed for the traditional \TeX{} fonts and engines, and that compatibility with Unicode engines and OpenType fonts is partial.
Here is a minimal example of what may go into the preamble:

\usepackage[T1]{fontenc}
\usepackage{times}
\usepackage[italic]{mathastext}

The package records which font is set up for text, at the time it is loaded, and then arranges things in order for this text font to be used in math mode as well. So, with the preamble as above all letters, digits, and punctuation signs inside math mode will then be typeset in Times.\footnote{\textit{Hanford et al.}, 1999} The exact list of characters concerned by \texttt{mathastext} is a subset of the basic ASCII set:

\begin{center}
\begin{tabular}{cccc}
\texttt{abedefghijklmnopqrstuvwxyz} & \texttt{ABCDEFGHIJKLMNOPQRSTUVWXYZ} & \texttt{0123456789} & \texttt{! ? \ast , . : ; + – = ( ) [ \] / # $ \% & < > | \{} \}
\end{tabular}
\end{center}

As one can see, this is a very limited list! Some possibilities exist regarding Greek letters and will be described later.

1.2 Miscellanea

Please note that most of this section was written many years ago (except for the item about math alphabets which was enriched with important background information and moved upfront at 1.3zb). But it should still be valid!

\textbf{Math alphabets:} Let us first recall fundamental facts of life, in the world of traditional PDF\TeX engine and \TeX fonts, as background for understanding what \texttt{mathastext} does in this context, which will be explained next. People familiar with using Unicode engines and \texttt{unicode-math}, please be aware that the semantics there of the LaTeX math alphabet commands are \textbf{significantly} modified!

- In the default \LaTeX set-up all five of \texttt{\textbackslash mathrm}, \texttt{\textbackslash mathbf}, \texttt{\textbackslash mathit}, \texttt{\textbackslash mathsf} and \texttt{\textbackslash mathtt} tell \TeX to use for their arguments specific OT1-encoded fonts,
- if the document body uses, as will be the case probably with any language other than English or its variants, some other encoding such as T1 for its fonts, there is no change whatsoever to the math configuration, indeed most font packages ignore it completely,
• it is thus a priori wrong to think of these commands as switching to some body text font, although letters within their scopes will act as in a text font, and in particular obey ligatures (this also applies to operator names defined by `amsmath`'s `\DeclareMathOperator` which are, with some extras not mentioned here, as using `\mathrm`),

• these commands are completely different in spirit from the \LaTeX\ `\text`, `\textbf`, and others, which change only some font axis; indeed the math alphabet commands inherit from legacy Knuth's `\rm`, `\bf`, and others which are complete font specifiers,

• in particular when nesting, it is the inner-most which wins,

• only mathematical characters (such as letters) which are declared to \TeX as being of “variable family type” react to being in the argument of a math alphabet command,

• lowercase Greek letters are by default in \LaTeX immune to math alphabets (so `\mathrm{\pi}` induces no change in output), but the eleven uppercase Greek letters are of “variable family type” because they are picked in the OT1-encoded font also used for operator names (the one to which `\mathrm` maps), and they occupy the exact same slots in the OML-encoding to which `\mathnormal` maps! So in default \TeX, `\mathnormal{\Gamma}` gives a slanted one. The slots occupied in OML-encoding by the lowercase Greek letters (to which encoding they are a priori assigned) give completely unrelated glyphs in the OT1-encoding, so it makes sense that the default \LaTeX declares them to not react to math alphabets. Notice though, that if \LaTeX had declared a `\mathnormalbold`, mapping to a bold OML-encoded font, it would have made sense to also have `\alpha`, `\beta`, etc... be of “variable family type”,

• but of course then one should a priori never use `\mathbf{\pi}` for example, because the default `\mathbf` selects an OT1-encoded font, where there is no pi glyph whatsoever and in particular not at the slot of pi in the OML encoding (slot number 25)!

Please keep all the above in mind when trying to understand what `mathastext` does with math alphabets. The most significant point described next naturally is that `mathastext` will sync `\mathnormal`, `\mathrm`, `\mathbf`, `\mathit`, `\mathsf` and `\mathtt` to map to the `mathastext`-ified body text fonts.

• `\mathnormal`, `\mathrm`, `\mathbf`, `\mathit`, `\mathsf` and `\mathtt` are modified to use the `mathastext`-ified text fonts; this can be disabled via `defaultalphabets` and related individual options, but the package always provides `\Mathnormal`, `\Mathrm`, etc..., to match the `mathastext` font configuration (prior to 1.3za `defaultalphabets` and related options also caused the `mathastext` alphabet commands not to be defined).
Recall that there may arise a “too many math alphabets” error if too many of these commands are used in the document: declaring them is not by itself the cause of the error. See the \LaTeX{} news entry of its 2021-11-15 release for the counter localmathalphabets (with default value 2) which can be now be used if one hits such a difficulty.

- We define a new math alphabet command \texttt{\textbackslash mathnormalbold} which gives direct access to the bold version of the \texttt{\textbackslash mathnormal} alphabet (rather than using either the \texttt{\textbackslash bm} command from the \texttt{bm} package or the \texttt{\textbackslash boldsymbol} command from the \texttt{amsbsy} package). As it does not exist in the default \LaTeX{} math font set-up, this alphabet is not subjected to the subdued option action.

- version 1.2 of \texttt{mathastext} has extended the scope of the math alphabets to apply to non-alphabetical characters and to operator names. This respects the automatic white spaces added by \TeX{} around math symbols.

- The extra skips around letters (see subsection 1.8 and subsection 1.9) are removed in the scope of the math alphabets.

- Depending on options, further math alphabet commands are defined by the package: \texttt{\textbackslash MathEulerBold}, \texttt{\textbackslash MathEuler}, \texttt{\textbackslash MathPSymbol}, and since 1.3x under the \texttt{LGRgreek} family of options also \texttt{\textbackslash mathgreekup} and \texttt{\textbackslash mathgreekit}. See subsubsection 1.7.3.

- Addition of \texttt{\textbackslash mathgreekupbold} and \texttt{\textbackslash mathgreekitbold} under the \texttt{LGRgreek} family of options.

- With the \texttt{LGRgreek+} option which enhances Greek letters with a specific behavior when in the arguments of the \texttt{\textbackslash mathrm, \textbackslash mathbf, etc...}, math alphabet commands, this special behavior is not triggered by the \texttt{\textbackslash Mathrm, \textbackslash Mathbf, et al.}, which are genuine unmodified math alphabet commands. See subsubsection 1.7.5.

\textbf{the en-dash as minus sign}: very often the - character from the text font does not give a good minus sign. So by default, the package uses the en-dash sign –. Use \texttt{noendash} to deactivate it. Starting with version 1.12 of the package this ‘en-dash as minus’ should work in all encodings, including Unicode (if \texttt{fontspec} has been loaded); see also \texttt{unicodeminus} for OpenType fonts.

\textbf{amsmath}: the behaviour of the \texttt{\textbackslash DeclareMathOperator} command of \texttt{amsmath} is modified by \texttt{mathastext} for it to use the correct font. Additionally, release 1.3n of \texttt{mathastext} at long last also handles an extra operation done by \texttt{amsmath} for ".:/-\star" to be used in operator names without the extra math spacing.\footnote{To the experts: there is a long story here that \texttt{\textbackslash newmcodes@} hardcodes the font, that it was not compatible with Unicode engines, that during some time (2013-2016) lualatex-math fixed that and very recently amsopn.sty 2016/03/08 v2.02 also, so now lualatex-math 1.6 does nothing} This customization is suppressed in \texttt{subdued} mode for the \texttt{normal} and \texttt{bold} math versions.
**hbar**: the default \TeX definition of \hbar would in our context make use of the h of the current math font (so for us, it is also the text font, perhaps in italic shape), but with a bar across the h from the original default math font for letters (usually cmmi). We redefine \hbar to use the text font macron accent (\=) as a mock math accent (this takes into account the italic option and is compatible with subscripts and superscripts).

Since 1.12 `mathastext` when dealing with a Unicode font sets the \hbar to be the character from the font having hexadecimal codepoint U+0127.

Since 1.3u the general 8bits font encoding is supported (see discussion of the `mathaccents` option at end of this list for the shared limitations). Brief testing with various usual \TeX fonts shows that the vertical positioning of the bar isn’t satisfying. It is planned to either add a parameter to adjust it or to modify altogether the mode of construction of the \hbar.

Use `nohbar` to tell `mathastext` not do provide its own \hbar.

**dotless i and j**: by default the package redefines \imath and \jmath to give (in math mode) the dotless i and j (if it exists at all) from the text font.\(^3\)

**asterisk**: versions of `mathastext` earlier than 1.2d [2013/01/02] did not do anything with the `\ast` control sequence but did pick the asterisk * in the document text font, and this often was a rather silly thing as the text asterisk is generally in a raised position. Furthermore, the * lost its status of a binary operator and was treated as an ‘ordinary’ symbol. An option `noasterisk` turned this feature off. Starting with 1.2d, the `noasterisk` option is deprecated and the new default is to do nothing. But when option `asterisk` is received by the package, then both `\ast` and * are simultaneously modified to use (as binary operators) the text asterisk, slightly lowered. The amount of lowering\(^4\) is decided by the mandatory argument to the command \MTlowerast\{\texttt{dimen}\}. The package initially does `\MTlowerast{.3\height}`. Doing `\MTlowerast{.5ex}` is not a good idea as it does not scale properly in the script and scriptscript styles. With an argument given as a multiple of \height, the asterisk will behave as expected in subscripts and superscripts of subscripts. But * is now ‘mathematically active’\(^5\) and \$R^*\$ or \$R^\ast\$ must be input as \$R^{\ast}\$ and \$R^{\ast}\$. Furthermore, they will obey the math alphabet commands.

\footnote{as it is already fixed "upstream" in \texttt{amsopn.sty}, but anyhow in both cases, this still hardcoded the font, so finally \texttt{mathastext} does the right thing from its point of view. See the code comments for more, there is an issue here with \texttt{Lua\TeX} not applying the curly right quote contrarily to \texttt{Xe\TeX}.}

\footnote{Since 1.12 it also redefined `i` and `j` for usability both in text and math modes, but this has been dropped at 1.3t. Breaking change!}

\footnote{with the option `symbolm1asc`, the asterisk is picked from the Symbol font, and the amount of lowering is non-customizable; however if a math alphabet command is used, the asterisk is then again from a text font and the lowering will be as specified by `\MTlowerast`.}

\footnote{in a hopefully safe way, for example \$\label{eq*1}\$ is ok.}
**Xe\TeX** and Lua\TeX: regarding the en-dash and the dotless i and j, the package is now under the Unicode engines compatible not only with the “Unicode” \LaTeX font encodings EU1 (Xe\TeX, old fontspec), EU2 (Lua\TeX, old fontspec), TU (Xe\TeX and Lua\TeX, modern fontspec), but also with traditional 8bits-encodings declared as a fontenc option.

**fontspec:** one more note to users of Xe\TeX/Lua\TeX with fontspec: it has to be loaded with the option no-math, and before mathastext.

**vec accent:** The default \texttt{\vec} accent is not appropriate for upright letters, so \texttt{mathastext} provides a \texttt{\fouriervec} which takes its glyph in a Fourier font, and an Ersatz \texttt{\pmvec} which is reasonably good looking on upright letters and works with the \texttt{\rightarrow} glyph. Contrarily to version 1.0, the default \texttt{\vec} is not overwritten with \texttt{\fouriervec}. And contrarily to version 1.1, one now needs to pass the option \texttt{fouriervec} to have the math accent \texttt{\fouriervec} defined by the package.\footnote{this costs a math family, as I never came back to this to try to do otherwise.}

**math accents:** if option \texttt{mathaccents} is used then \texttt{mathastext} attempts to let the math accents \texttt{\acute}, \texttt{\grave}, etc... use the suitable glyphs from the text font. Prior to 1.3u only OT1, T1, and LY1 were supported (via hardcoded slots). It should now work with any 8bits font encoding having been declared as an option to the fontenc package (and of course providing the ten needed text accents which will mock math accents).\footnote{The code will raise low-level \TeX errors if the user attempts to use an 8bits font encoding whose \LaTeX definition file is lacking the suitable uses of \texttt{\DeclareTextAccent} or if the low level \LaTeX macro implementation of text accents changes significantly; in such cases please report the problem to the author, so that it can be documented in future releases!}\footnote{i.e., the \texttt{\grave} etc... control sequences will, in math versions with an OpenType \texttt{mathastext}-ified font, expand to macros holding their initial meanings, unmodified by \texttt{mathastext}, which was in force at the \texttt{\begin{document}}.}

The \texttt{\vec} math accent is not handled here, as it is not available in the usual 8bits font encodings. See the \texttt{fouriervec} option or the \texttt{\pmvec} command.

The math accents obey the subdued option and will change in sync with the \texttt{mathastext}-ified text font used in each non subdued math version.

(Very) brief testing during 1.3u development with Xe\TeX and Lua\TeX let the author conclude that usage with the \texttt{\Umathaccent} primitive of an OpenType accent glyph slot (which in the text font is for usage as a postpended combining character) gives definitely bad horizontal placements for both engines (each in its own way). Thus, the redefinitions of accents for a \texttt{mathastext} declared math version with an OpenType font is by default canceled.\footnote{I.e., the \texttt{\grave} etc... control sequences will, in math versions with an OpenType \texttt{mathastext}-ified font, expand to macros holding their initial meanings, unmodified by \texttt{mathastext}, which was in force at the \texttt{\begin{document}}.} Use \texttt{unimathaccents} to force usage of the OpenType font text accents glyph slots with the \texttt{\Umathaccent} primitive. Expert users are invited to check out the code and to contribute suggestions if some extras can improve it.
varying font encodings: the very first release of mathastext dealt with only one font; very soon thereafter it acquired the capacity to define multiple math versions, each one using its own text font. But, as was documented at this location formerly, various encoding dependent decisions were done once and for all during package loading.

This meant in particular that the minus sign (using the text endash), the dotless i and j, the \hbar, the math accents were all set up for only one unique font encoding. It was thus recommended that all math versions share the same font encoding.

The 1.3u release has lifted this restriction. (1.3u)

1.3 Examples
Here is another simple example:
\usepackage{libertinus-type1}
\usepackage[italic,LGRgreek,defaultmathsizes]{mathastext}

The LGRgreek option is there to take advantage that the libertinus-type1 package also provides Greek letters in LGR encoding, which can thus be used by mathastext in math mode. And we do here as if we did not know about the existence of the libertinust1math package! This would have been the obvious choice, but then one wouldn’t need mathastext and I couldn’t even start this documentation.

More sophisticated preambles will use multiple times the \Mathastext command in the preamble with its optional argument \(\langle\text{math\_version}\rangle\) in order to define math versions corresponding to a given font configuration. These mathastext-enriched math versions are then activated in the document body via the \MTversion\(\langle\text{math\_version}\rangle\) command, which modifies both the text font and the math font.

We now give some examples with a verbatim copy of the preamble code corresponding to them, as can be found in the source of this documentation. The detailed option and command descriptions will be given later.

First of all, the package was loaded using this:
\usepackage[subdued,\%
asterrisk,\%
defaultmathsizes,\%
symbolmisc,symbolre,\%
LGRgreek]{mathastext}

---

10Michael Sharpe, A Type 1 font and \LaTeX support for Libertinus Math, https://ctan.org/pkg/libertinust1math.
In the definitions of the \texttt{mathastext}-enriched \textit{math versions} we keep commands which may have been redundant in the original preamble, because they were issued earlier for a previous math version definition.

Let’s start with Latin Modern typewriter proportional. Its usage was configured in the preamble using this:

\begin{verbatim}
\MTlettershape{n}
\MTupgreek
\MTgreekfont{cmtt}
\MTfamily{lmvtt}
\Mathastext[lmvtt]
\end{verbatim}

Its usage is triggered using

\begin{verbatim}
\MTversion{lmvtt}
\end{verbatim}

in the document. Here is an example:

Let \((X, Y)\) be two functions of a variable \(a\). If they obey the differential system \((V_{\nu, n})\):

\[
\begin{align*}
\frac{a}{\mathrm{d}a}X &= \nu X - (1 - X^2) \frac{2na}{1 - a^2} (1 + aXY) \\
\frac{a}{\mathrm{d}a}Y &= -(\nu + 1)Y + (1 - Y^2) \frac{2na}{1 - a^2} (1 + aXY)
\end{align*}
\]

then the quantity \(q = \frac{aX + Y}{X + aY}\) satisfies as function of \(b = a^2\) the \(P_{V_{\nu}}\) differential equation:

\[
\begin{align*}
\frac{\mathrm{d}^2q}{\mathrm{d}b^2} &= \frac{1}{2} \left( \frac{1}{q} + \frac{1}{q-1} + \frac{1}{q-b} \right) \left( \frac{\mathrm{d}q}{\mathrm{d}b} \right)^2
- \left( \frac{1}{b} + \frac{1}{b-1} + \frac{1}{q-b} \right) \frac{\mathrm{d}q}{\mathrm{d}b}
+ \frac{q(q-1)(q-b)}{b^2(b-1)^2} \left( \frac{\alpha + \beta b + \gamma(b-1)}{q-1} + \frac{\delta b(b-1)}{(q-b)^2} \right)
\end{align*}
\]

with parameters \((\alpha, \beta, \gamma, \delta) = \left( \frac{(\nu+n)^2}{2}, \frac{-(\nu+n+1)^2}{2}, \frac{n^2}{2}, \frac{1-n^2}{2} \right)\).

Test of uppercase Greek in math: \(\text{ΑΒΓΔΞΩ}\).

Both the Latin and Greek letters are upright, in conformity to the way the \texttt{lmvtt} version was defined.

Now with the fonts from the \texttt{libertinus-type1} distribution\textsuperscript{11}. The preamble code is:

\begin{verbatim}
\MTfamily{LibertinusSerif-TLF}
\MTlettershape{n}
\MTseries{m}
\MTgreekfont{LibertinusSerif-TLF}
\MTupgreek
\Mathastext[libertinus]
\Mathastext[libertinussemibold]
\end{verbatim}

\textsuperscript{11}Bob Tennent, \textit{Support for using Libertinus fonts with \LaTeX/pdflatex}, \url{https://ctan.org/pkg/libertinus-type1}. 

9
Its usage in the document body for the example below is triggered via
\MTversion{libertinus}\{libertinussemibold\}
This syntax modifies the text fonts to be those which were defined to hold for the
mathastext\-math version passed as optional argument, and sets the math fonts
according to the mandatory argument. Hence the math mode uses semibold font
but the text fonts use the normal weight.

Let \((X, Y)\) be two functions of a variable \(a\). If they obey the differential
system \((VI_{\nu,n})\):
\[
\frac{d}{da}X = \nu X - (1 - X^2)\frac{2na}{1 - a^2} \frac{aX + Y}{1 + aXY}
\]
\[
\frac{d}{da}Y = -(\nu + 1)Y + (1 - Y^2)\frac{2na}{1 - a^2} \frac{X + aY}{1 + aXY}
\]
then the quantity \(q = a^{\frac{X+Y}{X+Y}}\) satisfies as function of \(b = a^2\) the \(P_{VI}\)
differential equation:
\[
\frac{d^2 q}{db^2} = \frac{1}{2} \left( \frac{1}{q + \frac{1}{b}} + \frac{1}{q - b} \right) \left( \frac{d q}{d b} \right)^2 - \left( \frac{1}{b + \frac{1}{b - 1}} + \frac{1}{q - b} \right) \frac{d q}{d b}
\]
\[
+ \frac{q(q-1)(q-b)}{b^2(b-1)^2} \left\{ \alpha \beta + \frac{\gamma(b-1)}{q^2} + \frac{\delta(b+1)}{(q-b)^2} \right\}
\]
with parameters \((\alpha, \beta, \gamma, \delta) = (\frac{\nu+n}{2}, -\frac{\nu+n+1}{2}, \frac{n^2}{2}, 1-n^2)\).

Test of uppercase Greek in math: \(\text{ΑΒΓ∆ΞΩ}\).

Now with a Times clone. We will configure Latin letters to be in italic shape,
and Greek letters to be italic for lowercase and upright for uppercase:
\usepackage{times}% it modifies the \{rm,sf,tt\}default's
\MTfamily{\rmdefault}
\MTlettershape{it}
\MTseries{m}
\MTgreekfont{txr}
\MTtitgreek\MTupGreek
\Mathastext{times}
% \MTversion{times} will change not only math but also text, so it
% will re-enact the \rmdefault, \sfdefault, \ttdefault from loading times.sty

We now use this in the document body via
\MTversion{times}

Let \((X, Y)\) be two functions of a variable \(a\). If they obey the differential
system \((VI_{\nu,n})\):
\[
\frac{d}{da}X = \nu X - (1 - X^2)\frac{2na}{1 - a^2} \frac{aX + Y}{1 + aXY}
\]
\[
\frac{d}{da}Y = -(\nu + 1)Y + (1 - Y^2)\frac{2na}{1 - a^2} \frac{X + aY}{1 + aXY}
\]
then the quantity \( q = a^\frac{\partial X + Y}{\partial X + Y} \) satisfies as function of \( b = a^2 \) the \( P_{VI} \) differential equation:

\[
\frac{d^2 q}{db^2} = \frac{1}{2} \left\{ \frac{1}{q} + \frac{1}{q - 1} + \frac{1}{q - b} \right\} \left( \frac{dq}{db} \right)^2 - \left\{ \frac{1}{b} + \frac{1}{b - 1} + \frac{1}{q - b} \right\} \frac{dq}{db} \\
+ \frac{q(q-1)(q-b)}{b^2(b-1)^2} \left\{ \alpha + \frac{\beta b}{q^2} + \frac{\gamma(b-1)}{(q-1)^2} + \frac{\delta b(b-1)}{(q-b)^2} \right\}
\]

with parameters \((\alpha, \beta, \gamma, \delta) = \left( \frac{\nu+\rho}{2}, \frac{-(\nu+\rho+1)}{2}, \frac{n^2}{2}, \frac{1-n^2}{2} \right)\).

Test of uppercase Greek in math: \( \text{ΑΒΓΔΞΩ} \).

Let us be a bit more original and have our mathematics with italic letters from the sans serif font Helvetica, while the letters in text use New Century Schoolbook. Also we want Greek letters (both lowercase and uppercase) to be in italic shape. The preamble code was:

\begin{verbatim}
\usepackage{newcent} % attention that it modifies all three of \rmdefault, \sfdefault and \ttdefault
\MTfamily{\rmdefault} % \MTfamily{\rmdefault}
\MTlettershape{it} % \MTlettershape{it}
% \MTgreek \MTgreek % our demo does not use newcent for math anyway
\Mathastext{\newcent}
\end{verbatim}

\begin{verbatim}
\usepackage{scaled}{helvet}
\MTfamily{\sfdefault}
\MTlettershape{it} % \MTlettershape{it}
\MTseries{m}
\MTgreek % \MTgreek
% make both lowercase and uppercase Greek italic
\MTgreekfont{cmss}
\Mathastext{\helvet}
\end{verbatim}

And the next demo is configured in the document body via

\begin{verbatim}
\MTversion{\newcent}{\helvet}
\end{verbatim}

Let \((X, Y)\) be two functions of a variable \(a\). If they obey the differential system \((VI_{\nu,\rho}):\)

\[
a \frac{d}{da} X = \nu X - (1 - X^2) \frac{2n a X + Y}{1 - a^2} + aXY \\
a \frac{d}{da} Y = -(\nu + 1) Y + (1 - Y^2) \frac{2n a X + aY}{1 - a^2} + aXY
\]

then the quantity \( q = a^\frac{\partial X + Y}{\partial X + Y} \) satisfies as function of \( b = a^2 \) the \( P_{VI} \) differential equation:

\[
\frac{d^2 q}{db^2} = \frac{1}{2} \left\{ \frac{1}{q} + \frac{1}{q - 1} + \frac{1}{q - b} \right\} \left( \frac{dq}{db} \right)^2 - \left\{ \frac{1}{b} + \frac{1}{b - 1} + \frac{1}{q - b} \right\} \frac{dq}{db} \\
+ \frac{q(q-1)(q-b)}{b^2(b-1)^2} \left\{ \alpha + \frac{\beta b}{q^2} + \frac{\gamma(b-1)}{(q-1)^2} + \frac{\delta b(b-1)}{(q-b)^2} \right\}
\]

\end{verbatim}
with parameters \((a, \beta, \gamma, \delta) = ((\nu+n)^2, -\frac{(\nu+n+1)^2}{2}, \frac{n^2}{2}, -\frac{n^2}{2})\).

Test of uppercase Greek in math: \(\Theta\Phi\Delta\Xi\Omega\).

And after all that, we may wish to return to the default math typesetting (let’s shorten the extract here in case the reader makes an indigestion . . . ). This is easy because all previous usages were enclosed in braces \{\ldots\} so as to limit the scope. As \texttt{mathastext} was loaded with option \texttt{subdued} the default rendering (i.e. in the \texttt{normal} and \texttt{bold} math versions) is (almost) as if the package was not loaded at all, and it simply matches the document font configuration. Here it thus matches the

\begin{verbatim}
\usepackage{mlmodern}
\end{verbatim}

which was included in the document preamble prior to loading \texttt{mathastext}.

Let \((X, Y)\) be two functions of a variable \(a\). If they obey the differential system \((VI_{\nu,n})\):

\[
\begin{align*}
\frac{d}{da}X &= \nu X - (1 - X^2) \frac{2na}{1 - a^2} \frac{aX + Y}{1 + aXY} \\
\frac{d}{da}Y &= -(\nu + 1)Y + (1 - Y^2) \frac{2na}{1 - a^2} \frac{X + aY}{1 + aXY}
\end{align*}
\]

then the quantity \(q = a\frac{aX + Y}{X + aY}\) satisfies as function of \(b = a^2\) the \(PVI\) differential equation with parameters \((\alpha, \beta, \gamma, \delta) = ((\nu+n)^2, -\frac{(\nu+n+1)^2}{2}, \frac{n^2}{2}, 1\frac{-n^2}{2})\).

Test of uppercase Greek in math: \(\Gamma\Delta\Xi\Omega\) (no \texttt{Alpha}, no \texttt{Beta}).

If the scope of our earlier examples using \texttt{mathastext}-enriched math versions had not been limited we would have issued

\begin{verbatim}
\MTversion{normal}
\end{verbatim}

to return to the normal (almost not influenced by \texttt{mathastext}) math version.

The Greek letters varied across our examples thanks to the \texttt{LGRgreek} option which made the \texttt{MTgreekfont} command active for configuration of the math versions.\(^{12}\)

Since 1.3x this documentation uses globally the \texttt{mlmodern}\(^{13}\) font package and has added an example using the Libertinus font in type-1 format\(^{14}\) although there is an existing accompanying math font\(^ {15}\).

---

\(^{12}\)The document used the \texttt{cmtt}, \texttt{cmmss}, \texttt{txr}, as well as LibertinusSerif-TLF font families in \texttt{LGR} encoding. The first two are available (with no need to load explicitly any package in the document) if the \texttt{ETeX} installation provides the \texttt{cbfonts} (or \texttt{cbgreek-complete}) & \texttt{babel} packages, and the \texttt{LGR} encoded \texttt{txr} font (again no package loading is necessary) is part of the files of the \texttt{txfontsb} package. For LibertinusSerif-TLF, the files of the \texttt{libertinus-type1} package must be present.

\(^{13}\)Daniel Benjamin Miller, \textit{A blacker Type 1 version of Computer Modern, with multilingual support}, \url{https://ctan.org/pkg/mlmodern}. I have added to the preamble

\begin{verbatim}
\DeclareEncodingSubset{TS1}{mlmtt}{0}
\end{verbatim}

to circumvent some \texttt{ETeX} complaints about \texttt{textasciigrave} (this is a widespread problem when not using default fonts) related to occurrences of the backtick character in verbatim displays.

\(^{14}\)Bob Tennent, \textit{Support for using Libertinus fonts with \texttt{ETeX}/\texttt{pdfETeX}}, \url{https://ctan.org/pkg/libertinus-type1}.

\(^{15}\)Michael Sharpe, \textit{A Type 1 font and \texttt{ETeX} support for Libertinus Math},
1.4 Main options

1.4.1 The italic option

In the initial version 1.0, the Latin letters in mathematical mode assumed the exact same shape as in text mode, and this meant, generally speaking, that they would turn up upright. Doing this gives a very uniform look to the document, so that one has to make an effort and read it with attention, and this was one of the design goals of \texttt{mathastext}.

Nevertheless, soon after I posted the initial version of the package to CTAN, I was overwhelmed by numerous\(^{16}\) questions\(^{17}\) on how to have the letters be in italic shape.

The default is still, as in version 1.0, for everything to be in upright shape, but it suffices to pass to the package the option \texttt{italic} to let the Latin letters in math mode be in italic shape.\(^{18}\)

1.4.2 The frenchmath option

It is a variant of the italic option which keeps the uppercase Latin letters in upright shape\(^{19}\). Also lets the Greek letters, if the latter are under \texttt{mathastext} influence, be all upright, lowercase as uppercase.

1.4.3 The defaultmathsizes option

The default sizes give for superscripts of subscripts barely legible glyphs (author’s opinion!). So \texttt{mathastext} makes more reasonable choices. It also redefines \texttt{\Huge} and defines a \texttt{\HUGE} size, copied from the \texttt{moresize} package. To cancel all of this use option \texttt{defaultmathsizes}.

1.4.4 The subdued option

This option was introduced in v1.15. It provides a manner to switch on the \texttt{mathastextification} only for limited portions of the document, with the help of the mechanism of math versions. Without the \texttt{subdued} option, the \texttt{mathastextification} applies by default to the whole of the document (and one may also define additional math versions in the preamble); with the \texttt{subdued} option the \texttt{mathastextification} is done only in \textit{math versions} distinct from the standard and bold ones.

Despite some limitations I will now partially describe, the \texttt{subdued} option has its utility, as I think is illustrated enough by the examples given at the start of this document and it works reasonably well.

\footnotesize
\begin{itemize}
  \item \url{https://ctan.org/pkg/libertinust1math}. Note that it is then highly advantageous to use latex+dvipdfmx and not pdflatex for reasons of PDF file size.
  \item this means “more then one.”
  \item I thank in particular Tariq \textsc{Perwez} and Kevin \textsc{Klement} for their kind remarks (chronological order).
  \item more precisely stated, the value of \texttt{\itdefault} is used.
  \item more precisely stated, the value of \texttt{\shapedefault} is used.
\end{itemize}

13
mathastext was not written initially in order to allow its action to be completely canceled. It does not store (all) mathcodes nor does it set them (all) when changing math versions; only that would allow a perfect subdued mode (and \LaTeX{} is rather obstinate in making that tricky or at least uneasy if sticking to its official interface to math mode, as it is almost entirely preamble only).

Releases 1.3t and 1.3u do this kind of things to maintain usability across multiple mathastext-ified math versions of characters which are obviously font encoding dependent such as the minus sign as en-dash (or unicode minus), the dotless i, the \hbar{}, the text accents.

But this should be extended to all mathastext-ified characters which basically would amount to an extensive rewrite of large legacy portions of the code. Currently the support for the subdued mode and to multiple math versions amounts to some kind of a kludge, added to an initial design which handled a single unique text font.

To get the displayed math (almost) as if mathastext had not been loaded, one must also use the option defaultmathsizes. But this does not quite suffice, as, for example, the colon, the dot, and the minus sign belong in the default \LaTeX{} math mode set-up to three distinct fonts whereas mathastext will pick (even subdued) the three of them in the same font,\(^{20}\) and although it will make a reasonable choice of this font, this is not an exact re-installlement of the previously prevailing situation. And then other packages could have done arbitrary things regarding character mathcodes, so to be on the safe side one needs the basic option which limits the mathastextification to letters and digits.\(^{21}\)\(^{22}\)\(^{23}\) Even then, in some circumstances, this may not suffice: for example the euler package declares the digits to be picked from the same font as the Latin letters, but the subdued mathastext “normal” math version will pick them from the same font as used for operator names, which here with the euler package is the document body default text font.

The frenchmath option effect applies also to the subdued “normal” and “bold” math versions.

---

\(^{20}\)The minus sign is now perfectly subdued, because its original mathcode is stored and restored; this was only way to handle the case with Unicode engines where the math operator font is in a classic \TeX{} encoding, but the minus sign is configured by mathastext to use a Unicode en-dash or minus character in non-subdued math versions.\(^{1.3t}\)

\(^{21}\)The subdued mode does extinguish in the normal and bold math versions the action of options selfGreek, eulergreek, and symbolgreek (previously only LGRgreek was subdue-able).\(^{1.3d}\)

\(^{22}\)The \$\textbackslash imath$ and \$\textbackslash jmath$ now obey the subdued regime.\(^{1.3t}\)

\(^{23}\)Also \$\textbackslash hbar$ and the math accents (see mathaccents option) obey the subdued regime.\(^{1.3u}\)
1.4.5 The \texttt{LGRgreek} option

There is the issue of Greek letters. Sometimes the text font has Greek glyphs, in LGR encoding\textsuperscript{24} (this should be mentioned in the documentation of the font package). Then option \texttt{LGRgreek} tells \texttt{mathastext} to pick up these Greek letters.

It is naturally possible to leave the responsability to set up Greek letters to some other packages loaded previously to \texttt{mathastext}. And even if \texttt{mathastext} has been loaded with one of its Greek related options the command \texttt{\MTstandardgreek} will locally cancel its customization of Greek letters. The command \texttt{\MTcustomgreek} reenables the customization done by \texttt{mathastext}, if it was loaded with the \texttt{LGRgreek} or one of the other Greek related options.

Release 1.3x has added important new aspects to the handling of Greek letters via the \texttt{LGRgreek} option. Make sure to read the subsubsection 1.7.3.

1.5 More miscellanea

This may repeat information already given.

1.5.1 Avoid \texttt{OT1} encoding

We specified in our minimal working example a \texttt{T1} encoding (\texttt{LY1} would have been ok, too) because the default \texttt{OT1} does not have the \texttt{< > | \{} \} and \texttt{\}} glyphs. If \texttt{mathastext} detects \texttt{OT1} as the default encoding it will leave these characters to their defaults from the math fonts.\textsuperscript{25}

If \texttt{mathastext} detects the obsolete \texttt{OT1} encoding it does not do anything with \texttt{<, >, |, \{} \} and \texttt{\}} which (except for monospace fonts) are not available in that encoding. To fully benefit from \texttt{mathastext} it is recommended to use some other encoding having these glyphs such as \texttt{T1} or \texttt{LY1}.

1.5.2 Derivative, minus, asterisk

The text characters ’ and - are not used, and the asterisk is done optionally:

\textsuperscript{24}For example the default CM and its replacement Latin Modern for european languages are (transparently to the user) extended with LGR encoded fonts from the \texttt{cbfonts (cbgreek-complete) TeXLive} package.

\textsuperscript{25}the \texttt{subdued} option, described next, acts a bit otherwise, it forces, contrarily to its usual low-key character, the replacement of \texttt{OT1} by \texttt{T1} for the fonts ultimately used with letters and digits in math mode.
• the derivative sign \textquotesingle is left to its default as the text font glyph \textquotesingle is not, as a rule, a satisfying alternative.\footnote{v1.2 adds a customizable tiny space before \textquotesingle to separate it from the previous letter, this is really needed when using upright letters in math mode with the CM derivative glyph. Compare \textit{f} with \textit{f}'.}

• for the minus sign \texttt{mathastext} uses the endash character –, if available, and not the hyphen character -.\footnote{see the \texttt{unicodeminus} option if using an OpenType font.}

• the \texttt{asterisk} option is mandatory for \texttt{mathastext} to replace the binary math operator \texttt{*} (and the equivalent control sequence \texttt{\ast}) with a version which uses the text asterisk * suitably lowered\footnote{the amount of lowering can be customized.} (and with the correct spaces around it as binary operator). The reason is that after this inputs such as $R^{-*}$ or $R^{-\ast}$ raise errors and must be written $R^{-(*)}$ or $R^{-(\ast)}$.

Nothing is changed to the “large” math symbols, except for $\prod$ and $\sum$ in inline math which, like here: $\prod\sum$, will be taken from the Symbol Font if option symbol-misc was used. The left and right delimiters are taken from the text font only for the base size: any \texttt{\big}, \texttt{\bigl}, \texttt{\bigr}, etc... reverts to the original math symbols.

1.5.3 Load \texttt{mathastext} always last

The “large” math symbols are not modified in any way by \texttt{mathastext}. Only loading some math font packages such as \texttt{fourier}, \texttt{kpfonts}, \texttt{mathabx}, \texttt{mathdesign}, \texttt{txfonts}, \texttt{newtxmath}, etc... will change them. Think of loading these packages before \texttt{mathastext}, else they might undo what \texttt{mathastext} did.

More generally any package (such as \texttt{amsmath}) dealing with math mode should be loaded before \texttt{mathastext}.

1.5.4 Sans serif in math

The following set-up often gives esthetically pleasing results: it is to use the sans-serif member of the font family for math, and the serif for text.

\begin{verbatim}
\renewcommand\familydefault{sfdefault}
\usepackage{mathastext}
\renewcommand\familydefault{rmdefault}
\begin{document}
\end{verbatim}

1.5.5 \texttt{mathastext} with \texttt{beamer}

Starting with release 3.34 of \texttt{beamer}\footnote{Till \textsc{Tantau}, Joseph \textsc{Wright}, Vedran \textsc{Miletić}, \textit{\LaTeX} class for producing presentations and slides, \url{https://ctan.org/pkg/beamer}.}, \texttt{mathastext} is recognized as a “math font package”.
Only with earlier beamer versions is it necessary to issue \usefonttheme{professionalfonts} in the preamble. Example:

\documentclass{beamer}
%\usefonttheme{professionalfonts}% obsolete for mathastext since beamer 3.34
\usepackage[newcent]{newcent}
\usepackage[scaled=.9]{helvet}
\renewcommand{\familydefault}{\rmdefault}
\usepackage[defaultmathsizes,symbolgreek]{mathastext}
\renewcommand{\familydefault}{\sfdefault}
\begin{document}
\begin{frame}
This is some text and next comes some math: $E=mc^2$
\[
E=mc^2=a^n+b^n-c^n=\alpha\beta\gamma
\]
\begin{align}
E&=mc^2
\E&=h\nu
\end{align}
And again some text.
\end{frame}
\end{document}

1.5.6 mathastext with frenchmath

To use mathastext concurrently with the frenchmath package\textsuperscript{30} \textsuperscript{31} of Antoine Missier:

- load frenchmath with its option capsit,

- and load mathastext afterwards (with possibly some font packages loaded in-between), passing it the option frenchmath*.

Limited testing indicated that the combination of the two packages (using the options as indicated above) works satisfactorily. There may be some minor adjustments to do, as the mathastext-ified math font may cause issues to some of the frenchmath macros: for exemple \Oijk may not work well simply due to the font lacking a dotless j, but use then defaultimath.

You can either use the Greek related options of frenchmath or those of mathastext. Quite certainly better not to use both at same time, anyhow this has not been tested and is not supported.

\textsuperscript{30}Antoine Missier, Typesetting mathematics according to French rules, \url{https://ctan.org/pkg/frenchmath}.

\textsuperscript{31}The package mismath also by the Antoine Missier may probably be used with mathastext, but not in a fully inter-operative way, as the two packages conflict on some aspects. Reports welcome, we have not tested this.
1.5.7 Intervals and separators

For appropriate mark-up and typesetting of intervals with conventions about opening and closing delimiters which are not the default \( \text{T}_{\text{e}}\text{X} \) ones, one may use the `mathtools` provided \( \text{\texttt{\textbackslash DeclarePairedDelimiterX}} \). For example, here is how one can define an \( \text{\textbackslash Ioo} \) macro (the letter “o” standing for “open”) for typesetting an open (in the mathematical meaning of the word) interval using square brackets:

\[
\text{\texttt{\textbackslash DeclarePairedDelimiterX\textbackslash Ioo}[2]{\}]{{}\{\}#1;#2}\]

Use then \$I = \text{\textbackslash Ioo}\{A\}\{B\}$ type mark-up in your source, and the derived variants \( \text{\textbackslash Ioo}^* \) or \( \text{\textbackslash Ioo}[\text{\Big}] \) for example will also work.

Note for very advanced users: if employing \( \text{\texttt{\textbackslash MTnonlettersobeymathxx}} \), our \( \text{\textbackslash Ioo} \) must be used as \( \text{\textbackslash Ioo}^* \) or \( \text{\textbackslash Ioo}[\text{\Big}] \) (for example) else it raises an error. Alternatively, replace in the above \[ \] by \{\} and \[ \] by \{\} and then \( \text{\textbackslash Ioo} \) works (and also \( \text{\textbackslash Ioo}[\text{\Big}] \)). But \( \text{\textbackslash Ioo}^* \) is broken. This is a known limitation of the \( \text{\texttt{\textbackslash MTnonlettersobeymathxx}} \) functionality, and is one reason why `mathastext` does not make it the default behavior.

We used in this example the semi-colon as separator. This is seen sometimes in contexts where the interval extremities are decimal numbers, and the language convention is to use the comma as decimal point. The `binarysemicolon` option tells `mathastext` to configure the ; character to use in math mode “binary infix operator” type spacing, matching observed practice in some mathematical contexts. The `binarysemicolon` option is executed automatically by `mathastext` on receiving either the `frenchmath*` or the `frenchmath+` options.

On the topic of the decimal point, it is recommended to use the \( \text{\textbackslash np} \) macro from the `numprint` package with its `autolanguage` and `np` options. This is the best choice if one may have to also use the same mathematical expression with numerical quantities in another language having different conventions.

For those languages such as French where the convention is to use as decimal separator a colon, you may alternatively pass to `mathastext` either the `decimalcomma` or the `ncccomma` options, to tell it to load the eponymous packages `decimalcomma` or `ncccomma` respectively, which make the comma (to some extent) ‘intelligent’, i.e. decide on the spacing type (ordinary or punctuation) depending on next token. Do not load directly the packages but simply use the corresponding option and `mathastext` will do the loading and take appropriate needed measures for compatibility. The `decimalcomma` option is included in the `frenchmath*` option, and the `ncccomma` option is included in the `frenchmath+` option.

Let’s give another example of usage of `mathtools` here to define a macro for integer ranges:

\[
\text{\texttt{\textbackslash DeclarePairedDelimiterX\textbackslash Iffint}[2]{\llbracket}{\rrbracket}{#1,#2}}\]

---

32 Morten Høgholm, Lars Madsen and the \( \text{\texttt{\textbackslash L\text{\textbackslash A} \text{T}_{\text{e}}\text{X}}} \) project, *Mathematical tools to use with amsmath*, https://ctan.org/pkg/mathtools. As explained elsewhere in this documentation always load `mathastext` after `mathtools`.


This used control sequences \llbracket and \rrbracket from the fourier package (and possibly others). A poorman definition might be:
\ifdefined\llbracket\else\def\llbracket{\{\![}\}\fi
\ifdefined\rrbracket\else\def\rrbracket{\]!\}\fi

Regarding open intervals in the French notation such as ]a,b[, an alternative avoiding usage of specific mark-up is provided by the ibrackets\footnote{Antoine Missier, \textit{Intelligent brackets}, \url{https://ctan.org/pkg/ibrackets}.} package which makes the square brackets mathematically active, in the same spirit as for the (semi) ‘intelligent’ comma mentioned above. Brief testing indicates this package is compatible with \texttt{mathastext}, even when using multiple math versions. Read the fine print below for some limitations though.

Note for very advanced users: compatibility is only partial as the effect of ibrackets is canceled after  Mathastexobeymathx . This is expected and a special compatibility layer would be needed, of the same type as has been done to support fully the decimalcomma and ncccomma packages via eponymous mathastex options. There is no plan at this time to add such a patch making the compatibility exhaustive.

It is possible to use the noparenthesis option to turn off completely the mathastex actions on square brackets (and parentheses).

1.6 Math versions

\LaTeX{} has the concept of math versions\footnote{Math versions are discussed in the document fntguide.pdf from your \TeX{} distribution.}, but most font packages do not define any such version beyond the default normal and bold (the package unicode-math for unicode engines does use this concept).

\texttt{mathastex} extends the concept of math versions in order to allow the math mode fonts (for letters, digits, punctuation and a few other ascii symbols) used in the different parts of the document to be kept in sync with the text fonts.

Most math symbols (sums, products, integrals, logical signs, etc...) are kept the same throughout the document though as it is not in \texttt{mathastex} power to modify them.

For examples see the earlier subsection 1.3. The interface to define a math version includes the commands \texttt{\Mathastext} and \texttt{\MTDeclareVersion}.

Once such a math versions has been defined in the preamble, \texttt{\MTversion{name_of_version}}, or equivalently \texttt{\Mathastextversion{name_of_version}}, enacts the font switches when encountered in the body of the document. As is usual with \LaTeX{} one can limit the scope to inside a group, or also switch back to the main set-up via \texttt{\Mathastextversion{normal}}.

When \texttt{\Mathastext} is used in the preamble, it records the current text font defaults (\texttt{\familydefault} et al. or what has been configured by \texttt{\MTfamily} and similar commands) and (except for the normal and bold versions if in subdued regime) sets up both the math font and the text font in the defined mathastex-math version to be this text font. It is still possible to switch on via \texttt{\MTversion} in the document body distinct fonts for text and math: an optional argument (the name of another mathastex-declared math version) to \texttt{\MTversion} is allowed.
(such as for example \texttt{\textbackslash MTversion[newcent]{helvet}} for one of the examples of
the subsection 1.3). It instructs to use as text font the font which was configured
to be used in this second \texttt{mathastext}-math version.\footnote{When not using math versions at all (so not using subdued mode either) another way to achieve
distinct fonts in text and math is naturally to modify the document text font \texttt{after} having loaded\texttt{mathastext} (or after last usage of \texttt{Mathastext} without optional argument). Another way is to
use \texttt{MTfamily}, \texttt{MTencoding}, \texttt{MTseries}, \texttt{MTshape}, \texttt{MTlettershape} in the preamble before a
call to \texttt{Mathastext} which will configure math fonts without having modified the document text
fonts. However if one does \texttt{MTversion(normal)} in the document then the text font will be reset
to what was recorded as math font by the \texttt{Mathastext} call in the preamble (as said above, when
not using subdued option).}

The native \LATEX{} command \texttt{\textbackslash mathversion} is \texttt{\langle version\_name \rangle} would change only
the fonts for the math mode, not the text mode fonts. It is important to
use rather the package command \texttt{\textbackslash MTversion} (or one of its synonyms \texttt{\textbackslash mathas-
textversion}, \texttt{\textbackslash Mathastextversion}, \texttt{\textbackslash MTVersions}), with its mandatory argument
\texttt{\langle version\_name \rangle}, as it does additional actions:

- it sets the font for math mode (letters, math operator names, digits, punctua-
tions, some other symbols) according to the version name given as mandatory
argument,

- it resets the text font of the document and the \texttt{\langle family,rm,sf,... \rangle defaults}
to their values as registered at the time of definition of the version. \textit{Use the
starred variant in case this is not desired.} As explained above it is possible
to specify within brackets an extra optional version name, and the text font
will be set according to it.

For all math versions if not using the \texttt{subdued} option, or only for the \texttt{non-normal}
and \texttt{non-bold} math versions if using the \texttt{subdued} option, \texttt{\textbackslash MTversion} does further
additional tasks:

- it resets the \texttt{\hbar}, \texttt{\imath} (see \texttt{\inodot}), \texttt{\jmath}, math accents (see option\footnote{\texttt{mathaccents}})
and minus sign as en dash according to the used font encoding
for the \texttt{mathastext}-ified text font,

- (see sections 1.8 and 1.9) it re-issues the command \texttt{\textbackslash MTmathactiveletters}
to let a to z, A to Z, be mathematically active in order to automatically
insert the skips as defined by the user with \texttt{\textbackslash MTsetmathskips}, and the italic
corrections (if the font is not italic or slanted),

- (see section 1.10) it resets the extra spaces after the symbols \texttt{\exists}, \texttt{\forall} and before
the derivative \texttt{\prime} to the values as decided by the user in the preamble on a \textit{per version}
basis,

- (see section 1.11) it re-issues the commands \texttt{\textbackslash MTmathoperatorsobeymathxx}
and \texttt{\MTeasynonlettersobeymathxx} to let the math operator names and
(‘easy’) non letter characters obey the math alphabets,
• in case of option \texttt{asterisk}, it re-issues \texttt{\MTactiveasterisk},

• it does the additional set-up for Greek letters in case of the package received one of the Greek related options.

The scope is limited to the current \LaTeX{} environment or group.

It is sometimes not compatible with \texttt{mathastext} to load a font package after it, as the font package may contain instructions which will modify the math set-up. This may be a bit hidden to the user: for example the \texttt{epigrafica} package loads \texttt{pxfonts}. Hence it will interfere with \texttt{mathastext} if it is loaded after it.\footnote{\texttt{may typically give a ‘too many math alphabets’ error message.}} But one can use instead \texttt{\renewcommand\{\rmdefault\}{epigrafica}},\footnote{\texttt{sometimes one needs to look in the .sty file of the font package to figure out the font name (it is rarely as here with epigrafica, the same as the package name), and, if one does not know the arcana of finding .fd files in one’s \TeX{} distribution, one should look at the log file of a test document to see if for example T1 is available for that font; for epigrafica it is not, only OT1 and LGR are possible.}} followed with \texttt{\MTfamily{epigrafica}\MTastext} which will only change the font in math.

To use \texttt{epigrafica} for Greek in math mode one can use the package option \texttt{LGRgreek} and the command \texttt{\MTgreekfont{epigrafica}\MTastext}. Or \texttt{\usepackage{epigrafica} \usepackage[LGRgreek]{mathastext}}.

\section*{1.7 Greek letters}

\subsection*{1.7.1 The Greek-related options}

The Computer Modern fonts are very light and thin in comparison to many text fonts, and as a result rarely mix well with them (particularly if the Latin letters in math mode are upright). The following options are provided by \texttt{mathastext}:

\textbf{no option}: nothing is done by the package, Greek letters are the default Computer Modern ones or have been set-up by other packages; for example by the \texttt{fourier} package with option ‘upright’, which gives upright Greek letters.

\textbf{LGRgreek}: (this was substantially updated at 1.3x, make sure to read the new documentation at \texttt{subsubsection 1.7.3}) this option is for fonts which additionally to Latin letters also provide Greek letters in LGR encoding. Here is a list from a 2012 standard \TeX{} installation: the Computer Modern, Latin Modern, and the CM-LGC fonts; the Greek Font Society fonts (such as GFS Didot), the epigrafica and kerkis packages, the txfonts package which extends the txfonts package with LGR-encoded Greek letters; the Droid fonts, the DejaVu fonts, the Comfortaa font, and the Open Sans font. The LGR encoded \texttt{CM/LM} fonts (in serif, sans-serif and typewriter family) give the nice Greek letters in upright shape from the \texttt{cbfonts} package. To get these letters in your \texttt{mathastext} math mode, you can do the following:
% instructions to load the document fonts:
\usepackage{nice_font}
% and then the following:
\renewcommand{\familydefault}{cmr} % or cmss or cmtt for sans resp. mono
\usepackage[LGRgreek]{mathastext}
\renewcommand{\familydefault}{\rmdefault}
\Mathastext % this re-initializes mathastext with the nice_font,
% without changing the LGR font cmr/cmss/cmtt used for Greek letters
% in math mode.
\begin{document}
If you use the italic option note that the italic Greek letters from the
cbfonts are not the same glyphs as the default Greek letters from the OML
encoded font cmmi.

**LGRgreek+:** extends LGRgreek to allow abusive usage of \mathrm and alike com-
mands with Greek letters. This is very much not in the spirit (especially with
traditional “8bit” \TeX fonts) of the \LaTeX kernel concept of math alphabet
commands. Check \texttt{subsection 1.7.5} for relevant information.

**eulergreek:** the Greek letters will be taken from the Euler font (the document
does not have to load the eulervm package, \texttt{mathastext} directly uses some file
included in this package, as it provides a mechanism to scale by an arbitrary
factor the Euler font.) The letters are upright.

**symbolgreek:** the Greek letters will be taken from the (Adobe Postscript) Symbol
font. A command is provided so that the user can scale the Symbol font to
let it better fit with the text font. The letters are upright.

**selfGreek:** this option concerns only the eleven Greek capitals from the OT1-
encoding. It does nothing for the lowercase Greek letters. The encoding
used in the document does not have to be OT1.

There is also **LGRgreeks** (and **LGRgreeks+**) which tells \texttt{mathastext} to pick up in
each math version the letters from the LGR encoded font used in that version, and
**selfGreeks** to tell \texttt{mathastext} to do as for **selfGreek** but separately in all math
versions.

Under the subdued option the Greek letters in the normal and bold math versions
are kept to their defaults as found at the time of loading the package.

The commands \texttt{\MTstandardgreek} allow at any point in the document to turn
inactive any Greek related option passed to \texttt{mathastext}. And conversely \texttt{\MTcus-
tomgreek} reactivates it.

### 1.7.2 Shape of Greek letters

Classic \TeX uses in math mode italic lowercase and upright uppercase Greek letters.
French typography uses upright shape for both lowercase and uppercase. And the
ISO standard is to use italic shape for both lowercase and uppercase.
The Euler and Symbol fonts not being available in other than their default upright shape, this question of shapes for Greek letters raises issues only in the case of the options \texttt{LGRgreek} and \texttt{selfGreek}.

The options \texttt{frenchmath}, \texttt{itgreek}, \texttt{upgreek}, \texttt{itGreek} and \texttt{upGreek} modify the Greek letter shapes according to the following rules, listed from the lowest to the highest priority:

**no option:** the lowercase Greek letters are in the same shape as Latin letters, and the uppercase in the same shape as is applied to digits and operator names,

**frenchmath:** both lowercase and uppercase are in the same shape as the digits and operator names (most of the time this means “upright shape”, but it can be otherwise),

**itgreek:** says that Greek letters (both lowercase and uppercase) will be in ‘it’ shape. More precisely the expansion of \texttt{\MTgreekitdefault} is used.

\begin{itemize}
\item changed: This was changed at 1.3x, formerly the value of \texttt{\itdefault} which was in force at the time of using \texttt{\Mathastext} (or at time of loading the package if no use is made of \texttt{\Mathastext}) was used.
\end{itemize}

**upgreek:** says that Greek letters (both lowercase and uppercase) will be in ‘n’ shape. More precisely the expansion of \texttt{\MTgreekupdefault} is used.

\begin{itemize}
\item changed: This was changed at 1.3x, formerly the value of \texttt{\updefault} which was in force at the time of using \texttt{\Mathastext} (or at time of loading the package if no use is made of \texttt{\Mathastext}) was used. But since \LaTeX{} 2020-02-02 this caused many Font Warnings in the log because \texttt{\updefault} is now ‘up’, not ‘n’ as formerly.
\end{itemize}

**itGreek, upGreek:** same but they apply only to the uppercase Greek letters. Their effect is computed after having taken into account either \texttt{itgreek} or \texttt{upgreek} presence.

So, the default gives the classic \TeX{} behavior when option \texttt{italic} was passed.

As mentioned already the package allows to define various “math versions”. There are commands to be used inside the preamble to influence the shapes, and even the font, used for Greek letters in each given \texttt{\Mathastext}-declared math version: \texttt{\MTitgreek}, \texttt{\MTupgreek}, \texttt{\MTitGreek}, \texttt{\MTupGreek} and \texttt{\MTgreekfont\texttt{name_of_font}}.

Their effect is as the options of the alike name, except that the effect applies only to \texttt{\Mathastext}-math versions declared \texttt{next} in the preamble (be it via \texttt{\Mathastext} or \texttt{\MTDeclareVersion}).

To use \texttt{\MTgreekfont} you need to know the name of a suitable font family available in LGR encoding: for example \texttt{lmr}, \texttt{txr} (needs \texttt{txfonts} package on your system), \texttt{DejaVuSerif-TLF} (needs \texttt{dejavu} package on your system), etc...
\MTitgreek, \MTupgreek, \MTitGreek, \MTupGreek have some effect only if one of the LGRgreek, LGRgreeks, selfGreek or selfGreeks options was passed to the package.

Once any of these commands has been made use of, changes in the shape configuration of the Latin letters will stop having any influence on the shape of the Greek letters.

\MTgreekfont has an effect only for LGRgreek and selfGreek. It is without any effect with LGRgreeks and selfGreeks.

1.7.3 Control sequences to access directly upright or italic shape for Greek under LGRgreek option

Some changes were made at 1.3x to enhance the LGRgreek (and LGRgreeks) options with new features. Everything which will be explained here applies only to these two options.

First of all the package now makes available control sequences to access either the upright or italic shape of the Greek letters: \alphaup, \alphait, etc... Which shape is meant by 'up' or 'it' is configured via defining \MTgreekupdefault and \MTgreekitdefault respectively prior a \Mathastext command in the preamble (possibly with \[〈version_name〉\] optional argument). Their default definitions are to expand to '\n' and '\it' respectively. They can also be defined prior to loading mathastext.

See the Table 1 and Table 2 for illustrations (using here the Libertinus Serif font).

The regular control sequences without 'up' or 'it' postfix will map to either one of the two versions according to how the shapes were configured, i.e. in almost all cases via usage of either the itgreek, upgreek, etc... options or \MTtitgreek et al. commands. This is on a per mathastext-enriched math version basis, depending only on how the options or commands were used in the preamble.

Furthermore two math alphabets are provided

\mathgreekup
\mathgreekit

which can be used to map a letter to the corresponding Greek fonts:
\begin{align*}
\mathgreekup\{a\} &= \mathgreekup\{\alpha\} = \mathgreekup\{\alphait\} = \alphaup \\
\mathgreekup\{G\} &= \mathgreekup\{\Gamma\} = \mathgreekup\{\Gammait\} = \Gammaup \\
\mathgreekit\{z\} &= \mathgreekit\{\zeta\} = \mathgreekit\{\zetait\} = \zetait
\end{align*}

\[\zeta = \zeta = \zeta = \zeta\]

\[\zeta = \zeta = \zeta = \zeta\]

\[\zeta = \zeta = \zeta = \zeta\]

\[\zeta = \zeta = \zeta = \zeta\]

No check is done of pre-existing such math symbol, they will be replaced by the mathastext definition with no warning. If they happen to be pre-defined as \LaTeX commands, not as math symbols, errors will happen during the loading of mathastext.
| \Alphait A | \Xiiit Ξ | \alphait α | \xiiit ξ |
| \Betait B | \Omicronit O | \bettait β | \omicronit o |
| \Gammatit Γ | \Piit π | \gammait γ | \piit π |
| \Deltait Δ | \Rhoit ρ | \deltait δ | \rhoit ρ |
| \Epsilontit E | \Sigmait Σ | \epsilontit ε | \sigmait σ |
| \Zetait Ζ | \Tait τ | \zetait ζ | \tait τ |
| \Etait H | \Upsilontit Y | \etait η | \upsilontit υ |
| \Thetait Θ | \Phiiit ϕ | \thetait θ | \phiit ϕ |
| \Iotait I | \Chiit χ | \iotait ι | \chiit χ |
| \Kappaait K | \Psiiit ψ | \kappaait Ϲ | \psiit ψ |
| \Lambetait Λ | \Omegaait Ω | \lambait λ | \omegait ω |
| \Muait M | \Digammait F | \muait μ | \digammait ν |
| \Nuait N | \nuait ν | \nuait ν | \nuait ν |

Table 1: Greek letters via ‘up’ control sequences (math mode only)

| \Alphait A | \Xiiit Ξ | \alphait α | \xiiit ξ |
| \Betait B | \Omicronit O | \bettait β | \omicronit o |
| \Gammatit Γ | \Piit π | \gammait γ | \piit π |
| \Deltait Δ | \Rhoit ρ | \deltait δ | \rhoit ρ |
| \Epsilontit E | \Sigmait Σ | \epsilontit ε | \sigmait σ |
| \Zetait Ζ | \Tait τ | \zetait ζ | \tait τ |
| \Etait H | \Upsilontit Y | \etait η | \upsilontit υ |
| \Thetait Θ | \Phiiit ϕ | \thetait θ | \phiit ϕ |
| \Iotait I | \Chiit χ | \iotait ι | \chiit χ |
| \Kappaait K | \Psiiit ψ | \kappaait Ϲ | \psiit ψ |
| \Lambetait Λ | \Omegaait Ω | \lambait λ | \omegait ω |
| \Muait M | \Digammait F | \muait μ | \digammait ν |
| \Nuait N | \nuait ν | \nuait ν | \nuait ν |

Table 2: Greek letters via ‘it’ control sequences (math mode only)
Some refactoring\textsuperscript{42} was required to achieve this at 1.3x and it is not 100\% backwards compatible: if none of the \texttt{itgreek} etc. things was used, the Greek letters formerly would follow the shape of Latin letters (for lowercase Greek) and of operator names (for uppercase Greek). Now, some check is made for each of these two shapes whether it is ‘it’ or ‘sl’ and then the ‘italic’ shape, i.e. \texttt{\MTgreekitdefault} which by default is ‘it’ (without the quotes) is used, else the ‘upright’ shape, i.e. \texttt{\MTgreekupdefault} which by default expands to ‘n’ (without the quotes) is used. Naturally these checks are done on a per \texttt{mathastext}-math version basis, if multiple math versions are used.

So for example those who used some adventurous ‘sc’ for the main shape (the one used per default for operator names) and used the option \texttt{LGRgreek} but none of the \texttt{itgreek} et al. options, and none of the \texttt{\MTitgreek} et al. commands, now will need to adjust \texttt{\MTgreekupdefault} to expand to ‘sc’ prior to some \texttt{\Mathastext} or \texttt{\Mathastext[(version\_name)]} or \texttt{\MTDeclareVersion} in the preamble depending on context.

It is hoped most documents, even those using multiple math versions, which made use of the \texttt{LGRgreek} (or \texttt{LGRgreeks}) option will simply produce unmodified output. Please report to the author unexpected results not fitting the above attempted description of the only partial backwards compatibility.

\subsection*{1.7.4 \texttt{\mathgreekupbold} and \texttt{\mathgreekitbold}}

Again this applies only to \texttt{LGRgreek} and \texttt{LGRgreeks} options (and the 1.3za added \texttt{LGRgreek+} and \texttt{LGRgreeks+}).

See the Table 3 and Table 4 for illustration of usage (in math mode only) of code\textsuperscript{(1.3za)} such as

\begin{verbatim}
\mathgreekupbold{\alpha}
\end{verbatim}

or

\begin{verbatim}
\mathgreekitbold{\alpha}
\end{verbatim}

Note that all three of \texttt{\alpha}, \texttt{\alphaup} and \texttt{\alphait} would give the same output. These two tables again use the Libertinus Serif font via an \texttt{mathastext} math version which was configured in the preamble using this set-up (and the package \texttt{LGRgreek} option):

\begin{verbatim}
\MTfamily{LibertinusSerif-TLF}
\MTlettershape{n}
\MTseries{m}
\MTgreekfont{LibertinusSerif-TLF}
\MTupgreek
\Mathastext[libertinus]
\end{verbatim}

\textsuperscript{42}Technically, formerly two symbol fonts were declared, one for the lowercase Greek letters and one for the uppercase Greek letters; now those are dropped and replaced by two symbol fonts, one for ‘italic’ Greek letters, the other for ‘upright’ Greek letters.
Some examples here to illustrate the effect of the math alphabet commands on Latin letters also:

\[ \mathfrak{a} = \alpha \]
\[ \mathfrak{G} = \Gamma \]
\[ \mathfrak{z} = \zeta \]
\[ \mathfrak{W} = \Omega \]

\[ \mathfrak{A} \rightarrow \mathbb{A} \]
\[ \mathfrak{B} \rightarrow \mathbb{B} \]
\[ \mathfrak{G} \rightarrow \mathbb{G} \]
\[ \mathfrak{Z} \rightarrow \mathbb{Z} \]
\[ \mathfrak{W} \rightarrow \mathbb{W} \]

\[ \mathfrak{A} \rightarrow \mathfrak{a} \]
\[ \mathfrak{B} \rightarrow \mathfrak{b} \]
\[ \mathfrak{G} \rightarrow \mathfrak{g} \]
\[ \mathfrak{Z} \rightarrow \mathfrak{z} \]
\[ \mathfrak{W} \rightarrow \mathfrak{w} \]

Table 3: Greek control sequences in the argument of \texttt{\textbackslash mathgreekupbold}.

1.7.5 Special behavior of \texttt{\textbackslash mathrm}, \texttt{\textbackslash mathbf}, \texttt{\textbackslash mathit} with Greek letters via the \texttt{LGRgreek+} option

With option \texttt{LGRgreek+} or \texttt{LGRgreeks+}, \texttt{mathastext} makes Greek letters control (1.3za) sequences \texttt{\textbackslash alpha}, \texttt{\textbackslash beta}, ... (but not \texttt{\textbackslash alphau} or \texttt{\textbackslash betait} and the others) react in a special manner within the scope of \texttt{\textbackslash mathnormal}, \texttt{\textbackslash mathrm}, \texttt{\textbackslash mathit}, \texttt{\textbackslash mathbf}, and \texttt{\textbackslash mathnormalbold}, but not further math alphabet commands, and not when using the \texttt{mathastext} defined commands named with an uppercased initial.
<table>
<thead>
<tr>
<th>Greek Symbol</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Alpha</td>
<td>\mathnormal \alpha</td>
</tr>
<tr>
<td>\Beta</td>
<td>\mathit \beta</td>
</tr>
<tr>
<td>\Gamma</td>
<td>\mathbf \Gamma</td>
</tr>
<tr>
<td>\Delta</td>
<td>\mathit \Delta</td>
</tr>
<tr>
<td>\Epsilon</td>
<td>\mathbf \Sigma</td>
</tr>
<tr>
<td>\Zeta</td>
<td>\mathit \zeta</td>
</tr>
<tr>
<td>\Eta</td>
<td>\mathbf \zeta</td>
</tr>
<tr>
<td>\Theta</td>
<td>\mathit \Theta</td>
</tr>
<tr>
<td>\Iota</td>
<td>\mathbf \Iota</td>
</tr>
<tr>
<td>\Kappa</td>
<td>\mathit \Kappa</td>
</tr>
<tr>
<td>\Lambda</td>
<td>\mathbf \Lambda</td>
</tr>
<tr>
<td>\Mu</td>
<td>\mathit \Mu</td>
</tr>
<tr>
<td>\Nu</td>
<td>\mathbf \Nu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Greek Symbol</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Xi</td>
<td>\mathnormal \Xi</td>
</tr>
<tr>
<td>\Omicron</td>
<td>\mathit \Omicron</td>
</tr>
<tr>
<td>\Pi</td>
<td>\mathbf \Pi</td>
</tr>
<tr>
<td>\Delta</td>
<td>\mathit \Delta</td>
</tr>
<tr>
<td>\Sigma</td>
<td>\mathbf \Sigma</td>
</tr>
<tr>
<td>\Upsilon</td>
<td>\mathit \Upsilon</td>
</tr>
<tr>
<td>\Psi</td>
<td>\mathbf \Psi</td>
</tr>
<tr>
<td>\Chi</td>
<td>\mathit \Chi</td>
</tr>
<tr>
<td>\Psi</td>
<td>\mathbf \Psi</td>
</tr>
<tr>
<td>\Digamma</td>
<td>\mathit \Digamma</td>
</tr>
</tbody>
</table>

Table 4: Greek control sequences in the argument of the \mathgreekitbold command.
This font has no bold italic Digamma nor digamma (last tested 2023/12/19).

Here is an example

\begin{array}{rc}
\text{abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathnormal abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathrm abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathit abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathbf abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathnormalbold abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekup abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekit abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekupbold abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekitbold abCD\alpha\pi\Delta\Gamma} & \text{abCD\alpha\pi\Delta\Gamma} \\
\end{array}

It used this source:

\begin{verbatim}
\edef\zzz{abCD\alpha\pi\Delta\Gamma}
\begin{array}{rc}
&\zzz\\ &\text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathnormal} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathrm} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathit} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathbf} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathnormalbold} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekup} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekit} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekupbold} & \text{abCD\alpha\pi\Delta\Gamma} \\
\text{mathgreekitbold} & \text{abCD\alpha\pi\Delta\Gamma} \\
\end{array}
\end{verbatim}
This was typeset here using a “libertinustexstyle” math version which (differently from the one used in an earlier section) has the default \TeX settings for the shape of Latin and Greek letters: i.e. italic Latin and lowercase Greek, upright uppercase Greek. Its preamble definition was something like this:

\MTfamily{LibertinusSerif-TLF}
\MTgreekfont{LibertinusSerif-TLF}
\MTlettershape{it}% not needed with italic option if nothing was changed prior
\MTitgreek\MTupGreek% this is also the default configuration
\Mathastext{libertinustexstyle}

The difference with using only \textgreek option is that with the latter the Latin math alphabets such as \textrm, \textit, \textbf produce a Latin letter when acting on a Greek control sequence, as the latter are defined by \mathastext under \textgreek to be of “variable family type” for usage with \textgreekup and \textgreekit. With \textgreek+, the Greek control sequences are not mathchar tokens anymore but macros with conditionals detecting some flag set by custom \textnormal, \textnormalbold, \textrm, \textit, and \textbf.

\textastext has no logical way to sync shape of Latin and Greek letters once usage has been made of Greek related commands. Hence \textnormal is currently configured to do nothing on Greek letters. This may change, please consider this behavior unstable.

Remark: this \textgreek+ functionality is considered by its author an abuse of the concept of a math alphabet command and required accommodating a serious deviation from internal logical design of \textastext. I don’t know if it is because \TeX documentation is deficient or misleading on such matters but it appears many \TeX users are surprised when \textnormal does not give an upright pi letter but this is completely to be expected in a world with fonts having only 128 or 256 glyphs, and from the fact that \textrm and \textbf originate in Plain \TeX \rm and \bf and are still quite akin to it, they are font switching commands nothing more or less.

1.8 Extra spaces around letters

This is a new feature added with release 1.3: the command \textsetmathskips allows the user to set up some spaces (more precisely, ‘mu glue’; but stretch and shrink are discarded) to be automatically inserted around the letters in math mode. Some (very) unrealistic uses:

% this may be anywhere in the document (also within a math group):
\textsetmathskips{x}{20.33mu}{15.66mu}% 20.33mu before all x's and 15.66mu after.
\textsetmathskips{y}{\thickmuskip}{\thickmuskip}%
\textsetmathskips{z}{10mu}{5mu}% stretch and shrink are anyhow without effect.
\textsetmathskips{A}{\muexpr \thickmuskip*2}{\muexpr \medmuskip\thinmuskip/2}%
Here is what $\text{wxtyz}^{-\{\text{wxtyz}\}}=\text{BAC}^{-\{\text{BAC}\}}$ then gives using the Times font: $w \times t y t z \times w \times t y t z = B \ A \ C \ B \ A \ C$. Any TeX group or ltx group environment limits as usual the scope of this command. Furthermore the command \texttt{MTunsetmathskips} cancels previous use of \texttt{MTsetmathskips} for a given letter.

The implementation relies on the ‘mathematical activation’ of letters, which is done by default by the package since release 1.2b. Should this cause compatibility problems, the command \texttt{MTmathstandardletters} cancels it entirely. To reactivate it, there is \texttt{MTmathactiveletters}. Note that \texttt{MTmathactiveletters} is done automatically by \texttt{mathastext} when loaded, and also each time the package enhanced math-version-switch command \texttt{MTversion} is used, except for the normal and bold math versions under the \texttt{subdued} option.

The extra skips are set at natural width; they do not contribute to the overall stretchability or shrinkability of the math formula and do not create break points. 

**Changed with 1.3i:** they are not applied within the scope of math alphabet commands.

### 1.9 Italic corrections

Note: this is somewhat technical discussion which may well be skipped in its entirety on first reading.

With the \texttt{italic} option the letters in math will be generally in italic shape (and, normally, upright in operator names).

For the built-in placement routines of TeX in math mode to work as well as they usually do, the characters from the math italic font obviously should have their bounding boxes wide enough for the glyphs not to collide with other symbols. A letter from a text italic font such as \texttt{f} extends way out of its declared bounding box; let us compare the bounding boxes\footnote{let’s be honest, we are lying here about what exactly the first of these is bounding; this is explained later!} for the letter \texttt{f} in the math italic font to the one from the text italic font: $f$ vs. $f$.

This could make us think that attempting to use in math a text italic font will lead to disaster. Well, surprisingly the situation is not that bad. Sure $\texttt{f(x)}$ is wider with the standard math italic $f(x)$ (21.31474pt) than it is with the text italic font used in math: \footnote{we used simply $\texttt{\mathit{f(x)}}$}. $f(x)$ (19.74986pt) but we should be surprised that our text italic \texttt{f} did not end up even closer to the opening parenthesis. Why is it so?

The explanation is that TeX uses in such a situation the \textit{italic correction} for the letter \texttt{f}. The italic correction also exists and is used for the math italic font, it was inserted in $\texttt{f}$ without us having to ask anything. Its value is 1.17865pt for the math italic \texttt{f} and 1.8919pt for the text italic \texttt{f}.ootnote{these values are for the Latin Modern fonts of course.}

\footnote{we used simply $\texttt{\mathit{f(x)}}$.}
included our bounding boxes are indeed more alike: $\text{f} vs \text{f}$. 

Without the italic corrections$^{46}$ it is $\text{f} vs \text{f}$. I said that $\text{f}^\text{x}$ included the italic correction automatically, but if we tell \TeX to use the text italic in math, and typeset the alphabet, we obtain something exactly identical to typing the letters in text, hence without any italic correction:

\begin{align*}
abcdefghijklmnopqrstuvwxyz & \quad \text{text italic in text} \\
abcdefghijklmnopqrstuvwxyz & \quad \text{text italic in math} \\
abcdefghijklmnopqrstuvwxyz & \quad \text{math italic in math} \\
abcdefghijklmnopqrstuvwxyz & \quad \text{math italic in text}
\end{align*}

Where are our italic corrections gone? the last line was done with \texttt{\usefont{OML}{mlmm}{m}{it}} and the line before that using math mode is longer and confirms that italic corrections have been used for the math italic in math mode.

Turning to the \TeXbook (and its Appendix G) we learn that in such circumstances, for the italic corrections to be put in from the font, one of its parameters, the interword space (aka 2\texttt{\fontdimen2}), should be zero. It is indeed zero for the math italic font, not for the text italic.

It is possible to make \TeX believe it is. Doing so, we obtain in math mode with the text italic:

\begin{align*}
abcdefghijklmnopqrstuvwxyz & \quad \text{text italic in math} \\
abcdefghijklmnopqrstuvwxyz & \quad \text{math italic in math}
\end{align*}

We saw that the italic correction was taken into account automatically (independently of the value of the interword space font parameter) in expressions such as $\text{f}(x)$. Another clever thing done by \TeX is to use it for the placement of superscripts; the next examples systematically use the text italic in math. We see that $\text{f}^\text{j}$ is very different from $\text{f}^\text{j} ...$ where the latter was coded with $\texttt{\hbox{\itshape f}^\text{j}}$. The inputs $\texttt{\mathit{\hbox{\itshape f}^\text{j}}}$ and $\texttt{\mathit{f^\text{j}}}$ give almost identical results: $\text{f} vs. \text{f}$. Close examination reveals that the horizontal spacing is exactly identical, however the exponent in the second case is a bit lower. Anyway, the point is that in the second case the italic correction for $\text{f}$ was indeed used.

Subscripts are another matter: they do not take into account the italic correction. For example $\texttt{\mathit{f_i}}$ gives the same horizontal positions as $\texttt{\mathit{\hbox{\itshape f}_i}}$: $f_i$ vs. $f_i$. Printing them one on another gives $f_i$ and reveals (use the zoom of your viewer!) that only the vertical placement was affected, not the horizontal placement.

We learn in Appendix G of the \TeXbook that the italic correction is used for the horizontal shift of the superscript with respect to the position of the subscript: $f_{i}^{j}$, or, going back now to the standard math italics $f_{i}^{j}$. In the next paragraphs we use $f_{i}^{j}$ for more accurate comparison of the positioning of the sub- and superscript.

If we try something like this: $\texttt{\{f\}/i^{i}}$ we obtain $f_{i}^{i}$. Our overlapping game with $\texttt{\rlap{\{f\}/i^{i}}}\{f\}/i^{i}$ gives $f_{i}^{i}$. We discover that the effect of the explicit italic correction has mainly been to translate the subscript horizontally to

\footnote{here we give correctly the bounding box for the math italic $f$... without its italic correction!}
be positioned exactly below the superscript! We most probably do not want this to happen for our indices and exponents in math mode. So perhaps we can rejoice in how astute \TeX{} has been in judiciously using the italic correction data, and there seems to be no need into fiddling with this algorithm which seems to work well even when applied to a text italic font. Actually we may even be of the opinion that the text italic version \( f_i^i \) is a bit better-looking than the true math italic \( f_i^i \).

But wait... \texttt{mathastext} was initially developed to easily use in math mode the document text font not in its italic variant, but as is, so, usually, upright. And upright \TeX{} fonts may also have italic correction data! And what I just said about the shift of the superscript with respect to the subscript apply equally well to such a font, if \TeX{} has been told to use it. Let’s try Latin Modern Upright for letters in math: \$f_i^i\$ now gives\(^48\) \( f_i^i \). We see the italic correction in action for the positioning of the superscript! Compare with \$\text{\texttt{mathrm(\hbox{f}_i^i)}\}$: \( f_i^i \). Overlapping with \( \text{\texttt{rlap(\texttt{mathrm(f_i^i)}\)}} \text{\texttt{mathrm(\hbox{f}_i^i)}} \$ gives \( f_i^i \) and shows that the upright \( f \) has an italic correction which was used to shift the superscript to the right (and it is now in a slightly lower position). Let’s now do \$\text{\texttt{mathrm({f/_i^i})\}}\$: this gives \( f_i^i \) and the subscript is shifted to the right, and is now on the same vertical axis as the superscript. There are also some slight vertical displacements, \( \text{\texttt{rlap(\texttt{mathrm(f_i^i)}\)}} \text{\texttt{mathrm({f/_i^i})\}} \$ gives \( f_i^i \).

People will tell me crazy, but if we decide for using upright fonts in math, wouldn’t it be satisfying to have the subscript and superscript positioned on the same vertical axis? the letter has no slant, why should the indices display one?

We end up in this strange situation that it is attractive to systematically incorporate the italic corrections after the upright Latin letters in math! But we don’t want to do this inside the arguments to math alphabets as this would make impossible the formation of ligatures (the standard \$\text{\texttt{mathrm(ff)}\}$, \$\text{\texttt{mathit(ff)}\}$, \$\text{\texttt{mathbf{ff}}\}$, \$\text{\texttt{mathsf{ff}}\}$ all give ligatures ff, \( \text{\texttt{ff}} \), \( \text{\texttt{ff}} \), and \( \text{\texttt{ff}} \) and we would like to preserve this behavior).

Starting with version v1.2b, \texttt{mathastext} adds the italic correction automatically after each letter of the Latin alphabet in math mode, except when these letters are italic or slanted.\(^49\)

These italic corrections are canceled inside the arguments to the math alphabet commands, to allow the formation of ligatures as is expected in the standard default \TeX{} font set-up in math.

\(^{47}\)there are also some tiny vertical displacements of the sub- and superscripts.

\(^{48}\)we just use \$\texttt{mathrm(f_i^i)}\$.

\(^{49}\)the situation is rather ironical! by the way, the warnings in section 1.11 with \$x^{-?}\$ or similar are less of an issue here, because the letter is only followed by \( / \) and anyhow the whole is put by \texttt{mathastext} within group braces, so no surprises with \$x^{-y}\$ or \$\texttt{mathbin x}\$. Nevertheless it is still true that (in math mode only) the letters a-z, A-Z, expand to composite objects, something which could
The feature-implementing commands `\MTicinmath`, `\MTnoicinmath`, `\MTicalsoinmathxx` are described in section 2.2.1.

Note: from brief testing on 2012/12/28, \TeX{} seems not to obey in math mode italic corrections for OpenType fonts. Hence the \TeX{} placement algorithms for math mode described in this section do not work well when an OpenType (text) font is used for the letters in math mode, and the document is compiled with the \TeX{} engine. On the other hand Lua\TeX{} seems to implement the italic corrections when using OpenType fonts, but only with italic fonts (as far as I could tell). Try the following (which will use the OpenType Latin Modern font) on a recent \TeX{} installation and compare the output of both engines:

```latex
\documentclass{article}
\usepackage{fontspec}
\begin{document}
\Huge
$\mathit{f_i^i}$
$\mathrm{f_i^i}$
\end{document}
```

Comment out the `fontspec` line and use pdf\TeX{}. All three outputs are different on my \TeX{} installation. \TeX{} does not have the italic corrections. Lua\TeX{} does, but only for the italic font. pdf\TeX{} has them for both the italic and the upright font.\footnote{2022/10/29: no change with current TL2022.}

\section*{1.10 Extra glue after `\exists`, `\forall`, and before the prime glyph}

`\MTforallskip`, `\MTexistsskip`, and `\MTprimeskip` are three commands with each a mandatory argument like for example `3\mu` plus `1\mu` minus `1\mu` or just `2.5\mu`. They are especially useful when using an upright font in math mode. The \mu{} is a unit length used in math mode (‘math unit’, 1/18th of the ‘quad’ value of the symbol font in the current style). Its value is relative to the current math style. Its use is mandatory in the commands described here.

- compare \forall{} B with \forall{} B, typeset after `\MTforallskip{2\mu}`,
- compare \exists{} N with \exists{} N, typeset after `\MTexistsskip{2\mu}`,
- and finally compare f’ with f’, typeset after `\MTprimeskip{2\mu}`.

These three commands may be used throughout the document, or also in the preamble, in which case the declared math versions will record the then current values of the skips. `mathastext` applies the following (small) default skips: 0.6667\mu{} for the skip after `\forall`, 1\mu{} for the skip after `\exists`, and 0.5\mu{} for the skip before the prime. The examples above become \forall{} B, \exists{} N and f’.\footnote{2016/11/04: the situation hasn’t changed, at least on current TL2016.}

\footnote{The derivative glyph from the `txfonts` math symbols adapts itself better to an upright letter, no skip seems to be needed then.}

---

\footnote{\texttt{MTmathstandardletters} cancels this mechanism.}
With the \textit{italic} option the defaults are set to zero. Indeed $\forall B$, $\exists N$ and $f'$ look fine without additional skips. If the document decides then to declare in the preamble a math version with an upright font it is thus recommended to use the commands in the preamble before the \texttt{\textbackslash Mathastext[(version_name)]} (or \texttt{\textbackslash MTDclareVersion}) command defining the version. They will be remembered when this math version is entered in the document. The commands may also be used directly in the document body.

Under the \texttt{subdued} option, the \texttt{normal} math version (at the start of the document body, or after \texttt{\textbackslash MTversion(normal)}) and the \texttt{bold} math version (either at the start of the document body after \texttt{\textbackslash boldmath}, or after \texttt{\textbackslash MTversion(bold)}) do not have any extra skip inserted (even one of zero width) after $\forall$, $\exists$, or before the $'$. 

### 1.11 Extended scope of the math alphabets commands

Ever since the initial version of the package, some characters usually unaffected by the math alphabet commands $\texttt{\textbackslash mathbf}$, $\texttt{\textbackslash mathtt}$, $\texttt{\textbackslash mathsf}$... are declared to be of ‘variable family type’, in order for them to obey these commands: for example the hash sign $#$ gives $#$ if input as $\$\textbackslash mathbf\{\#\}\$ (\texttt{mathastext}, especially in its beginnings, wanted as many characters as possible to be picked up from the text font and to behave similarly to letters and digits).

So it was especially frustrating that mathematical characters such as $+$, or $<$, or $|$ could not be declared of ‘variable family’ (in addition to being picked up in the text font) as this would, for reasons of the inner workings of T\hbox{\TeX}, not be compatible with the automatically inserted spaces around them.

A revolutionary ;-) novelty is introduced with version 1.2 of the package:

1. the pre-declared or user-declared (using the \texttt{amsmath \textbackslash DeclareMathOperator} or equivalent) operator names obey the math alphabet commands, $^{52}$

2. and, \texttt{optionally}, all non alphabetical characters$^{53}$ treated by \texttt{mathastext}, i.e., if not disabled by options, $!$,$?;:+=-()[]<>\{}$, the asterisk $*$, and $./|\#\%\&$ $^{54}$ will also obey the math alphabet commands (when not used as delimiters). The important thing is that the spaces added by T\hbox{\TeX} before and after are not modified.

$^{52}$contrarily to the next feature, this one is not likely to create incompatibilities with other packages, so it is activated by default.

$^{53}$of course some of them are input preceded by a backslash, and the backslash itself is input as $\textbackslash$backslash.

$^{54}$\# $\%$ & obey the math alphabets since the initial version of \texttt{mathastext}; the dot $.$, the slash $/$, the vertical bar $|$ and the backslash $\backslash$ do not have specific spacings inserted by T\hbox{\TeX} around them, and the procedure is then activated by default since 1.2 for these characters as they are ‘easy non-letters’. But for $\mid$ and $\setminus$ which are $|$ and $\backslash$ with special spacing (of type $\textbackslash mathrel$ and $\textbackslash mathbin$ resp.) the procedure has some constraints explained in the framed box on next page and the activation requires $\textbackslash MTnonlettersobeymathxx$.  

34
Let us compare, for example, the new behavior of `\mathtt` and `\mathbf` with the traditional default behavior:

\[
\sin(n!) < \cos(m - p) \quad [\sin(x + y) = \cos(z - t)]
\]

The first feature is activated by default, except of course for the normal and bold math versions when the package was given the *subdued* option. The second feature is *off* by default for the characters listed first. It is *on* for the ‘easy’ cases `#$\%&./|\` (activating the feature for them puts no constraint on the user input and should not be too upsetting to other packages), and also for `*` but only if this was required explicitly by the option `asterisk`, as the user then is supposed to know that `$R^*` is no valid input anymore and should be replaced by `$R^{*}`.

The remaining ‘difficult’ cases create similar constraints, which will be commented more upon next. The commands\(^{55}\) for deactivation are:

- `\MTmathoperatorsdonotobeymathxx`,
- `\MTeasynonlettersdonotobeymathxx`,
- `\MTnonlettersdonotobeymathxx`,

and those for activation:

- `\MTmathoperatorsobeymathxx` regards operator names and is executed by default,
- `\MTeasynonlettersobeymathxx` is done by default and applies to `#$\%&./|\`

and also to `*` under package option `asterisk`,

- `\MTnonlettersobeymathxx` is *not* done by default (see explanations why in the framed box next) and regards `!?;+--()[]<>`

and also `\mid` and `\setminus` but applies to the braces `{ }` only if `\MTexplicitbracesobeymathxx` is also used.

---

**Important:** the package does `\MTnonlettersdonotobeymathxx` by default. The reason is that activating the mechanism adds some constraints to the way things must be input, adding

\[
\texttt{\usepackage{mathastext}\MTnonlettersdonotobeymathxx}
\]

to a pre-existing document might well create errors: all these characters treated by `mathastext`, such as `?, [`, `< now represent (in math mode only!) *two* ‘tokens’ and this will utterly confuse `\LaTeX` if some precautions are not taken: `$x^?`,

---

\(^{55}\)these commands are to be used outside of math mode. Their scope is limited to the current `\LaTeX` environment or group. They use the `\everymath` and `\everydisplay` mechanism so if the document needs to modify these token lists it has to do so in a responsible manner, extending not annihilating their previous contents.
must now be coded as $x^{?}$, $R^{+}$ and $\mathopen{<}A\mathclose{>}$ (the rule is to do as if ?, +, < or > were each really two characters).

Even if this rule is respected in the document source, it is still a possibility that incompatibilities with other packages will arise because \texttt{mathastext} does a \textit{mathematical activation} of the characters which could be unexpected and unchecked for by other packages. This is precisely the case with the \texttt{amsmath} package, and the problem goes away by just making sure that \texttt{amsmath} is loaded before \texttt{mathastext} (generally speaking, \texttt{mathastext} should be loaded last after all packages dealing with math things).

The braces \{ and \} remain unresponsive to the alphabet changing commands even after \texttt{\MTnonlettersobeymathxx}. One must issue also \texttt{\MTexplicitbrace-sobeymathxx}, but it has the disadvantage that \{ and \} become then unusable as variable-size delimiters: \texttt{\big\{ or \big\} create errors and one must make use of \texttt{\big\lbrace and \big\rbrace}. But one can now enjoy \{a, a > b\}, \{a, a \succ b\}, \{a, a \succ b\}, or even \{a, a \succ b\}.

Even with \texttt{\MTnonlettersobeymathxx}, the parenthese-like symbols (, ), [ ] and < and > do not react to math alphabet commands. This is mainly explained by the fact that the text font will not contain suitable glyphs, hence no attempt was made to make the delimiters pick up their glyphs there.

But \texttt{mathastext} does try to pick up most of the ‘small variants’ of the delimiters from the text font: $\left< x \right>$ gives < x > (but $\left< b \right>$ gives < b >. Notice that this differs from standard \LaTeX{} for which $\left< x \right>$ gives < x >. As it is perhaps a bit strange to have < x > next to < X > there is option \texttt{nosmalldelims}: with this option the small-sized variants of the delimiters are not modified by \texttt{mathastext} (option \texttt{nosmalldelims} has the side effect that, for the non-delimiter uses of \{, \} to be \texttt{mathastext}-ified it is necessary to issue \texttt{\MTnonlettersobeymathxx and \MTexplicitbrace-sobeymathxx}.)

At any rate, as said above, whether ‘small’ or not, delimiters are unresponsive to math alphabet commands, due to technical aspects of \TeX{}, and the way \texttt{mathastext} handles these things. Examples: \texttt{\mathbf{\{a,b\}}} gives < a, b > (no use of \texttt{\left/\right}, hence brackets do obey the math alphabets — as we issued \texttt{\MTnonlettersobeymathxx} a bit earlier), \texttt{\mathbf{\{left\langle a,b\rangle\}}} gives < a, b > (brackets used with \texttt{\left/\right} do not obey the math alphabets), \texttt{\mathbf{\{mathopen{\langle}a,b \mathclose{\rangle}\}}} gives < a, b > (no \texttt{\left/\right}, brackets do obey the math alphabets).

For comparison, the \LaTeX{} standard behavior for

\footnotesize
\begin{itemize}
  \item This last example uses the \texttt{\mathnormalbold} additional alphabet defined by \texttt{mathastext}.
  \item Let me recall that braces will anyhow not be handled at all by \texttt{mathastext} if the document font encoding is OT1, except under option \texttt{alldelims}.
\end{itemize}
\textbf{\textopen{<}a,b\textclose{>}} is \textless a, b \textgreater{} (neither brackets nor the comma do respond).

### 1.12 Unicode engines

The package \texttt{mathastext} is minimally Unicode aware since 1.12 and can be used with \TeX\ or \LaTeX. Starting with release 1.3, it needs \texttt{luatex} to be at least as recent as the one which was provided with the TL2013 distribution.

However \texttt{mathastext} applies only to (a subset of) the 32–127 ascii range, and optionally to Greek letters, but for the latter only if provided via \TeX\ fonts such as Euler, Symbol or LGR-encoded fonts. It does not know how to use a given Unicode font simultaneously for Latin and Greek letters.

Thus, first consider much better alternatives:

- Since 2018, the package \texttt{mathfont} adapts Unicode text fonts to usage in math mode. It works with both \TeX\ and \LaTeX.

- For \TeX\ only, \texttt{mathspec} also allows usage of arbitrary text fonts in mathematics.

- and of course \texttt{unicode-math} is the standard package for using OpenType fonts which are equipped with the needed extra support being used in \TeX\ math mode.

If using any one of the above you probably don’t need, don’t want, and should not use \texttt{mathastext}.

Let me insist that \texttt{mathastext} has not been tested in any systematic manner under the Unicode engines; and that it is expected to be most definitely incompatible with \texttt{unicode-math}, although your mileage may vary and some features may appear to work.

When using \texttt{mathastext} with either \TeX\ or \LaTeX\ it is recommended to use the \texttt{fontspec} package (see remark below on \texttt{encodingdefault}). Furthermore, if using \texttt{fontspec} it is \textit{necessary} to load it with its \texttt{no-math} option, and this \textit{must} happen before loading \texttt{mathastext}.

- Use \texttt{fontspec} with its \texttt{no-math} option, and load it \textit{prior} to \texttt{mathastext}. As some packages load \texttt{fontspec} themselves (for example \texttt{polyglossia}), a

---

58 Conrad Kosowsky, \textit{Use TrueType and OpenType fonts in math mode} \url{https://ctan.org/pkg/mathfont}.

59 Andrew Gilbert Moschou, \textit{Specify arbitrary fonts for mathematics in Xe\TeX} \url{https://ctan.org/pkg/mathspec}.

60 Will Robertson, et al., \textit{Unicode mathematics with support for Xe\TeX\ and Lua\TeX} \url{https://ctan.org/pkg/unicode-math}.
The `amsmath` package, if used, must be loaded prior to `mathastext`.

- Under `lualatex` engine, it is recommended to also load the package `lualatex-math`.

I already mentioned in the section 1.9 the fact that the italic corrections were not available for OpenType fonts under the X\TeX engine and only partially available for the Lua\TeX engine, with the result that the spacings in math mode when using for the letters an upright text font will be less satisfying than with the standard PDF\TeX engine (the OpenType fonts not being usable with the latter engine, this is not a criterion of choice anyhow).

To define math versions when using unicode fonts, use `fontspec's \setmainfont` before the `\Mathastext\[version\]` command, or simply before loading `mathastext` for the default math versions.

It is possible to mix usage of Unicode fonts and classical \TeX fonts. All used 8bits font encoding must have been passed as options to the `fontenc` package.

### 1.12.1 The unicodeminus option

For legacy reason, `mathastext` uses by default the EN DASH U+2013 for the minus sign in math mode, if the font is determined to be a “Unicode” font.

There is now the `unicodeminus` to use rather MINUS SIGN U+2212.\footnote{Thanks to Tobias Brink who asked for this feature.} Check its \footnote{A `tex mathastext.dtx` (in a temporary repertory) on a copy of `kpsewhich mathastext.dtx` will extract extended versions of these examples as test files.} documentation on page 58.

### 1.12.2 Two examples

I include here two examples which compiled successfully with X\TeX and Lua\TeX, the first one on a Linux machine, the second one on a Mac OS X machine.\footnote{A `tex mathastext.dtx` (in a temporary repertory) on a copy of `kpsewhich mathastext.dtx` will extract extended versions of these examples as test files.}

```latex
\documentclass{article}
\usepackage[hscale=0.8]{geometry}
\usepackage{multicol}
\usepackage[no-math]{fontspec}
\usepackage{lmodern}
\usepackage[subdued,italic]{mathastext}
\setmainfont[Color=999999]{Verdana} \Mathastext[Verdana]
\setmainfont[Color=0000FF]{Arial} \Mathastext[Arial]
\setmainfont[Color=00FF00]{DejaVu Serif} \Mathastext[DejaVu]
```
1.13 Compatibility issues

Compatibility issues (or just questions of who decides last) are naturally to be expected with packages dealing with the math setting; the fix is simply to load
mathastext last. And one should always load amsmath before mathastext (this is especially true when using Unicode engines but applies in general as well).

Any definition made in a package loaded before mathastext of the font to be used for letters or for the common characters in the ascii basic range will be overruled by the loading of mathastext (this includes the case when the earlier package had made the character ‘mathematically active’). Conversely most of the set-up done by mathastext may well be overruled by packages loaded later which do math related things.

Starting with version 1.2, mathastext makes some characters ‘mathematically active’ to achieve certain effects: automatic insertion of the italic corrections when using an upright text font in math, extended scope of the math alphabet commands which now apply to non-letter symbols (and also to math operator names, but this is much easier to achieve). And the (already mathematically active) right quote is modified to have some extra space added before the derivative glyph ‘.

This is compatible with using \label and \ref in and outside of math mode. But a difficulty arises when some other package has made the character ‘globally active’ everywhere in the document. The action of mathastext is made anew at each mathematical inline or displayed formula. If it is detected that a character has been activated then nothing further will be done (so the mathastext feature\footnote{italic correction insertion for the latin letters, receptivity to the math alphabet action for the other characters.} for that character is lost) except if it appears that this activation was done by the Babel system. In that case mathastext does not make the character mathematically active but it modifies in the appropriate manner the action of Babel for that character in math mode. Furthermore mathastext makes the character mathematically \textit{inactive}.

Here is indeed some code that you should not try at home:

\begin{verbatim}
\documentclass{article}
\usepackage[french]{babel}
\usepackage{mathtools}\mathtoolsset{centercolon}
\begin{document}
$:$
\end{document}
\end{verbatim}

\textbf{DO NOT DO THIS AT HOME}: it creates an infinite loop.\footnote{This seems to still be the case with Babel 3.9f and frenchb.ldf 2.6e, as tested on Sep. 2, 2013. Again tested with up-to-date TL2015 Jan. 15, 2016 with same result.} This is due to the fact that the colon is simultaneously active (this is made by babel-french at begin document) and mathematically active (done by mathtools in the preamble). The interaction gives an infinite loop. Such a situation will be cured by mathastext, even loaded before mathtools, if use is made of MTnonlettersobeymathxx. At
each math formula \texttt{mathastext} will detect that Babel has activated the colon, and
will cancel the mathematical activation (the precise definition done by \texttt{mathtools}
was already lost at begin document due to overwriting by \texttt{babel} but the fact that
the character was mathematically active remained true).

So far I have briefly described the problem of document active characters (see the
test file \texttt{mathastexttestalphabets.tex} for more explanations and illustrations,
and the commented source code of the package). Pure mathematical activation
revealed an incompatibility of another type with \texttt{amsmath}. To fix it, \texttt{mathastext}
now replaces an inner macro of \texttt{amsmath} (\texttt{\resetMathstrut@}) with its own version.

\begin{itemize}
\item Always load \texttt{amsmath} before \texttt{mathastext}.
\end{itemize}

Actually this last commandment was already made necessary by the use of the
text endash to represent the minus sign in math mode, and, especially for Unicode
engines, some aspects of the \texttt{\DeclareMathOperator} macro from \texttt{amsmath}.

\begin{itemize}
   \item Important! As is mentioned in the section 1.11, after command \texttt{\MTnon-
   lettersobeymathxx}, characters such as ?, or [, now represent two ‘tokens’
and this will utterly confuse TeX if some precautions are not taken. Examples: $0^{+}$ or $x\mathrel{?}y$ or $R^{*}$ must be input now as $0^{+}$
and, respectively, $x\mathrel{?}y$ or $R^{*}$. This is why the package
does \texttt{\MTnonlettersdonotobeymathxx} by default.
\end{itemize}

One thing to take note of is that this mechanism uses the \texttt{everymath} and \texttt{everydisplay}, so if it is needed to add to these TeX ‘token lists’ some additional
things this should be done in a way preserving the former contents.

If one issues (after \texttt{\begin{document}}) \texttt{\everymath={}} and \texttt{\everydisplay={}}
this annihilates not only all the \texttt{mathastext} (evil ?) doings with math active characters
but also everything else some other package might have put in these token
registers, so it is better, if the need arises to cancel the math activation of characters
done by \texttt{mathastext} to use the command \texttt{\MTeverymathoff}, which does all of
\texttt{\MTmathoperatorsdonotobeymathxx}, \texttt{\MTnonlettersdonotobeymathxx} (already default), \texttt{\MTmathstandardletters}, \texttt{\MTnormalprime}, and \texttt{\MTnormalasterisk}.
This is supposed to be used in a group or environment (as there is no \texttt{\MTactivemathom}.
It must be used prior to entering math mode.

\begin{itemize}
   \item New with 1.3i: \texttt{mathastext} patches \texttt{url} of packages \texttt{url} and \texttt{hyperref}, and
also \texttt{\nolinkurl}, to force them to do automatically \texttt{\MTeverymathoff}. Indeed
they use math mode, and it is better to turn \texttt{mathastext} off for their dealings.
\end{itemize}
2 Package commands

2.1 Commands for regular usage

2.1.1 Preamble-only commands

These commands mainly facilitate the definition of math versions, in a \texttt{mathastext}
extended sense. It is not necessary to use them to activate the package basic
functionalities, as loading \texttt{mathastext} is enough (except with the \texttt{subdued} option).

- \texttt{\Mathastext} (or \texttt{\mathastext}) reinitializes \texttt{mathastext}: it sets the fonts used
in math mode (in versions \texttt{normal} and \texttt{bold}) for letters, digits and a few ascii
symbols to the \texttt{current} defaults of encoding, family, series and shape.\textsuperscript{66} Both the
normal and bold math version are modified by this action of \texttt{\Mathastext}.

- \texttt{\Mathastext[\textit{version\_name}\texttt{\textbackslash}]} rather than redefining the fonts for math mode,
\texttt{\Mathastext} declares a new math version, and it is this math version which will
use the then current text font in math mode.\textsuperscript{67}

- \texttt{\Mathastext[\textit{version\_name}\texttt{\textbackslash} \textit{parent\_name}\texttt{\textbackslash}]} declares \textit{version\_name}\texttt{\textbackslash} and
configures it to inherit from \textit{parent\_name}\texttt{\textbackslash} all which is not under the scope
of \texttt{mathastext}, such as large symbols. The main use will be with \texttt{\textbackslash{bold}} in or-
der for the symbols and large symbols to be typeset as in the \texttt{bold} math version.
For example, this document has in its preamble:
\begin{verbatim}
\usepackage{newcent}\% this package makes New Century the roman font
\Mathastext[newcent]\% this math version will use New Century
\MTseries{b} \% next \Mathastext will use a bold font
\Mathastext[boldnewcent][bold]\% large symbols, etc, will be bold too
\end{verbatim}
We can check that it does work:

\begin{verbatim}
\MTversion{newcent}: abcd e \begin{array}{cccc}
\boxplus & \boxtimes & \boxdot & \bigcirc \\
\boxplus & \boxtimes & \boxdot & \bigcirc \\
\end{array}
\MTversion{boldnewcent}: abcd e \begin{array}{cccc}
\boxplus & \boxtimes & \boxdot & \bigcirc \\
\boxplus & \boxtimes & \boxdot & \bigcirc \\
\end{array}
\end{verbatim}

Naturally, for this one needs an initial math font setup with some nice bold
fonts also for large symbols. This is the case with the excellent \texttt{txfonts} pack-
age of Young Ryu. As the present document must use many fonts and declares
many math alphabets, we did not load the full package and fonts but only the
large symbols:
\begin{verbatim}
\DeclareSymbolFont{largesymbols}{OMX}{txex}{m}{n}
\SetSymbolFont{largesymbols}{bold}{OMX}{txex}{bx}{n}
\DeclareFontSubstitution{OMX}{txex}{m}{n}
\end{verbatim}

\textsuperscript{66}\Mathastext updates also the font and shapes for the Greek letters (\texttt{LGRgreek} option), and the
skips to be inserted after the symbols \texttt{\forall} and \texttt{\exists}, see \textit{infra}.
\textsuperscript{67}The allowed version names are as for the \LaTeX\ \texttt{\DeclareMathVersion} macro. \textit{Do not use}
\texttt{\Mathastext[foo]} with \texttt{foo} equal to \texttt{“normal”} or \texttt{“bold”}; this is already taken care of by the
initial loading of the package or a later command \texttt{\Mathastext} without any optional argument.
And it will be rejected.
\textbf{MTencoding}\((\text{enc})\), \textbf{MTfamily}\((\text{fam})\), \textbf{MTseries}\((\text{ser})\), \textbf{MTshape}\((\text{sh})\), and \textbf{MTlettershape}\((\text{sh})\). For example valid respective arguments are, respectively, \langle T1 \rangle, \langle phv \rangle, \langle m \rangle, \langle n \rangle, and \langle it \rangle: this is the Helvetica font in T1-encoding, regular (medium) series, upright shape, and the letters will be in italic shape. Once used their effect applies to all succeeding calls to \texttt{Mathastext}, and can only be undone by using them again with other settings, again followed by a call to \texttt{Mathastext}.

\textbf{NOTE:} only if \texttt{Mathastext} is used next (possibly with a version name as optional argument) will these commands have any real effect.

\textbf{MTWillUse}\[ltsh\]{\langle enc\rangle}{\langle fam\rangle}{\langle ser\rangle}{\langle sh\rangle}\] tells \texttt{mathastext} to use the font with the specified encoding, family, series, and shape for the letters and digits (and all other afflicted characters) in math mode. The optional argument \langle ltsh \rangle specifies a shape for the letters, for example \langle \texttt{itdefault} \rangle, or directly \langle it \rangle or \langle sc \rangle.

\textbf{MTDeclareVersion}\[ltsh\]{\langle name\rangle}{\langle enc\rangle}{\langle fam\rangle}{\langle ser\rangle}{\langle sh\rangle}\[other\_version\]: declares that the document will have access to the font with the specified characteristics, under the math version name \langle name \rangle. For example:

\texttt{MTDeclareVersion[sc]{palatino}{T1}{ppl}{b}{sl}}
deares under the name palatino a version where mathematics will be typeset using the Palatino font in T1-encoding, bold, slanted, and the letters will in fact be in caps and small caps (and bold). When the initial optional argument is absent, and \texttt{mathastext} was loaded with the \textit{italic} option, then the default letter shape will be \texttt{it}, else letters will have the same shape as used for digits and operator-names.

Another optional argument may be used as last argument. Similarly as its use with \texttt{Mathastext} this makes the declared math version inherit, for things not modified by \texttt{mathastext} like large symbols, the font set up of the math version whose name was passed as optional argument (typical use will be with \texttt{[bold]}).

\textbf{MTboldvariant}\((\text{var})\): when used before \texttt{Mathastext}, specifies which bold (b, sb, bx, ...) to be used by \texttt{mathbf} (and \texttt{boldmath}). Default is the \texttt{\bfdefault} at the time of loading \texttt{mathastext}. When used before the declaration of a version, decides the way \texttt{mathbf} will act in this version.

\textbf{MTEulerScale}\((\text{factor})\): scales the Euler font by \langle factor \rangle.

\textbf{MTSymbolScale}\((\text{factor})\): scales the Symbol font by \langle factor \rangle.

\textbf{MTitgreek, MTupgreek, MTitGreek, MTupGreek}: these commands are active in case the \texttt{LGRgreek} option was used; they act as the options of the similar

\begin{itemize}
\item These commands exist also with long names: \texttt{Mathastextencoding}, etc... The same applies to the other commands mentioned in this section.
\item I do not especially recommend to use this in real life!
\item more precisely, the shape is the latest value passed in one of the previously used package commands to specify the shape of letters, or the \texttt{\itdefault} of the time of loading the package.
\end{itemize}
names \textit{itgreek}, \textit{upgreek}, \textit{itGreek}, \textit{upGreek}, but only for the Greek letters in the versions yet to be defined. Their effect become recorded only when the version is declared via \texttt{\textbackslash Mathastext} or \texttt{\MTDeclareVersion}.

- \texttt{\MTgreekfont}\{\texttt{fontfamily}\}: a command with a mandatory argument which specifies the font family for Greek letters in all \texttt{mathastext} math versions declared afterwards via \texttt{\textbackslash Mathastext} or \texttt{\MTDeclareVersion}. Only effective if \texttt{LGRgreek} (or \texttt{LGRgreek*}) or \texttt{selfGreek} option was passed to the package.

Check the \texttt{LGRgreek} documentation for some relevant information.

### 2.1.2 Commands for body or math

- \texttt{\MTversion}\{\texttt{nametext}\}\{\texttt{namemath}\}, \texttt{\MTversion^\{\texttt{namemath}\}} (also known as \texttt{\textbackslash Mathastextversion} (and as \texttt{\MTVersion}, and \texttt{\mathastextversion}): (1.3c)
  - the non-starred version changes both the document text fonts and the math fonts (for those characters treated by \texttt{mathastext}): the mandatory argument is the math version to be used for math; the optional argument is the name of (another) \texttt{mathastext}-declared math version, the font which was chosen during its declaration will be set as document text font (and \texttt{\familydefault} etc... also are redefined). In the absence of the optional argument, the mandatory one is used. The versions must be either \texttt{normal}, or \texttt{bold}, or previously declared ones via \texttt{\textbackslash Mathastext} or \texttt{\MTDeclareVersion}.
  - the starred variant does the math set-up, but changes nothing to the text fonts (see subsection 1.6 for a description of the math set-up, which summarizes what is done additionally to only using \LaTeXX’s \texttt{\mathversion}).

- \texttt{\MTversion}\{\texttt{nametext}\}\{\texttt{namemath}\} does \texttt{\textbackslash MTeverymathdefault} (except for \texttt{\MTversion\{normal\}} and \texttt{\MTversion\{bold\}} under package option \texttt{subdued}), which in particular activates the insertion of skips around characters specified by \texttt{\MTsetmathskips} and also, if the font used is not oblique the insertion of italic corrections (for better positioning of subscripts; see the discussion in subsection 1.9). Under the \texttt{frenchmath} option the package checks separately the letter shape for lowercase and uppercase.

- \texttt{\MTversion} also does \texttt{\MTexistsdoesskip}, \texttt{\MTforalldoesskip}, and also \texttt{\MT\primedoesskip}, \texttt{\MTmathoperatorsobeymathxx}, except under the \texttt{subdued} option for \texttt{normal} and \texttt{bold}, in which case it does the opposite actions. (1.3j)

- \texttt{\textbackslash hbar}: this macro is by default redefined (in a way compatible with the \textit{italic} option) combining the \texttt{h} letter and the \textbackslash accent from the \texttt{mathastext} font. Note that \texttt{\textbackslash mathrm\{\textbackslash hbar\}} and \texttt{\textbackslash mathbf\{\textbackslash hbar\}} will work and that \texttt{\textbackslash hbar} does scale in subscripts and exponents. Since 1.3u, this is a priori compatible with all 8bits
text font encodings supporting the \textasciitilde text accent in the \LaTeX{} way.\textsuperscript{71} \textsuperscript{72}

- \texttt{\textbackslash fouriervec}: this is a \texttt{\textbackslash vec} accent taken from the Fourier font; the \texttt{fourier} package need not be loaded. Active only if option \texttt{fouriervec}.

- \texttt{\textbackslash pmvec}: this provides a poor man \texttt{\textbackslash vec} accent command, for upright letters. It uses the right arrow. Does not change size in subscripts and exponents.

new description:

- \texttt{\textbackslash Mathnormal, \textbackslash Mathrm, \textbackslash Mathbf, \textbackslash Mathit, \textbackslash Mathsf, \textbackslash Mathtt}: they use the \texttt{mathastext}-ified fonts. By default, \texttt{\textbackslash mathnormal, \textbackslash mathrm, \textbackslash mathbf, \textbackslash mathit, \textbackslash mathsf, \textbackslash mathtt} are redefined to map to these new commands using the \texttt{mathastext} fonts. The option \texttt{defaultalphabets} tells to keep them with their original meanings. Alternatively the original commands can be saved under other names before loading \texttt{mathastext}: the underlying architecture is not deleted by the package, and aliases defined before loading \texttt{mathastext} will work as expected.

- \texttt{\textbackslash mathnormalbold}: a bold version of \texttt{\textbackslash mathnormal}, i.e. picks up the math alphabet used for ascii letters as mathematical variables, but in a bold weight. When the package typesets such letters in the same shape as for operator names (i.e. neither \texttt{italic} option nor the \texttt{\MTLettershape} command have been used) the output is as the one of \texttt{\textbackslash mathbf}.

This command is also made available under \texttt{subdued} option in the “normal” and “bold” math versions, as \LaTeX{} does not define it a priori, contrarily to \texttt{\textbackslash mathbf} and other math alphabet commands.

- \texttt{\textbackslash mathgreekup}: math alphabet, only available under \texttt{LGRgreek} (or \texttt{LGRgreeks}) option, which gives access to ‘upright’ Greek letters (picked up from a font available in \texttt{LGR}-encoding). Note that the package also defines \texttt{\textbackslash alphaup, \ldots, \piup, \ldots} mathematical character tokens, see \textsuperscript{subsubsection} 1.7.3. What “up” shape really means may be math version dependent. It is configurable in the preamble via re-defining \texttt{\MTgreekupdefault} and then declaring the math version via \texttt{\Mathastext} (with optional argument if for a math version other than the “normal” one), or \texttt{\MTDeclareVersion}. The font used is also math version dependent: it is the one which was similarly configured via usage of \texttt{\MTgreekfont} prior to the \texttt{\Mathastext} or \texttt{\MTDeclareVersion} step. In absence of any such configuration in the preamble, it will be (in all math versions) the family default at time of loading the package (which thus has then to be available in \texttt{LGR} encoding; it is not a problem if the family default has no \texttt{LGR} support as long as suitable usage of \texttt{\MTgreekfont} later on configures a suitable font).

Also \texttt{\textbackslash mathgreekupbold}.

These math alphabets are also available under \texttt{subdued} option in the “normal” and “bold” math versions, as \LaTeX{} does not (a priori) define analog ones, so \texttt{mathastext} has no reason not to

\textsuperscript{71} The horizontal skips for letter \texttt{h} from \texttt{\MTsetmathskips} are ignored for \texttt{\hbar}.

\textsuperscript{72} The \texttt{\hbar} redefinition is canceled in normal and bold math versions under the \texttt{subdued} option.
leave them live. Note though that $\pi$ will work only if the original $\pi$ is of “variable family type” which is not the case except if some math package handling Greek was used, but then why load \texttt{mathastext} with option LGRgreek? But you can use \texttt{\mathgreekup{p}} as the slot number of $p$ in the \LaTeX font for mathematical letters is the same as the slot number of $\pi$ in LGR encoding.

Or, use rather $\piup$ because it is not undefined by \texttt{mathastext} in subdued normal mode, as \LaTeX has no a priori definition for it. Or use (but why?) \texttt{\mathgreekup{p}it}.

The LGR font family used will be the latest one configured by \texttt{\MTgreekfont} usage followed by \texttt{\Mathastext} (without optional argument) in the preamble which is what is needed to modify the non-subdued aspects of subdued “normal” math; if no such configuration was done, the font family will be the family default found at time of loading the package.

Worse: $\Delta$ is per \LaTeX default of variable family type but its slot number in its assigned font is not at all the one of the LGR encoding, so \texttt{\mathgreekup{bold}{\Delta}} will give some unrelated glyph. This is because \texttt{mathastext} restores the pristine $\Delta$ in subdued normal mode to its original meaning. But it keeps its own defined $\Deltaup$ and $\Deltait$, so you can use \texttt{\mathgreekup{bold}{\Deltaup}} for example. Or \texttt{\mathgreekup{bold}{D}} as the mathematical letter $D$ slot number in \LaTeX is also the one of $\Delta$ in LGR encoding.

I am sorry for such lengthy explanations, but this is to comment on why \texttt{mathastext} keeps also in subdued normal math some of its Greek related functionality, if option LGRgreek was used. Most \texttt{mathastext} users will not use the subdued option anyhow.

• \texttt{\mathgreekit}: math alphabet, only available under LGRgreek (or LGRgrees) (1.3x) option, which gives access to ‘italic’ Greek letters (picked up from a font available in LGR-encoding). The actual shape is configurable via re-defining \texttt{\MTgreekit: default} and then redeclaring the math version via \texttt{\Mathastext} (with optional argument if for a math version other than the “normal” one), or \texttt{\MTDeclare- Version}.

Also \texttt{\mathgreekitbold}. (1.3za)

See the discussion of \texttt{\mathgreekup} for some \TeX hacker level information on what happens with subdued option in the “normal” (or “bold”) math version.

• \texttt{\inodot}, \texttt{\jnodot}: the corresponding glyphs in the \texttt{mathastext}-ified font for use in math mode. By default, \texttt{\imath} and \texttt{\jmath} are redefined to use them. Since 1.3t, these macros obey the subdued regime.

• \texttt{\MathEuler}, \texttt{\MathEulerBold}: math alphabets to access all the glyphs of the Euler font, if option eulergreek (or eulerdigits was passed to the package.

• \texttt{\MathPSymbol}: math alphabet to access the Symbol font.

• when one of the options symbolgreek, eulergreek, or selfGreek is passed to the package the capital Greek letters which look like their Latin counterparts acquire names: \texttt{\Digamma}, \texttt{\Alpha}, \texttt{\Beta}, \texttt{Epsilon}, \texttt{\Zeta}, \texttt{\Eta}, \texttt{\Iota}, \texttt{\Kappa}, \texttt{\Mu}, \texttt{\Nu}, \texttt{\omicron}, \texttt{\Rho}, \texttt{\Tau}, \texttt{\Chi} (no \texttt{\Digamma} for Symbol). Also an \texttt{\omicron} control sequence is provided.
• LGR Greek and ‘var’-letters: only the \varsigma is available in this encoding, so using for example \varphi will load the previous default math font. It might thus be suitable when recompiling already written \LaTeX sources to add to the preamble \let\varphi=\phi, \let\varepsilonpsilon=\epsilon, etc... in case only the ‘variant’ form of the letter was used in the documents.

• Miscellaneous mathematical symbols from the postscript Symbol font are made available (or replaced) by option symbolmisc. They are \prod \sum \implies \iff \shortiff \mapsto \longmapsto \aleph \infty \emptyset \surd \nabla \swi \angle \forall \exists \neg \clubsuit \diamondsuit \heartsuit \spadesuit \smallint \wedge \vee \cap \cup \bullet \div \otimes \oplus \pm \ast \times \propto \mid \leq \geq \approx \supset \subset \subseteq \in \sim \cong \perp \equiv \notin \langle \rangle. And a \DotTriangle is made available by option symbolre (which overwrites \Re and \Im: \Re, \Im). The \infty and \propto have these names to leave up to the user the choice to replace (or no) the original (larger) \infty and \propto.

Regarding the \prod and \sum commands: they will use the Symbol glyphs \prod \sum in inline math, and in display math the Computer Modern ones (or whatever is set up by other packages; here we have the symbols from txfonts):

\[
\prod \sum
\]

The package provides \prodpsy and \sumpsy: if one really wants in all situations the Symbol glyphs, one can do \let\prod=\prodpsy and \let\sum=\sumpsy. Also \MToriginalprod and \MToriginalsum will refer to the \prod and \sum before redefinition by the package: this is to allow constructs such as \mtoriginalprod or \mtoriginalsum, because they would not work with the \prod and \sum as re-defined by the package.

2.2 Commands for expert usage

A few preliminary comments, mainly destined to advanced users aware of some \TeX innards (more extensive explanations are to be found in the code comments).

The timing for actions of mathastext falls into three cases:

1. things done during the loading of the package, or delayed to \AtBeginDocument,

2. things done as the result of user commands, either in the preamble or in the body of the document,

\footnote{Option asterisk is also required to treat the *. Recall from subsection 1.11 that the asterisk in math mode (also when using the control sequence \ast) appears then to \TeX to be a composite object.}
3. things done everytime math mode is entered.

The second category overlaps with the others, as the (preamble) use of some commands can have either immediate effect or only trigger some actions in \AtBeginDocument or perhaps only influence the things done later by mathastext each time math mode is entered.

The third category deserves some brief additional comments: it mainly (but not exclusively) regards the “math activation” of characters, and conversely all “math activations” fall into this category. The package re-checks each time math mode is entered if some characters have been made in-between catcode active, or math active, and takes appropriate decisions: one important aspect of this issue is that babel’s mechanism for activating character was not, last time I checked, very robust against math active characters. I now checked again (on January 15, 2016) that

\documentclass{article}
\usepackage[french]{babel}
\usepackage{mathtools}\mathtoolsset{centercolon}
\begin{document}$:\$end{document}

creates an infinite loop (see section 1.13 where this was mentioned already, some years ago). Thus mathastext has (since 1.2e 2013/01/10) a somewhat elaborate mechanism related to these issues (see the code comments), installed into the list of things done by \TeX systematically each time it enters math mode. For some legacy reason the package also puts into this list a few other things which could arguably be done elsewhere once and for all. The command \MTeverymathoff cancels all actions done by mathastext.

2.2.1 Expert commands usable everywhere

- \MTsetmathskips\{(a-z/A-Z)\}\{\muglue\_before\}\{\muglue\_after\}: is used to specify extra skips (or rather mu glue) to be inserted in math mode, before and after a letter. The rationale is that standard text fonts used in math mode may sometimes cause glyph (near-) collisions with math symbols, as \TeX has some implicit expectations on the design of fonts for math letters.

These extra skips around letters are set at their natural width and do not add any stretchability or shrinkability to the math formula as a whole, nor do they result in extra potential break points.

Random (silly) examples:
\MTsetmathskips\{x\}\{\medmuskip\}\{\thickmuskip\}
\MTsetmathskips\{A\}\{.5\mu\}\{2.3\mu\}
and the effect: $vw \times yzA BCvwx yzABC$. The effect obeys the usual \LaTeXX{} scoping rules.

The first argument of \texttt{MTsetmathskips} may be any expandable code giving a letter; this facilitates use of \texttt{MTsetmathskip} in \texttt{@for} loops such as this one:

\begin{verbatim}
\makeatletter
\@for\@tempa:=a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\do{\MTsetmathskips{\@tempa}{2mu}{2mu}}
\makeatother
\end{verbatim}

Starting with v1.3i: the extra skips are not applied to the letters within the scope of math alphabet commands, or the letters from operator names (pre-defined or user declared).

Note that contrarily to the \texttt{MTexistsskip}, \texttt{MTforallskip}, and \texttt{MTprimeskip} commands described next, these extra skips (which may be specified in the preamble) are not recorded in the definition of the math version (as defined via \texttt{Mathastext} with its optional argument or via \texttt{MTDeclareVersion}). The declared skips hold throughout the document until modified or canceled, independently of math versions (of course, \texttt{mathastext} cancels the skips in the normal and bold math versions if package option \texttt{subdued} was used).

- \texttt{MTunsetmathskips}\{a-z/A-Z\}: cancels the skips for that letter (they are not set to 0\textmu{} but completely removed).

The argument may be a macro (or any expandable code) expanding to a letter.

- \texttt{MTexistsskip}\{math glue\}: specifies the amount of skip or more generally glue to put after each $\exists$ math symbol. Indeed, upright letters (or digits for that matter) often appear to be positioned a bit too close to the quantifier: $\exists B$. The package default is to add a 1\textmu{} skip (this default is set to zero in the case of \texttt{italic}): $\exists B$. One can change the default with the following syntax: \texttt{MTexistsskip}{2mu plus 1mu minus 1mu}, which if used in the preamble and followed with a \texttt{Mathastext} command (or \texttt{MTDeclareVersion}), will be recorded in the definition of this math version (and subsequent ones). One may also use the command at any time in the document. In the case of the option \texttt{subdued}, the skip is canceled in the normal and bold math versions. In the case of the option \texttt{italic}, the default skip is set to zero.

- \texttt{MTnormalexists}, \texttt{MTexistssdoeskip}: the latter (done by default if not \texttt{subdued}, and also on each use of \texttt{MTversion} in the body of the document) makes it so that $\exists$ takes into account the math glue as specified by \texttt{MTexistsskip}. The former is its opposite.

- \texttt{MTforallskip}\{math glue\}: the default is to add a $0.6667\textmu{}$ math skip after each $\forall$ (except with the option \texttt{italic} for which the default skip is set to zero). Compare $\forall F$ (has the skip) with $\forall F$ (has no skip). Use this command in the
preamble to set up the skip or glue to be used in the next to be declared math versions. In the case of the option subdued, the skip is canceled in the normal and bold math versions. In the case of the option italic, the default skip is zero for all math versions. One may use the command at any location in the document.

• $\MTnormalforall$, $\MTforallloesskip$: the latter (done by default if not subdued, and also on each use of $\MTversion$ in the body of the document) makes it so that $\forall$ takes into account the math glue as specified by $\MTforallskip$. The former is its opposite.

• $\MTprimeskip\{\text{math glue}\}$: the default is to add a 0.5\textmu skip before the derivative glyph, except for the italic option. In the case of the option subdued, the skip is canceled in the normal and bold math versions.

• $\MTlowerast\{\text{dimen}\}$: a \raisebox command is used to lower the text asterisk to produce a reasonable math asterisk. The package uses this command initially with argument $0.3\text{height}$, this will have to be fine-tuned for each given text font but worked out ok with the fonts we tried. Note that the dimension argument will be used also in sub-scripts and sub-sub-scripts, so it is best not to use an absolute dimension.

• $\MTmathoperatorsobeymathxx$, $\MTmathoperatorsdonotobeymathxx$: the former is done by default, it makes operator names obey math alphabets. See also section 1.11. This functionality does not rely on “math active characters”. Automatically issued by each $\MTversion$, except under option subdued when switching to normal or bold.

• $\MTcustomgreek$: in case \texttt{mathastext} has been loaded with one of its Greek related options, this activates the corresponding customization of Greek letters in math mode. It is issued automatically by the package in the preamble (except if loaded with subdued option) and at each switch of math version via $\MTversion$ or $\MTversion*$ (except for the normal and bold math versions in subdued mode). Also available as $\\texttt{Mathastextcustomgreek}$. May be used even inside of math mode.

• $\MTstandardgreek$: in case \texttt{mathastext} was loaded with one of the Greek related options this command reverts the customization, it resets the Greek letters to their definitions in force at package loading time. Can be used in the preamble, but is mainly for the document body (may even be used inside math mode ...). Done automatically under the subdued option when switching to the normal or bold math version. Also available as $\\texttt{Mathastextstandardgreek}$. 

2.2.2 Expert commands which are preamble-only

• $\MTgreekupdefault$: a command with no argument whose expansion specifies, under LGRgreek regime, the shape for the ‘up’ Greek control sequences (and for
the no-postfix Greek control sequences under \texttt{upgreek} option) in all \texttt{mathastext} math versions declared \texttt{afterwards} via \texttt{\Mathastext} or \texttt{\MTDeclareVersion}. The a priori default for this shape is ‘n’ (without the quotes). See subsubsection 1.7.3. This command can also be defined \texttt{prior} to loading the package, as the package itself only does:
\begin{verbatim}
providecommand*\MTgreekupdefault{n}
\end{verbatim}

- \texttt{\MTgreekitdefault}: a command with no argument whose expansion specifies, under \texttt{LGRgreek} regime, the shape for the ‘it’ Greek control sequences (and for the no-postfix Greek control sequences under \texttt{itgreek} option) in all \texttt{mathastext} math versions declared \texttt{afterwards} via \texttt{\Mathastext} or \texttt{\MTDeclareVersion}. The a priori default for this shape is ‘it’ (without the quotes). See subsubsection 1.7.3. This command can also be defined \texttt{prior} to loading the package, as the package itself only does:
\begin{verbatim}
providecommand*\MTgreekitdefault{it}
\end{verbatim}

\textbf{2.2.3 Expert commands usable only outside of math mode}

They are usable only from outside math mode because they act via turning on or off the execution, each time math mode is entered, of certain macros added by \texttt{mathastext} to the \texttt{\everymath} and \texttt{\everydisplay} token list variables.

- \texttt{\MTmathactiveletters}: activates the ‘math activation’ of Latin letters. This is done by the package during loading, except under the \texttt{subdued} option. It is again executed in the body at each \texttt{\MTversion}, except under the \texttt{subdued} option when switching to the \texttt{normal} or \texttt{bold} math versions.

  The letters are made mathematically active\textsuperscript{74} to insert the extra skips as specified by \texttt{\MTsetmathskips} (see section 1.8), and also possibly the italic corrections when using upright fonts (see section 1.9).

- \texttt{\MTmathstandardletters}: cancels the ‘math activation’ of the letters. Must be re-issued after each \texttt{\MTversion}, but see \texttt{\MTeverymathdefault}.

- \texttt{\MTicinmath}: this command is executed by default by \texttt{mathastext} except in case of option \texttt{subdued} or if the user chosen letter shape is oblique (\texttt{it} or \texttt{sl}). It tells \texttt{mathastext} to add italic corrections after all letters in math mode, except within the scope of math alphabets.

  This command and the next ones in this item can be used in the preamble as well as in the body of the document (in case of \texttt{subdued} option, using the commands from within the preamble will remain without effect, as the document body will start in the subdued normal math version anyhow.) But each \texttt{\MTversion} in the

\textsuperscript{74}the mathcode’s are only modified at the time of execution of \texttt{\everymath}, \texttt{\everydisplay}. 

51
body will re-emit $\text{\textbackslash MTicinmath}$ (in case of non-oblique letter shape), except if the \texttt{subdued} option was used and the chosen math version is \textit{normal} or \textit{bold}.

The effect of this and the other commands of this item is local to the group or environment in which it has been issued.

It may theoretically be used from inside math mode, but the included $\text{\textbackslash MTmathactiveletters}$ will have an effect only if issued prior to entering math mode.

$\text{\textbackslash MTnoicinmath}$: this command deactivates the package added italic corrections. It can be used inside as well as outside of math mode (or in the preamble of the document).

$\text{\textbackslash MTICinmath, \textbackslash MTnoICinmath}$: these commands activate the italic corrections only for the uppercase letters (but recall that $\text{\textbackslash MTicinmath}$ is done by default, thus this will typically have to follow $\text{\textbackslash MTnoicinmath}$.)

$\text{\textbackslash MTicalsoinmathxx}$: this command de-activates the de-activation of the italic corrections inside the arguments to the math alphabet commands. It can be issued inside as well as outside of math mode. Will be effective only if $\text{\textbackslash MTicinmath}$ or $\text{\textbackslash MTICinmath}$ is in force. To cancel its effect either enclose it in a group or environment or re-issue $\text{\textbackslash MTicinmath}$ after it.

- $\text{\textbackslash MTnormalasterisk, \textbackslash MTeactiveasterisk}$: the latter will use for * and \ast the text font asterisk, suitably lowered; the former tells $\text{\textbackslash mathastext}$ to not modify the $\text{\textbackslash LaTeX}$ default. Both are no-op without option \texttt{asterisk}.

- $\text{\textbackslash MTeasynonlettersobeymathxx, \textbackslash MTeasynonlettersdonotobeymathxx}$: the former is done by default, it makes characters , /, \, #, $, \%$, and & (if not excluded by package options) obey math alphabet commands. See also section \texttt{1.11}. This functionality does \textit{not} make the characters “math active” (but it does modify $\text{\textbackslash mathcode}$’s, naturally).

- $\text{\textbackslash MTnonlettersobeymathxx, \textbackslash MTnonlettersdonotobeymathxx}$: the former will make (except if excluded by relevant package options) !, ?, ,, ;, +, -, =, (, ), [, ], <, and > obey the math alphabet commands (when not used as delimiters). These characters are made “math active”, and each one now expands to two tokens. This makes for example $a^!$ illegal input and it will have to be coded $a^{\text{	extbackslash{}}!}$. Hence, by default, the package does $\text{\textbackslash MTnonlettersdonotobeymathxx}$.

\begin{center}

Under \texttt{subdued} option, $\text{\textbackslash MTnonlettersobeymathxx}$ effect is of course canceled in the \textit{normal} and \textit{bold} math versions; but please note that when switching back to a non-subdued math version it will be mandatory to issue again $\text{\textbackslash MTnonlettersobeymathxx}$ explicitly if its effect is to be re-activated.

In particular, executing $\text{\textbackslash MTnonlettersobeymathxx}$ in the preamble or at the start of the document body serves nothing, because the document is in the subdued \textit{normal} math version regime then. It must thus be executed after the first usage of $\text{\textbackslash MTeversion}$ switching to a non-subdued math version, and again on each successive exit from the \textit{normal} or \textit{bold} math versions.

\end{center}
• \texttt{\MTexplicitbracesobeymathxx}: extends an earlier \texttt{\MTnonlettersobeymathxx} to also treat \texttt{\{ and \}. But then \texttt{\left\{, \right\}} must be coded \texttt{\left\{\{, \right\}rbrace} rather. There is also \texttt{\MTexplicitbracesdonotobeymathxx}.

• \texttt{\MTnormalprime, \MTprimedoesskip}: the latter (done by default if not subdued, and also on each use of \texttt{\MTversion} in the body of the document except for the subdued normal and bold math version) makes it so that \texttt{'} takes into account the math glue as specified by \texttt{\MTprimeskip}. The former is its opposite. In all cases the right quote ' is a mathematically active character producing ' as is the default in \TeX{}, it is only its meaning which changes to include or not an extra skip. For some (legacy) reason, this change of meaning is done anew by \texttt{mathastext} each time math mode is entered. The commands of this item are thus no-op from inside math mode.

• \texttt{\MTeverymathdefault}: this hook is executed by \texttt{\MTversion\{\textit{version}\_name\}}, except under option subdued when switching to the normal or bold math versions. Its default meaning is:

\begin{verbatim}
\MTactiveasterisk % this has no effect without option asterisk
\MTprimedoesskip % this makes prime glyph obey extra space
\MTesynonlettersobeymathxx
\MTicinmath % this does \MTmathactiveletters, hence also skips from % \MTsetmathskips are obeyed.
\MTfixfonts % only operant under LuaLaTX.
\end{verbatim}

Notice that under subdued option, switching to the normal or bold version does \texttt{\MTeverymathoff} which includes \texttt{\MTnonlettersdonotobeymathxx}.

The default \texttt{\MTeverymathdefault} which is issued when going back to a non-normal or bold math version doesn't do \texttt{\MTnonlettersobeymathxx}: thus it is up to the user to correct this if needed (no issue without subdued option).

Notice also that \texttt{\MTversion\{\textit{version}\_name\}}, except for normal or bold if subdued does \texttt{\MTforalldoesskip} and \texttt{\MTexistsdoesskip}, which are not included in \texttt{\MTeverymathdefault} actions as they are not related to \texttt{everymath} and \texttt{everydisplay}.

• \texttt{\MTeverymathoff}: does \texttt{\MTnormalasterisk, \MTnormalprime, \MTnonlettersdonotobeymathxx, \MTesynonlettersdonotobeymathxx, \MTmathestandardletters} and \texttt{\MTdonotfixfonts}.

The commands \url{\texttt{\url{nolinkurl}}} of package hyperref and \url{\texttt{url}} from \url{\texttt{url.sty}} (which use math mode under the hood) are patched by \texttt{mathastext} to do \texttt{\MTeverymathoff} automatically: this is needed because \texttt{mathastext} modifies anew some mathcodes each time math mode is entered, hence may overwrite to some extent the specific preparation done by \texttt{\url{\texttt{\url{url,hyperref}.sty}}}.

Automatically done by \texttt{\MTversion} under option subdued if switching to the normal or bold math versions; and \texttt{\MTversion} then does also \texttt{\MTnormalexists} and \texttt{\MTnormalforall}.

• \texttt{\MTfixfonts}: this is operant only under LuaLaTeX. It has the effect that each time math mode is entered macro \texttt{\MTfixmathfonts} will be executed. The latter forces so-called \texttt{base} mode for the used text font in math mode, in an effort to (only partially, see code comments) fix the fact that OpenType features such as Lining Figures were in some
cases not being applied in math mode when one uses text fonts there (text fonts are declared by Lua\TeX{}+\texttt{luaotfload} to use node mode, which is non-functional in math.) It is invoked automatically by the package (except for normal and bold math versions under \texttt{subdued} option), and in normal situations, there is no reason to use it directly.

- \texttt{\textbackslash MTdonotfixfonts}: cancels the job of \texttt{\textbackslash MTfixfonts}. Done automatically in \texttt{subdued} mode when in the \texttt{normal} or \texttt{bold} math version; in normal contexts, there is no reason to use this command. Only operant under Lua\TeX{}.

### 2.2.4 Expert commands usable only in math mode

- \texttt{\textbackslash MTfixmathfonts}: this used to be an internal package macro but it is given a public name at 1.3p because I discovered that $..\hbox{\texttt{\textbackslash mathversion}(foo)$..$}..$ causes an issue and one needs to invoke again \texttt{\textbackslash MTfixmathfonts} after the \texttt{\hbox}, for some reason. To be used only under Lua\TeX{} and only for such rare cases where it may be needed.

### 3 Package options

#### 3.1 Summary of main options

\texttt{italic}: tells \texttt{mathastext} to typeset the ascii letters in math using italic shape; indeed, its legacy historical default is to typeset them in roman (upright) shape.

\texttt{frenchmath}: lowercase ascii letters in italic shape, uppercase in upright shape. Also lets the Greek letters, if the latter are under \texttt{mathastext} influence, be upright, i.e. also the lowercase ones.

\texttt{subdued}: tells \texttt{mathastext} to not change the default fonts or the math alphabets for the normal and bold math versions. The \texttt{mathastext}-ification activates only after \texttt{\textbackslash MTversion\{version\_name\}} usage in the document body, where the \texttt{\{version\_name\}} was declared as an \texttt{mathastext} enriched math version in the preamble via \texttt{\textbackslash Mathastext\{\{version\_name\}\}} or akin package commands.

\texttt{LGRgreek}, \texttt{eulergreek}, \texttt{symbolgreek}: the Greek letters will be taken, respectively from the text font itself (which must be available in LGR encoding), or respectively the Euler or Symbol font.

\texttt{symbolmax}: all characters other than letters and digits, are taken from the Symbol font. This option also makes a number of further glyphs available, such as some basic mathematical arrows, and the sum and product signs. For documents with very simple needs in mathematical symbols, \texttt{mathastext} with option \texttt{symbolmax} may give in the end a PDF file size quite smaller than the one one would get without the package.\footnote{It is even better if compiled via \texttt{latex+dvipdfmx}.}

75
**defaultmathsizes:** prevents `mathastext` from setting up, as it does per default, larger subscripts and superscripts in math mode, and from copying code from the `moresize` package\textsuperscript{76} in order to redefine `\Huge` and define a `\HUGE` command.

### 3.2 Complete list of options

Some items are described succinctly as more developed descriptions were given earlier. They may sometimes simplify by omission and not consider all possible configurations, particularly those resulting from usage of the package commands in the preamble to configure math versions.

Note that this list in not in alphabetical order, the items are grouped roughly by themes. So, objectively, the best for the diligent reader is to read thoroughly all descriptions.

- **basic**: only mathastextify letters and digits.

- **subdued**: acts in a subdued way, which means that the \LaTeX{} “normal” (default) and “bold” (triggered by `\boldmath` or `\mathversion{bold}`), undone by `\unboldmath` or on exit from a scope limiting context such as an environment) math versions are left (not quite: check subsection 1.4.4 for specifics) unchanged and the `mathastext` action is triggered only when switching via `\MTversion{⟨version_name⟩}` (or its starred variant) in the document body to a version previously defined in the preamble via `\Mathastext[⟨version_name⟩]` (or alternative declarative interface such as `\MTDeclareVersion`).

- **italic**: let the Latin letters (both lowercase and uppercase) use the italic shape (`\itdefault`) in math mode. If the package handles Greek letters, also lowercase (but not uppercase) Greek letters will use this a priori italic shape except if some other option such as `upgreek` was used.\textsuperscript{78}

- **frenchmath**: configures the lowercase Latin letters to use italic shape (`\itdefault`), and uppercase Latin letters to be in same shape as for digits and operator names (i.e. a priori `\shapedefault`).

If the package handles Greek letters both lowercase (if under control of `mathastext`, i.e. not for `selfGreek`) and uppercase Greek letters will use the same shape as operator names, except if some other option such as `itgreek` was used.\textsuperscript{79}

\textsuperscript{76}Christian Cornelissen, *Allows font sizes up to 35.83pt*, https://ctan.org/pkg/moresize.

\textsuperscript{77}Under this option `\MTversion{normal}` and `\MTversion{bold}` execute automatically `\MTmathoperatorsdonotobeymathxx`, `\MTeasynonlettersdonotobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTmathstandardletters`.

\textsuperscript{78}Since 1.3x, in presence of the `LGRgreek` option in addition to `italic`, the `\MTgreekitdefault` shape is then used for lowercase Greek letters and `\MTgreekupdefault` for uppercase.

\textsuperscript{79}Under `LGRgreek` and since 1.3x, the `\MTgreekupdefault` is used for Greek letters if no other option such as `itgreek` was employed.
This configuration (i.e. that uppercase Latin letters will be in the same shape as the one for digits and operator names) is not undone in the subdued “normal” and “bold” math versions. It holds throughout the document, but math versions declared by \texttt{mathastext} may use \texttt{MTshape} and \texttt{MTlettershape} to, in effect, obtain whatever configuration is desired.

As a bonus, note that doing
\begin{verbatim}
\usepackage[\texttt{basic,subdued,frenchmath}]{mathastext}
\end{verbatim}

provides a simple manner to obtain the expected shapes of Latin letters in French mathematical typography, in an arbitrary math font configuration from other packages, in case those packages do not provide an option to achieve this.

But, even if \texttt{mathastext} is used via \texttt{LGRgreek} to configure Greek letters, on the other hand the control sequences for Greek letters are all really restored to their defaults (or whatever was configured by other packages loaded prior to \texttt{mathastext}) in the subdued “normal” math version, which limitates the usefulness of the previous paragraph.

On the bright side, the \texttt{\alphaup}, \texttt{\alphait}, \ldots, control sequences will however be with their \texttt{mathastext} meaning, see \texttt{LGRgreek} for more information.

It is not possible (except of course if one is ready to do some low-level \TeX{} coding to re-execute where needed in the document body a few lines of the package internals with appropriate modifications; I said \TeX{}, not \LaTeX{}, as the latter is very much decided to make impossible any kind of math configuration change at this level if not in the preamble) to achieve a “French math” style only in some math versions and not in others. The reason why is that to achieve distinct shapes for uppercase versus lowercase Latin letters, the uppercase letters are assigned internally to the font (which can change from math version to math version) used for operator names. One can still make them slanted using \texttt{MTshape}, but this will also slant the digits, as they are picked from the same font. On the other hand if we do not use the \texttt{frenchmath} option, both uppercase and lowercase Latin letters are always assigned to the same font, so no math version can give them separate distinct shapes. For a small demo though, one can naturally painstakingly use either the \texttt{\mathrm} or \texttt{\mathnormal} alphabet commands to obtain, say under the \texttt{italic} option and no additional configuration, respectively the up shape and the italic shape.

None of the \texttt{frenchmath}, \texttt{frenchmath*}, and \texttt{frenchmath+} options bear any direct connection with the \texttt{frenchmath} package by Antoine Missier (this is in contrast with the fact that the \texttt{decimalcomma} option is directly related with the \texttt{decimalcomma} package by the same author as it tells \texttt{mathastext} to require it). But see subsubsection 1.5.6 for important information about the utility of \texttt{frenchmath*} if the two packages are to be used concurrently.

\begin{verbatim}
\newcommand*{\defaultalphabets}{\mathastext always defines \texttt{\Mathnormal}, \texttt{\Mathrm}, \texttt{\Mathbf} etc... to refer to the \texttt{mathastext}-ified text fonts, and redefines the math alphabets \texttt{\mathrm}, \texttt{\mathit}, \texttt{\mathtt} etc... (but not \texttt{\mathcal} of course) to use them. To avoid the remapping and keep the \texttt{\mathrm} et al. to refer to the non \texttt{mathastext}-ified fonts, use this option. The \texttt{\Mathnormal} et al. commands with an initial uppercase will always be available whether or not this option is made use of.

Prior to 1.32a (and since 1.15f), this option also prevented the package to declare the \texttt{\Mathnormal} et al. and \texttt{\mathnormal\bold} commands. In this context, recall that the dreaded “too many math alphabets” error can only occur on use in the document of too many of such commands, and not at the time of their declarations. The author’s notes from time of 1.15f release (2012/10/25) only say that it may not be “useful” to package user to have both (for example) }
\emph{and} \texttt{Mathrm}, which sounds weird if they are to acquire distinct meanings. So since 1.3za both will exist. In the default package configuration \texttt{mathrm} is configured to expand to \texttt{Mathrm} (with some extra behavior under \texttt{LGRgreek+}), and with this option or the \texttt{defaultrm} option \texttt{mathrm} is kept with its original meaning (and the \texttt{LGRgreek+} extras do not work).

**changed:**

- **defaultnormal, defaultrm, defaultrbf, defaultit, defaultsf, defaulttt:** tell \texttt{mathastext} to not set up, respectively, the \texttt{\mathnormal, mathrm, mathbf, mathit, mathsf, and mathtt} commands to use the mathastext-ified font which are accessible always via \texttt{\Mathnormal, \Mathrm, \Mathbf, \Mathit, etc...} Prior to 1.3za these options also prevented the creation of the corresponding \texttt{mathastext} command with an uppercased initial.

- **ncccomma:** it triggers the loading of the \texttt{ncccomma} package\(^{80}\) and configures \texttt{mathastext} for compatibility (this is canceled if \texttt{nopunctuation} option is used, or \texttt{basic} as it implies it). \texttt{Note that mathastext has NO auto-detection mechanism of ncccomma, the correct way is to use the eponymous option.}

  The effect of the \texttt{ncccomma} package will apply to the entire document body, even to portions using the \texttt{normal} or \texttt{bold} math versions with \texttt{mathastext} having been loaded with the \texttt{subdued} option. Also, in case of usage of package babel with \texttt{french} option, the effect of \texttt{ncccomma} will also apply to those parts of the document using another language than French.\(^{81}\)

- **decimalcomma:** it triggers the loading of the \texttt{decimalcomma} package\(^ {82}\). The same remarks apply as for the \texttt{ncccomma} option. In particular \texttt{note that mathastext has NO auto-detection mechanism of decimalcomma, the correct way is to use the eponymous option.}

  The effect applies to all math versions, even the \texttt{normal} and \texttt{bold} math versions with \texttt{mathastext} having been loaded with the \texttt{subdued} option.

**changed:**

- **frenchmath**\(^*:\) does all three of \texttt{frenchmath, decimalcomma and binarysemicolon}.\(^{1.3zb}\)

  Prior to 1.3zb, this option did what is now available via \texttt{frenchmath+}. The 1.3zb change was made as a follow-up consecutive to the 2.7 release \texttt{frenchmath}. Indeed this option as explained

---

\(^{80}\)Alexander I. Rozhenko, \textit{Use comma as decimal separator in mathematics}, \url{https://ctan.org/pkg/ncccomma}.

\(^{81}\)There is a ‘feature’ of \texttt{babel-french} that the effect of package \texttt{ncccomma} is canceled if one switches from French to English; and switching back to French does not reenact it. For background on this issue see \url{https://github.com/latex3/babel/issues/190}.

This does not apply to \texttt{decimalcomma} 1.3 or later.

\(^{82}\)Antoine Missier, \textit{Comma for decimal numbers}, \url{https://ctan.org/pkg/decimalcomma}.
in subsubsection 1.5.6 is provided as a compatibility layer with \texttt{frenchmath}, and it was mandatory to modify its meaning to refer to package \texttt{decimalcomma}, not \texttt{ncccomma}, consecutive to the internal change of \texttt{frenchmath} at its 2.7 release to use \texttt{decimalcomma}.

- \texttt{frenchmath\textasciip文化交流}: does all three of \texttt{frenchmath}, \texttt{ncccomma} and \texttt{binarysemicolon}. (1.3zb) This is what used to be called \texttt{frenchmath*} prior to 1.3zb.

- \texttt{endash}, \texttt{emdash}: use the text font en-dash (–) or even the em-dash (—, but this seems crazy) for the minus sign rather than -. \texttt{endash} option is default for the package.

- \texttt{unicodeminus}: use the MINUS SIGN \texttt{U+2212} (requires fontspec.) Or, in the form \texttt{unicodeminus=HHHH} with four uppercased hexadecimal digits: use the \texttt{U+HHHH} code point. As \texttt{noendash} really means “use the hyphen from the text font”, \texttt{unicodeminus} remains without effect under it, or, naturally, under \texttt{nominus}. Without this option, \texttt{mathastext} uses the EN DASH \texttt{U+2013} by default for OpenType fonts.

- \texttt{asterisk}: use the text font (or the Symbol font) asterisk in math mode.

- \texttt{nobar}: prevents \texttt{mathastext} from defining its own \texttt{\hbar}.

- \texttt{noendash}: the minus sign will be the - from the text font, not the en-dash –.

- \texttt{nolessnomore}: besides !, ., ;, +, – = ( ), / # $ % & \texttt{mathastext} treats also < > | { } and \. Use this option to let it not do it. This is the default in case of \texttt{OT1}-encoding.

- \texttt{further excluding options}: \texttt{noexclam} !?, ., ;, +, – = ( ), / # $ % & \texttt{mathastext} treats also < > | { } and \. Use this option to let it not do it. This is the default in case of \texttt{OT1}-encoding.

- \texttt{alldelims}: true by default, means that the characters excluded by \texttt{nolessnomore} are treated. Use this option in case of a mono-width \texttt{OT1}-encoded font.

- \texttt{nosmalldelims}: this prevents \texttt{mathastext} from trying to pick up in the text font the ‘small variants’ of some math delimiters; it only affects what happens when a character such as a left parenthesis ( or [ is used as a delimiter, and in the event that \texttt{TeX} has chosen the smallest sized variant. This has no impact on what happens when they are not used as delimiters: then, and if not disabled by the corresponding options, these characters are always picked up from the text font. \footnote{In this very special situation of option \texttt{nosmalldelims}, the braces are an exception to this rule and they require both of \texttt{\MTnonlettersobeymathxx} and \texttt{\MTexplicitbracesobeymathxx} for being picked up from the text font when not used as delimiters.}

- \texttt{symbolgreek}, \texttt{symboldigits}: to let Greek letters (digits) use the Symbol font.
• `symbolre`: replaces \Re and \Im by the Symbol glyphs ℜ, ℑ and defines a \Dot Triangle command (\cdot).

• `symbolmisc`: takes quite a few glyphs, including logical arrows, product and sum signs from Symbol. They are listed `supra`. Doing `\renewcommand{\int}{\smallint}` will maximize even more the use of the Symbol font.

• `symboldelimiters`: the characters apart from letters and digits will be taken from the Symbol font.

• `symbol`: combines `symbolgreek`, `symbolre`, and `symbolmisc`.

• `symbolmax`: combines `symbol` and `symboldelimiters`.

• `eulergreek`, `eulerdigits`: to let Greek letters (digits) use the Euler font.

• `LGRgreek`: this configures the Greek letters in math mode to use the text font (i.e. a priori the font which was default at time of loading the package) in LGR-encoding. The command `\MTgreekfont` can be used to set a specific (LGR-encoded) font family. Each use of `\MTgreekfont` must be followed at some point by a `\Mathastext` or `\Mathastext[⟨version_name⟩]` to be effective. Any subsequent math version declaration will be influenced by it until `\MTgreekfont` is used again to configure another font for Greek letters.\footnote{You can check the documentation of the `https://ctan.org/pkg/lgrmath` package for how to find out systematically which fonts are available on your system in LGR encoding.}

If `\MTgreekfont` is never used the font family for Greek under option `LGRgreek` will be, in all math versions except under `subdued` for the “normal” and “bold”, the family which was the default at time of loading the package. You must use `\MTgreekfont` to change it.

See further on this topic the `upgreek`, `itgreek`, `upGreek` and `itGreek` options as well as the `\MTupgreek`, `\MTitgreek`, `\MTupGreek` and `\MTitGreek` commands.

It is up to the user to ascertain that the font family is indeed available in the LGR encoding; if it is not, only at time of the first math mode typesetting will `\LaTeX` issue warnings such as this one:

```
Font shape `\LGR/ptm/m/n' undefined using `\LGR/cmr/m/n' instead on input line 28
```

The `LGRgreek` option also triggers pre-definition of Greek character tokens such as `\alphaup` or `\betait`, see subsection 1.7.3 for the explanations.

Although under `subdued` option `\mathastext` restores Latin (but see `frenchmath`) and Greek letters in the “normal” and “bold” math versions it still under `LGRgreek` option keeps in these “subdued” math versions the package declared `\alphaup`, `\alphait`, ..., and the associated `\mathgreekup` and `\mathgreekit` commands to access the underlying fonts, and also since 1.3za `\mathgreekupbold` and `\mathgreekitbold`.

The font used by these math alphabet commands in the subdued “normal” and “bold” is either the one in LGR encoding which was the family default at time of loading the package or the one
configured last by \MTgreekfont when the command \Mathastext (without optional argument) was used in the preamble.

1.3za fixes here a bug which froze the target font to be the one at time of loading the package: this bug applied (only) to the subdued “normal” and “bold” math versions and was not readily visible as there is a priori no reason to use in these subdued math versions these mathastext-provided Greek font alphabets.

- **LGRgrees**: each declared math version will be supposed to be with a font which is also available in LGR-encoding. This is a shortcut to using \MTgreekfont systematically to keep in sync in all declared math versions the font for Greek with the font for Latin letters. Please note that macro \MTgreekfont becomes then inoperaent, and if you need one math version without this Latin-Greek synching, you will have to use rather LGRgreek and then \MTgreekfont manually appropriately.

- **LGRgreek+ and LGRgrees+**; they extend respectively LGRgreek or LGRgrees (1.3za) to let Greek letters control sequences when in the scope of \mathrm, \mathit, and \mathbf behave as would be expected by LaTeX users who have not read fntguide.pdf or any other \LaTeX documentation but have used unicode-math. See subsubsection 1.7.5 for details.

- **selfGreek**: this is for a font which is also available in OT1-encoding and contains the glyphs for the default eleven capital Greek letters.

  This option should have been named \texttt{OT1Greek} as it bears about the same relation with \texttt{OT1} encoding (for eleven capital Greek letters) as \texttt{LGRgreek} does with the LGR encoding (for the complete no-diacritics Greek alphabet).

- **selfGreeks**: each declared math version will be supposed to be with a font with the eleven capital Greek letters in its OT1-encoded version.

- **upgreek, itgreek**: options to tell mathastext to use \MTgreekupdefault or \MTgreekitdefault for the lowercase and uppercase Greek letters shape. These two commands can be defined prior to loading the package. This option is operant only under the LGRgreek(s) or selfGreek(s) options.

- **upGreek, itGreek**: influence only uppercase Greek.

- **mathaccents**: use the text font also for the math accents. As in vanilla \LaTeX, they are taken from the font for the digits and \texttt{\log/-like names}. Obey the alphabet changing commands.

- **unimathaccents**: extends mathaccents to OpenType fonts. Gave bad results in (1.3u) my brief testing.

- **defaultimath**: do not overwrite \texttt{\imath} and \texttt{\jmath} to use \texttt{\inodot} and \texttt{\jnodot}.

- **defaultmathsizes**: do not change the \LaTeX defaults for the sizes of exponents and subscripts.
• **fouriervec**: provides a \fouriervec command. The user can then add in the preamble \let\vec=\fouriervec. There is also always available a “poor man” vec accent \pmvec for upright letters.

Thanks to Kevin KLEMENT, Tariq PERWEZ and Ricard TORRES for sending bug reports and feature requests when the first version of the package was issued.

Numerous examples will be found there:
- [http://jf.burnol.free.fr/mathastext.html](http://jf.burnol.free.fr/mathastext.html)
- [http://jf.burnol.free.fr/showcase.html](http://jf.burnol.free.fr/showcase.html)
4 Change log

1.3zb [2023/12/29]
- Update to the frenchmath* option to maintain compatibility with the [frenchmath](https://ctan.org/pkg/frenchmath) package whose release 2.7 (2023/12/23) has replaced the ncccomma package by the decimalcomma package.
  - The frenchmath+ option holds the former meaning of frenchmath*.
  - Option decimalcomma to load the eponymous package by Antoine Missier. This is tacitly done by frenchmath*.
  - No more messages sent to the console output during loading, only info messages going into the log, and using (more or less) the official LaTeX interface: after close to 13 years of development of this package it was perhaps finally the time to do it.
  - Documentation improvements. Close to 13 years after the birth of the package, and as it nowadays rarely wakes up from dormancy, this was almost last chance to try to improve a few things.

1.3za [2023/12/20]
- Under LGRgreek and LGRgreeks options, new math alphabets \mathgreekupbold and \mathgreekitbold.
  - New options LGRgreek+ and LGRgreeks+. Thanks to Holger Gerhardt for feature request and code ideas. Please find and read the relevant documentation in the PDF.
  - The meaning of defaultalphabets and related individual options such as defaultbf has been modified (reverted to pre 1.15f release): even under these options, the package always creates \mathnormalbold, \Mathnormal, \Mathrm, \Mathbf etc..., commands. This may break documents which used these options in order to reserve these command names. This was done with some hesitancy, but for the sake of internal logical coherence.
  - Fix an obscure bug with no real consequences regarding interaction of subdued with LGRgreek and \MTgreekfont. See the LGRgreek documentation in the complete list of options for details.
  - Fix long-standing hyperlink problems in the documentation: blue color words should now all be functioning hyperlinks.

1.3z [2023/09/01]
Fix 1.3y regression which broke selfGreek option due to internal remanings. Thanks to Stephan Korell for report.

1.3y [2022/11/04]
(the 1.3x had an annoying documentation bug, and had already been pushed to CTAN, hence the version increase to 1.3y)
- mathastext now requires the \expanded primitive (which is available with all major engines since TeXLive 2019).
  - Revisit parts of the documentation (mainly the Examples, and the section on Greek letters) and shuffle the other parts to surely improve things. Mention the [mathfont](https://ctan.org/pkg/mathfont) and [frenchmath](https://ctan.org/pkg/frenchmath) packages.
  - Add the ncccomma option which loads the [ncccomma](https://ctan.org/pkg/ncccomma) package to allow the comma as decimal separator.
  - Add the binarysemicolon option to let the semi-colon be of type \mathbin, not \mathpunct.
  - Add the frenchmath+ option which does all three of frenchmath, ncccomma and binarysemicolon.
  - Under the LGRgreek and LGRgreeks options only:
    - make available upright and italic Greek letters in math mode via \alphaup, \alphait, ... control sequences, in addition to those not using such postfixed-names.
    - add \mathgreekup and \mathgreekit math alphabets.
    - add \MTgreekupdefault and \MTgreekitdefault. The former replaces \updefault which was used in some places and since LaTeX 2020-02-02 caused systematic Font Warnings about the substitution of up by n.
These new features required an extensive internal refactoring which is expected to not induce changes to most existing documents. But it may induce changes to those using some unusual configuration in the preamble, as made possible via the
package macros; this can apply only to documents authored by those few people who actually read the documentation. For full details make sure to read the PDF documentation about this change.

- Fix \textbackslash Digamma under LGRgreek option uses the shape for lowercase not uppercase Greek.
- Fix some incongruities in log messages related to Greek letters and emitted during math version creation in the preamble.

1.3w [2019/11/16]
- LaTeX 2019-10-01 release (up to patch level 3 inclusive) together with amsmath conspired :-) to break mathastext, in connexion with math accents. This has been fixed upstream, but I am releasing nevertheless a hot fix to this https://github.com/latex3/latex2e/issues/216 issue (this is compatible with future LaTeX releases).

- Fix: \hbar is originally a robust command but becomes a \mathchardef token if (e.g.) amsfonts is loaded and then with recent LaTeX \hbar<space> is made undefined and mathastext definition of it remained without effect. The \mathastext own \hbar is now defined \protected.

- Fix: option noendash (or symboldeliminators which implies it) caused (since 1.3u) a bug under Unicode engines when setting up the minus sign.

- Version names declared via the optional argument of \Mathastext or as first argument of \MTDeclareVersion must not be normal or bold. Enforce that! (this was marked as a bug to fix since 2012/10/24...)

1.3v [2019/09/19]
- LaTeX 2019-10-01 release has made more math macros robust. This applies in particular to the math accents and to the \hbar. This required for mathastext to adapt. Also \leftarrowfill and \rightarrowfill are now defined robust by the kernel, hence mathastext does the same. These changes are dropped if mathastext detects an older LaTeX format.

- These LaTeX kernel changes motivated an examination of some redefinitions done (optionally) by mathastext:
  - The user math alphabet macros got redefined as expanding to some other (robust) math alphabet macros, but were not robust in the strict sense. This does cause some issues for moving arguments in the context of multiple math versions, hence it was a bug. The special behaviour of the math alphabet commands (they redefine themselves and other macros on first use) makes is somewhat problematic for mathastext to keep them updated across math versions and at the same time strictly LaTeX2e robust. Thus mathastext now requires the e-TeX primitive \protected and uses it for the definitions of the user level math alphabet macros.
  - There are a number of \mathchardef tokens which (under certain options and/or configuration via the package user interface), mathastext redefines as macros. These macros cause no issue in moving arguments (they are not "fragile"), still it is probably better if they expand only at the time of typesetting. To this effect they are now also \protected: \exists, \forall, \colon, \setminus, \mid, \prod, \sum, \imath, \jmath.
  - The macro \verb (which expands to a \delimiter) is now defined robust by LaTeX. Its mathastext redefinition is a \protected one rather.
  - The \{ and \} (which get redefined only under \MTexplicitbracesobeymathxx regime) are now strictly robust in the LaTeX2e sense (formerly they were \let to some robust macros, and this did not make them strictly LaTeX2e-robust entities).

- The various changes in mathastext described in the previous item apply independently of the LaTeX release version. The LaTeX format itself requires the e-TeX extensions since 2015.

1.3u [2019/08/20]
- new feature: the initial release dealt with only one font, and although shortly thereafter the 1.11 version added support for extended math versions, it was documented that some font-dependent setup (minus as endash, dotless i and j, hbar, math accents) was done only once. This release makes the relevant characters font encoding savvy in each mathastext-extended math version. Thus, they should render correctly even with multiple math versions using fonts with varying encodings.

This reinforces importance of using \MTversion and not the LaTeX \mathversion when switching to a new math version (which got declared via the package interface). The implementation is compatible with Unicode engines and mixed usage of TU encoding (OpenType fonts) with traditional 8bits TeX font encodings. For all engines, all used (8bits) encodings must have been passed as options to the fontenc package.

Thanks to Falk Hanisch for feature request and code suggestions.
* new option unimathaccents: this adds to option mathaccents the demand to use the text font accents for OpenType fonts in math mode via the \Umathaccent primitive. Indeed, as my testing showed that this gave non-satisfactory results both with XeTeX and LuaTeX regarding the horizontal placement of the accents, the main option mathaccents acts only on 8bits encoded fonts.

* bugfix: the \Mathastext without optional argument forgot to repeat some font-encoding dependent initialization set-up done originally during package loading.

* bugfix: under the subdued option macros \MTmathactiveletters or \MTnonlettersobeymathxx now act like no-ops if issued explicitly while in the normal or bold math version. Formerly, this was not the case and could cause bugs such as a disappearing minus sign in math mode.

* bugfix: the letter h used in the \hbar obeyed the extra skips as set-up by \MTsetmathskips, badly interfering with the horizontal positioning of the bar accent. They are now ignored (as well as the added italic correction).

1.3t [2018/08/22]
* bugfix: the 1.3s bugfix about subdued compatibility with fontspec was deficient.

* bugfix: very old (v1.2, 2012/12/20) bug causing low-level TeX error during package loading (with pdflatex) when setting up the math minus sign to be the text font endash character, in cases with \encodingdefault other than OT1, T1 or LY1, e.g. something like T2A.

* \imath and \jmath obey the subdued regime. And the minus sign is now handled especially to ensure perfect compatibility with the subdued option.

* breaking change: mathastext does not redefine anymore \textsl{i} and \textit{j} to let them be usable both in text and math mode.

1.3s [2018/08/21]
* fix to an issue with subdued option in a fontspec context.

1.3r [2016/11/06]
* documentation tweaks.

1.3q [2016/10/31]
* new option unicodeminus.

* the Recent Changes section of the documentation has been removed as it was a duplicate of information available in the Change Log.

* some other changes in the documentation, in particular the use of straight quotes in verbatim.

1.3p [2018/05/13]
* bugfix: release 1.3n had forgotten to activate by default its new customization of the amsmath macro \newmcodes (it was done from using \MTversion in the document body but not by default at start of body.)

* public name \MTfixmathfonts for a 1.3o macro.

1.3o [2016/05/03]
* mathastext fixes an issue related to a feature of LuaLaTeX and luatex85 that OpenType fonts are declared in one of two modes: node and base, and only the latter is functional in math mode. But by default text fonts are declared in mode node. Thus mathastext now intervenes to make it so that the font it declares in math mode will use mode base. This fixes issues with for example old style figures being used while the text font used lining figures (or vice versa, depending on the font). But see the code comments for more.

1.3n [2016/04/22]
* at long last, mathastext takes care properly of annoying and perplexing amsmath's \newmcodes. The very recent change in amsopn.sty finally made it compatible with Unicode engines, but anyhow, mathastext must do its own patch to use the correct font. All of this taking into account the various options passed to the package. Lots of trouble for a tiny thing.

1.3m [2016/04/02]
* minor code maintenance before annual TL freeze.

1.3l [2016/01/29]
* compatibility with fontspec’s upcoming switch from EU1/EU2 to TU common to both Unicode engines.

1.3k [2016/01/24]
* typos fixed in the documentation. In particular, the README link to the package homepage had remained broken from day one of the package releases: mathastext.html therein was misspelled
as mathsastext.html ! (but the pdf documentation had the correct link; as well as the CTAN catalogue).

1.3j [2016/01/15]
* renamed and modified recent 1.3i’s \MTactivemathoff into \MTeverymathoff. Added \MTeverymathdefault.
* subdued mode is a bit stronger: also the asterisk reverts to the default (if it was modified due to option asterisk), the added extra \mskip’s (useful with upright fonts) for \', \\exists, and \\forall are suppressed rather than re-configured to use 0\mu. Related new commands \MTexistsdoesskip, \MTeverydisplaydoesskip, \MTprimedoesskip, \MTnormaltreeskip, \MTnormalforallsmallskip.
* the toggle for using mathematically active letters is only emitted once during package loading; the \Mathastext command does not do it anymore; the use in the preamble of \MTmathstandardletters, or \MTnoicinmath and related commands is not overruled by later use of \Mathastext.
* quite a few documentation improvements and rewrites, particularly in the description of commands which are related to the modifications of mathcodes (mainly for math activation of characters or letters) as done by mathastext at \everymath or \everydisplay.

1.3i [2016/01/06]
* \url from url.sty as well as \url and \nolinkurl from hyperref.sty use math mode and (by default) the monospace text font. To avoid mathastext overwriting the special preparation done by \url,hyperref.sty the commands \url/\nolinkurl are patched to do automatically \MTactivemathoff (now \MTeverymathoff) before entering math mode.
* the extra skips specified by \MTsetmathskips are not inserted around letters if inside the arguments of math alphabet commands, or within operator names.
* the added explicit italic corrections (for non-oblique fonts) were disabled within math alphabet scopes, except mathnormal; they are now disabled within all math alphabets, inclusive of mathnormal.

1.3h [2015/10/31]
* bugfixes: since 1.3d 2014/05/23 the option symbolgreek caused \ell to become undefined, and, similarly but far worse, options selfGreek, self-Greeks caused all lowercase Greek letters \alpha, \beta, etc. to become undefined.

1.3g [2015/10/15]
* following 2015/10/01 LaTeX release, removal of the "luatex" prefix from the names of the LuaLaTeX math primitives. Compatibility maintained with older LaTeX formats.

1.3f [2015/09/12]
* the replacement of amsmath’s \resetMathstrut, when it is done, emits an Info rather than a Warning as this could be potentially stressful to some users.
* the README self-extracts from the dtx source, as a text file README.md with Markdown syntax.

1.3e [2015/09/10]
* bugfix: under option nosmalldelims, \lbrace and \rbrace were redefined as math symbols and could not be used as delimiters.

1.3d [2015/02/26]
* the documentation mentions the improved compatibility of mathastext with the latest (3.34) beamer release: no more need for \usefonttheme{professionalfonts}.

1.3d [2014/05/23]
* new commands \MTstandardgreek and \MTcustomgreek.
* The Greek letters, in case of use of one of the package related options, are left to their defaults in the normal and bold math versions if the subdued option was also used (this was so far the case only with options LGRgreek/LGRgreeks).
* \newmcodes@ of amsmath is left untouched if package lualatex-math is detected.

1.3c [2013/12/14]
* added a starred variant to \MTversion which tells mathastext to only do the math set-up and not modify the text fonts.
* added second optional version name argument to \Mathastext and to \MTDeclareVersion, to transfer settings for things not otherwise changed by mathastext from a math version to the one declared. This is mainly for symbols and large symbols to be the bold ones when the user sets up
the series of a mathastextified font to be bold in a mathastext-declared version.

* renamed \defaultprod to \MToriginalprod, \defaultsum to \MToriginalsum, (this is in case of option symbolmisc).

* changes to the dtx organization; options for generating the documentation can be customized in generated mathastext.tex file.

* 1.2d code for \#, \$, \%, and \& modified erroneously the earlier correct 1.2c code and created a bug showing up with more than 16 math families (a possibility only with lualatex or xelatex).

1.3a [2013/09/04]

* the somewhat silly \string’s are removed from the \MTsetmathskips command of release 1.3, thus allowing its first argument to be a macro, or any expandable code, giving a letter.

* the amsmath \resetMathstrut@, which is incompatible with a mathematically active parenthesis ( is now modified only if necessary (i.e. \@ only when \MTnonlettersobeymathxx is issued) and is restored to its original value if not needed anymore (i.e. after \MTnonlettersdonotobeymathxx, as for example when switching to the normal version under option subdued).

* improved documentation.

1.3 [2013/09/02]

* commands \MTsetmathskips and \MTunsetmathskips added.

* commands \MTmathactiveletters and \MTmathstandardletters to govern the math activation of letters independently of its use for insertion of the italic corrections ( \MTicinmath and \MTnoicinmath correspondingly modified).

* the new \luatexmUmathcodenum as available since TL2013 allows identical treatment by mathastext of = and - under both LuaTex and XeTeX.

* \newmcodes@ of amsmath is left untouched in case of option basic.

* a sentence containing | which was written to the log during the loading caused a problem if | was active (typically if \MakeShortVerb{|}) was added to the preamble prior to the loading of mathastext).

* some preemptive measures taken regarding things such as \mid, \brace, and \rbrace, as some packages define these things in manners which made the re-definitions done by mathastext issue errors.

1.2f [2013/01/21]

* minor code improvements. Change log added to the user manual.

1.2e [2013/01/10]

This version should be the last one in the 1.2 series as it seems to correct most of the main problems which were introduced with the massive use of mathematically active characters in versions 1.2 and 1.2b.

* It is indeed a thorny point when one wants to modify an active character in math mode only (without breaking usage in label’s and ref’s for example). The package now does that _only_ if the activation originated in the Babel system as it is then possible to modify appropriately the Babel macros \user@active<char> and \normal@char<char>, at the time of entering math mode (mathastext does all its activation job at \everymath and \everydisplay).

The relevant issues are discussed in section 2.10 of the user manual, in the test file mathastexttestalphabets.tex, and in the source code comments for macro \mst@mathactivate. The inherent incompatibility of Babel with packages having made mathematically active the characters itself makes document active is circumvented by this interference of mathastext. A generally applicable Babel patch could be derived from the method used by mathastext.

For the non catcode active characters, mathematical activation is used. This is done at the entrance in math mode.

* Sadly, the feature of added italic corrections introduced in version 1.2b did not behave as described in the user manual, due to forgotten group braces. Fixed.

* The command \MTlowerast from the user manual of v1.2d was not the one implemented in the source code. Fixed.

* The test files automatically extracted from a latex run on the dtx file have been revised and extended.

* The code is better documented.

1.2d [2013/01/02]

* an incompatibility with amsmath (its macro \resetMathstrut@), exists since version 1.2 of the package. This is fixed here.
Various improvements in dealing with the asterisk and in the mechanism of letting non-letter symbols obey the math alphabet commands.

Documentation extended and improved.

1.2c [2012/12/31]

- Mathastext now inserts automatically after all (Latin) letters in math mode their italic corrections, if the font used is upright (sic). This improves the spacings for the positioning of subscripts. The feature is de-activated inside the math alphabets commands (apart from \mathnormal), so as to not prohibit the formation of ligatures.
- The documentation has been extended to explain in detail the issues which are relevant to the new feature of added italic corrections.
- Version 1.2 had some bad bugs when confronted to active characters. This is corrected and additionally \MTnonlettersdonotobeymathxx is made the default, as the user input is too much constrained in its absence.
- A less fatal, but still annoying, typo had made the dot in 1.2 of type \mathpunct rather than \mathord.
- The inner namespace has been rationalized a bit.

1.2 [2012/12/20]

- A new command sets up the amount of space to be automatically inserted before the derivative glyph (useful when using an upright font).
- The scope of the math alphabets has been extended to apply to the non-alphabetical characters, and also to operator names.
- The format of the dtx file has changed. The package file is self-extracting from the dtx, and four additional test files are also produced during latex mathastext.dtx.

1.15f and 1.15g [2012/10/25]

- \$, \#, \&, and \% had been re-defined by mathastext since its inception in a rather strange (but working) way, which could cause surprises to other packages. Fixed.
- The subdued mechanism for the math alphabets is implemented in a simpler and more efficient manner than in 1.15e.
- The defaultxx options act a bit differently, and are more useful in case of a too many math alphabets situation.

1.15e [2012/10/22]

- New user commands to specify skip or glue to be inserted after the math symbols \exists and \forall.
- Complete (user transparent) rewrite of the code implementing the subdued option; and its action has been extended to apply also to the \mathbf, \mathit, \mathsf, \mathtt alphabets and not only to \mathrm and \mathnormal as in the previous versions.
- Improvements in the documentation.

1.15d [2012/10/13]

- The Unicode situation is now correctly treated, throughout the code (this had been left in a half-done way from version 1.14 of April 2011).
- This includes an issue related to amsmath and its DeclareMathOperator macro which has been fixed.
- And the code related to \relbar and \Relbar (and \models) has been revised.

1.15c [2012/10/05]

- It is now possible to use distinct fonts in LGR encoding for the Greek letters according to the current math version.
- Improvements to the documentation.

1.15b

- Corrected a ‘feature’ of 1.15 which was backward-incompatible.

1.15 [2012/09/26]

- The subdued option allows the mathastextification to act only locally.
- Some measures taken to deal with amsmath related issues when using xetex or luatex.

1.14c

- A bug is fixed: the \Mathastext macro reinitializes the fonts in the normal and bold math versions, but it also erroneously redeclared the math alphabet changing commands which could have been set up in previously defined math versions (via earlier calls to \Mathastext\{version_name\}).
1.14b [2011/04/03]
* there was a bug with $\$, \#, \&, \% in math mode which showed up when ten or more math families had been declared. This bug affected also the minus sign under the same circumstances, when Unicode engines were used. Fixed.
* the options LGRgreek and selfGreek act now a bit differently, and new options LGRgreeks and selfGreeks have been defined.
* I also cleaned up a bit the code, for a more structured namespace.

1.14
* mathastext now modifies also the math alphabets \mathit, \mathsf and \mathtt, thus making it a quite generic complete manner to adapt the math configuration to fonts provided with no math support.

1.13b
* when the Symbol font is used for \prod and \sum this will be only for inline math; display math will use the default glyphs

1.13 [2011/03/11]
* the LGRgreek option is added.

1.12
* various bugs have been corrected.
* the endash and alldelims options are active by default.
* the package is more Unicode aware.
* the \Mathastext command has been improved to facilitate the mechanism of math versions also when using XeTeX or LuaTeX (with package fontspec.)
* the en-dash and dotless i and j now work with all encodings, Unicode inclusive.

1.11 [2011/02/06]
* optional argument to \Mathastext macro.

1.1 [2011/02/01]
* options italic and frenchmath.

1.0 [2011/01/25]
* Initial version.
5 Implementation

The usual catcode regime for letters and digits is assumed and some characters such as *, `, ", = are supposed to be of catcode other at the time of loading of `mathastext`. The source of `mathastext` takes precautions for some other characters such as the right quote ``, which may thus be active with no harm at the time of loading. By the way, I think `\LaTeX2e` should have provided to authors a standard macro to be used at the beginning of a style file to make sure the catcodes are standard. Shorthands created by Babel should be mostly no problem as Babel does the activation only at the `\begin{document}`.

The comments have been accumulating through successive versions with only partial efforts to achieve some sort of coherence; as a result some are a bit strange or obsolete to various degrees. And the similar remark applies to some ancient parts of the code itself!

Should I require `2005/12/01 \LaTeX`? (not sure about the month).

1 `\NeedsTeXFormat{LaTeX2e}`
2 `\ProvidesPackage {mathastext}`
3 `[2023/12/29 v1.3zb Use the text font in math mode (JFB)]`
4 `\def\mst@infoline#1{\immediate\write\m@ne{\space\space\space mathastext:\space\space\space) #1}}`
5 `\immediate\write\m@ne{}`
6 `\PackageInfo{mathastext}{Starting the math mode configuration@gobble}`
7 `\LaTeX 2019-10-01 release has made robust math macros such as the math accents and \hbar.`
8 `\newif\ifmst@robust@obsessed@LaTeX@era`
9 `\@ifl@t@r\fmtversion{2019/10/01}{\mst@robust@obsessed@LaTeX@eratrue}{}`
10 `\edef\mst@robustifyingspace{\ifmst@robust@obsessed@LaTeX@era\space\fi}`

Testing for `Xe\TeX` and Lua\LaTeX.

1.3g 2015/10/15: update for the naming of primitives, the situation has evolved both on Xe\TeX side and on the Lua\LaTeX side (`LaTeX base 2015/10/01`): I was told "U" named math primitives were always available for Lua\LaTeX. For Xe\TeX, the Xe\LaTeX prefix got replaced by U prefix with 0.99.. a certain number of 9. I opted for rather simple approach of just trying the "modern" names and if they don’t exist fall back on earlier (and in danger of being deprecated) names.

11 `\let\mst@Umathcharnumdef\Umathcharnumdef`
12 `\let\mst@Umathcodenum \Umathcodenum`
13 `\let\mst@Umathcode \Umathcode`
14 `\let\mst@Umathchardef \Umathchardef`
15 `\let\mst@Umathaccent \Umathaccent`
16 `\newif\ifmst@XeTeX`
17 `\ifx\XeTeXInterchartoks\undefined`
18 `\mst@XeTeXfalse`
19 `\else`
20 `\mst@XeTeXtrue`
21 `\ifx\mst@Umathcharnumdef\undefined`
22 `\let\mst@Umathcharnumdef\XeTeXmathcharnumdef`
23 `\let\mst@Umathcodenum \XeTeXmathcodenum`

69
1.2: all inner macros of `mathastext` now starts with `\mst@` for a cleaner name-space.

1.31 2016/01/29: hmmm… at this late stage where nobody would expect me to still look at the code, I have found at least two macros which still didn’t: `\do@the@endashstuff` and `\do@the@emdashstuff`.

Ok, doing something more serious: compatibility with upcoming TL2016 fontspec and its switch to ‘TU’ NFSS font encoding in replacement of ‘EU1/EU2’. Anyhow, the code in `mathastext` has been common to the two Unicode engines for a while, hence it is not hard to adapt to the replacement of EU1/EU2 by TU, maintaining compatibility with legacy installations.

\mst@OneifUniEnc

The `\mst@OneifUniEnc` is expandable but must be used after having set `\mst@tmp@enc`...

\mst@enc \mst@fam \mst@ser \mst@opsh \mst@bold \mst@ltsh

Macros to store the font settings, each math version will store its own records.

\mst@greekfont

1.15c: for use by the LGRgreek and selfGreek options. Defined as an `\edef` in order to be able
to set-up once and for all the Greek at the time of \usepackage. Modifiable in the preamble via \MTgreekfont{(font_name)}\Mathastext.

\edef\mst@greekfont{\familydefault}

Package options 2011/03/09: 1.13 introduces the option LGR\textgreek and systematic use of \if... conditionals, for better readability (by myself) of the code.

1.3x of 2022/11/03 adds ncccomma, binarysemicolon and frenchmath* options.
1.3za adds LGR\textgreek+ and LGR\textgreekfs+.
1.3zb adds decimalcomma and modifies frenchmath* to use it. And provides frenchmath+ as an alias to former frenchmath*. Consecutive to the change at 2.7 of frenchmath which replaced ncccomma by decimalcomma and broke the compatibility recipe explained in subsubsection 1.5.6.

\newif\ifmst@italic
\newif\ifmst@frenchmath
\newif\ifmst@ncccomma
\newif\ifmst@decimalcomma
\newif\ifmst@binarysemicolon
\newif\ifmst@binarysemicolon
\newif\ifmst@endash\mst@endashtrue
\DeclareOption{endash}{\mst@endashtrue}
\DeclareOption{noendash}{\mst@endashfalse}
\newif\ifmst@emdash
\DeclareOption{emdash}{\mst@emdashtrue\mst@endashfalse}
\newif\ifmst@alldelims\mst@alldelimstrue
\edef\mst@tmp{\encodingdefault}\ifx\mst@oti\mst@tmp\else\mst@alldelimstrue\fi
\DeclareOption{alldelims}{\mst@alldelimstrue}
\DeclareOption{nolessnomore}{\mst@alldelimsfalse}
\newif\ifmst@nosmalldelims
\DeclareOption{nosmalldelims}{\mst@nosmalldelimstrue}
\newif\ifmst@noplus
\DeclareOption{noplus}{\mst@noplustrue}
\newif\ifmst@nominus
\DeclareOption{nominus}{\mst@nominustrue}
\newif\ifmst@noparen
\DeclareOption{noparenthesis}{\mst@noparentrue}
\newif\ifmst@nopunct
\DeclareOption{nopunctuation}{\mst@nopuncttrue}
\newif\ifmst@noequal
\DeclareOption{noequal}{\mst@noequaltrue}
\newif\ifmst@noexclam
\DeclareOption{noexclam}{\mst@noexclamtrue}
\newif\ifmst@asterisk
\DeclareOption{noasterisk}{\PackageWarningNoLine{mathastext}{option `noasterisk` is deprecated.\~\~\~J\space\space\space
Check the documentation}}
\DeclareOption{asterisk}{\mst@asterisktrue}
\newif\ifmst@nospecials
\DeclareOption{nospecials}{\mst@nospecialstrue}
\newif\ifmst@basic % 1.3 to avoid unnecessary patch of amsmath \newmcodes@
\DeclareOption{basic}{\mst@basictrue
\ExecuteOptions{noparenthesis,nopunctuation,\
noequal,noexclam,nospecials,nolessnomore}}
\newif\ifmst@nohbar
\DeclareOption{nohbar}{\mst@nohbartrue}
\newif\ifmst@nodigits
\DeclareOption{nodigits}{\mst@nodigitstrue}
\newif\ifmst@defaultimath
\DeclareOption{defaultimath}{\mst@defaultimathtrue}
\newif\ifmst@mathaccents
\DeclareOption{mathaccents}{\mst@mathaccentstrue}
\newif\ifmst@unimathaccents % 1.3u
\DeclareOption{unimathaccents}{\mst@mathaccentstrue\mst@unimathaccentstrue}
\newif\ifmst@needsymbol
\newif\ifmst@symboldelimiters
\DeclareOption{symboldelimiters}{\mst@needsymboltrue\mst@symboldelimiterstrue}
\newif\ifmst@symboldigits
\DeclareOption{symboldigits}{\mst@needsymboltrue\mst@symboldigitstrue}
\DeclareOption{symbolgreek}{\mst@needsymboltrue\mst@symbolgreektrue
\mst@customgreektrue}
\newif\ifmst@symbolre
\DeclareOption{symbolre}{\mst@needsymboltrue\mst@symbolretrue}
\newif\ifmst@symbolmisc
\DeclareOption{symbolmisc}{\mst@needsymboltrue\mst@symbolmisctrue}
\DeclareOption{symbol}{\ExecuteOptions{symbolgreek,symbolmisc,symbolre}}
\DeclareOption{symbolmax}{\ExecuteOptions{symbol,symboldelimiters}}
\newif\ifmst@needeuler
\newif\ifmst@eulerdigits
\DeclareOption{eulerdigits}{\mst@needeulertrue\mst@eulerdigitstrue}
\DeclareOption{eulergreek}{\mst@needeulertrue\mst@eulergreektrue
\mst@customgreektrue}
\newif\ifmst@selfGreek
\DeclareOption{selfGreek}{\mst@selfGreektrue\mst@customgreektrue}
\DeclareOption{selfGreeks}{\mst@selfGreektrue\mst@customgreektrue
\mst@customgreektrue}
Starting with 1.15f the meaning of the ‘defaultxx’ options has changed. They now prevent \textbf{mathastext} from defining additional alphabets rather than prevent it from identifying the ‘mathxx’ with the new ‘Mathxx’. The ‘Mathnormal’ and ‘Mathrm’ alphabet commands are always created as they are SymbolFontAlphabets.

This was again changed at 1.3za. The additional alphabets are always declared, the options only prevent mapping the existing ‘mathxx’ to the new ‘Mathxx’. This may be breaking change if people used these options because they had a need for the \textbf{Mathbf} etc... names.

Here and elsewhere 1.3za has removed an \textbf{ifmst@nonormalbold} conditional.

\textbf{mathastext} considers the default script and especially scriptscript sizes to be far too small, and it will modify them. An option maintains the default.

1.15: the subdued option.
1.3q: the unicode option. Thanks to Tobias BRINK for suggesting its incorporation. The parsing of \CurrentOption does not seek any robustness, it just does its job if the option is used correctly.

\def\mst@unicodeminus {2013}
\def\mst@checkoption #1unicodeminus#2\mst@#3\mst@@ {
  \ifx\#3\%
    \PackageWarningNoLine{mathastext}{Unknown option `\CurrentOption\string'}\else
  \expandafter\def\expandafter\mst@unicodeminus\expandafter{\@secondoftwo#2}\fi
}\DeclareOption*{
  \expandafter\mst@checkoption\CurrentOption\mst@unicodeminus\mst@\mst@@}
\ProcessOptions\relax
\mst@DeclareMathAccent

I somehow missed realizing \LaTeX 2019–10–01 if used together with amsmath made repeated usage of \DeclareMathAccent trigger an error: https://github.com/latex3/latex2e/issues/216. This broke usage of \mathastext macro in preamble.

1.3w works around this via \mst@DeclareMathAccent. And other changes were made in mathastext code to cope with these complications around robustness.

\def\mst@normalversionname{normal}\def\mst@boldversionname{bold}
\exists \mst@exists@skip \forall \mst@forall@skip
\MTnormalexists \MTexistsdoesskip \MTnormalforall \MTforalldoesskip

1.15e 2012/10/21: math skip/glue after \exists and \forall, this is useful with upright letters in math mode. Each math version has its own user defined values for the skips, stored as macros. The redefinitions of $\exists$ and $\forall$ are done only at the end of the package as the symbol option will also want to redefine these math symbols.

The subdued option (later and only for the normal and bold math version) and the italic option (here) set to zero the package default skips. With 1.2 the skips can be modified on the fly in the document, they are not necessarily set in the preamble once and for all for each math version.

1.3j adds \MTnormalexists, \MTexistsdoesskip, \MTnormalforall, \MTforalldoesskip. Earlier to 1.3j, \let\mst@exists@original\exists was done at End of Package, now it is done at Begin Document, and same for \forall. We pay attention that use of \MTnormalexists etc… inside the preamble does not create self-let’s.

Also subdued mode will do \MTnormalexists, \MTnormalforall (earlier than 1.3j, it only set the muskips to 0mu.) Same when using \MTversion{normal}, if subdued.

For some (random, legacy) reason, the handling of $\exists$ and $\forall$ is part of the things not included inside \everymath/\everydisplay.
The \texttt{mathastext}-defined \texttt{\exists} and \texttt{\forall} are created \texttt{\protected}. We feel this matches better with their default definition as \texttt{\mathchardef tokens} than dealing with $\LaTeX2e$ robust macros. Besides, the coding is simpler.

\begin{verbatim}
200 \newmuskip\mst@exists@muskip \% v 1.15e
201 \newmuskip\mst@forall@muskip
202 \def\mst@exists@skip{1mu}
203 \def\mst@forall@skip{.6667mu}
204 \ifmst@italic\ifmst@frenchmath\else
205 \def\mst@exists@skip{0mu}
206 \def\mst@forall@skip{0mu}
207 \def\mst@prime@skip {0mu}
208 \fi\fi
209 \protected\def\mst@exists{\mst@exists@original\mskip\mst@exists@muskip}
210 \protected\def\mst@forall{\mst@forall@original\mskip\mst@forall@muskip}
211 \AtBeginDocument{%
212 \let\mst@exists@original\exists
213 \let\mst@forall@original\forall
214 \def\MTnormalexists {\let\exists\mst@exists@original}
215 \def\MTexistsdoesskip {\let\exists\mst@exists}
216 \def\MTnormalforall {\let\forall\mst@forall@original}
217 \def\MTforalldoesskip {\let\forall\mst@forall}
218 \ifmst@subdued
219 \else
220 \MTexistsdoesskip
221 \MTforalldoesskip
222 \fi
223 }%
224 \newcommand*\MTnormalexists {\AtBeginDocument {\MTnormalexists }}
225 \newcommand*\MTexistsdoesskip {\AtBeginDocument {\MTexistsdoesskip }}
226 \newcommand*\MTnormalforall {\AtBeginDocument {\MTnormalforall }}
227 \newcommand*\MTforalldoesskip {\AtBeginDocument {\MTforalldoesskip }}
\end{verbatim}

The document body starts in the normal math version, whether or not \texttt{\mathastext} command as been used in the preamble (which either re-defines the normal/bold math version or defines another one in case of optional argument), and in case of \texttt{subdued} option should use the standard $\forall$ and $\exists$.

\begin{verbatim}
228 \ifmst@subdued
229 \else
230 \MTexistsdoesskip
231 \MTforalldoesskip
232 \fi
233 }%
234 \newcommand*\MTnormalexists {\AtBeginDocument {\MTnormalexists }}
235 \newcommand*\MTexistsdoesskip {\AtBeginDocument {\MTexistsdoesskip }}
236 \newcommand*\MTnormalforall {\AtBeginDocument {\MTnormalforall }}
237 \newcommand*\MTforalldoesskip {\AtBeginDocument {\MTforalldoesskip }}
\end{verbatim}

\texttt{\prime} \texttt{\mst@prime@skip} \texttt{\active@math@prime} \texttt{\MTnormalprime} \texttt{\MTprimedoesskip}

1.2 2012/12/17: math skip/glue before the $\prime$ glyph. This is useful with the default CM glyph and upright letters (in contrast the prime from \texttt{txfonts} works fine with upright letters).

For this we replace the $\LaTeX$ kernel \texttt{\active@math@prime} with our own skip-enhanced version \texttt{\mst@active@math@prime}.

\begin{verbatim}
238 \ifc@tacode{'}\texttt{=\active \global\let\\mst@active@math@prime}
239 \begin{quote}
240 is awfully wrong when the right quote is made active at begin document by some other package (as happens with \texttt{babel} for some languages). So \texttt{mathastext} treats now the right quote with the same method as applied to the other characters it makes mathematically active. This uses the macro \texttt{\mst@mathactivate} which is defined later in the package.

Babel does \texttt{\let\prim@s\bbl@prim@s} when \texttt{'} is made active via its services (the czech and slovak languages also store the initial version of \texttt{\prim@s}, else the quote would not work correctly when being again of \texttt{catcode 12}), and it doesn’t matter if \texttt{mathastext} is loaded before or
\end{verbatim}

75
after this happens, as the \mst@mathactivate does its job only as part of the \everymath and \everydisplay token lists.

1.2e being paranoid, we take precautions against a possibly catcode active right quote at the time of loading mathastext.

1.3i adds \MTactiveprime. 1.3j renames it to \MTprimedoesskip. Besides, it makes use in the preamble of \MTnormalprime or \MTprimedoesskip.

228 \newmuskip\mst@prime@muskip \% v 1.2
229 \def\mst@prime@skip{.5mu}
230 \ifmst@italic\ifmst@frenchmath\else\def\mst@prime@skip{0mu}\fi\fi
231 \def\mst@active@math@prime{\sp\bgroup\mskip\mst@prime@muskip\prim@s}
232 \catcode`\'=12
233 \gdef\mst@modifyprime{\mst@mathactivate\mst@active@math@prime}
234 \newcommand*\MTnormalprime \{\let\mst@modifyprime\@empty \}
235 \newcommand*\MTprimedoesskip \{\let\mst@modifyprime\mst@modifyprime\}
236 \ifmst@subdued
237 \MTnormalprime
238 \else
239 \MTprimedoesskip
240 \fi
241 \AtBeginDocument{\%
242 \everymath\expandafter
243 \{\the\everymath \mst@modifyprime \MTnormalprime\%
244 \everydisplay\expandafter
245 \{\the\everydisplay \mst@modifyprime \MTnormalprime\%
246 }

\MTexistsskip \MTforallskip \MTprimeskip

1.15e: These user macros set up the amount of muglue after \exists or \forall. The normal and bold math versions inherit the same skips; these skips are set to zero in case of the subdued, or the italic option. Each command \Mathastext[\{version_name\}] stores the current values in the definition of the math version.

1.2: \MTprimeskip added, the silly \onlypreamble are removed and the macros are modified to have immediate effect in the document, independently of their possible use in the preamble for the math versions to store values.

Note (september 2013): the names were badly chosen; \MTsetprimeskip for example would have been a better choice.

247 \newcommand*\MTexistsskip[1]\{\edef\mst@exists@skip{#1}\%
248 \mst@exists@muskip\mst@exists@skip\relax\}
249 \newcommand*\MTforallskip[1]\{\edef\mst@forall@skip{#1}\%
250 \mst@forall@muskip\mst@forall@skip\relax\}
251 \newcommand*\MTprimeskip[1]\{\edef\mst@prime@skip{#1}\%
252 \mst@prime@muskip\mst@prime@skip\relax\}
253 \let\Mathastextristsskip\MTexistsskip
254 \let\Mathastextristsskip\MTforallskip
255 \let\Mathastextristsskip\MTforallskip
256 \let\Mathastextristsskip\MTforallskip
257 \let\Mathastextristsskip\MTforallskip
258 \let\Mathastextristsskip\MTforallskip
259 \let\Mathastextristsskip\MTforallskip
260 \let\Mathastextristsskip\MTforallskip
261 \let\Mathastextristsskip\MTforallskip
262 \let\Mathastextristsskip\MTforallskip
263 \let\Mathastextristsskip\MTforallskip
264 \let\Mathastextristsskip\MTforallskip
265 \let\Mathastextristsskip\MTforallskip
266 \let\Mathastextristsskip\MTforallskip
267 \let\Mathastextristsskip\MTforallskip
268 \let\Mathastextristsskip\MTforallskip
269 \let\Mathastextristsskip\MTforallskip
270 \let\Mathastextristsskip\MTforallskip
271 \let\Mathastextristsskip\MTforallskip
272 \let\Mathastextristsskip\MTforallskip
273 \let\Mathastextristsskip\MTforallskip
274 \let\Mathastextristsskip\MTforallskip
275 \let\Mathastextristsskip\MTforallskip
276 \let\Mathastextristsskip\MTforallskip
277 \let\Mathastextristsskip\MTforallskip
278 \let\Mathastextristsskip\MTforallskip
279 \let\Mathastextristsskip\MTforallskip
280 \let\Mathastextristsskip\MTforallskip
281 \let\Mathastextristsskip\MTforallskip
282 \let\Mathastextristsskip\MTforallskip
283 \let\Mathastextristsskip\MTforallskip
284 \let\Mathastextristsskip\MTforallskip
285 \let\Mathastextristsskip\MTforallskip
286 \let\Mathastextristsskip\MTforallskip
287 \let\Mathastextristsskip\MTforallskip
288 \let\Mathastextristsskip\MTforallskip
289 \let\Mathastextristsskip\MTforallskip
2012/12/31: The \texttt{amsmath} macro \texttt{\resetMathstrut@} is not compatible with a mathematically active opening parenthesis: it does

\texttt{\mathchardef\@tempa\mathcode`\relax}

and is made a part of the hook \texttt{every@math@size} inside \texttt{\glb@settings}. This is called from \texttt{\check@mathfonts} which is done in particular in \texttt{\frozen@everymath}, hence before (but wait) what \texttt{mathastext} puts in \texttt{\everymath}. Also, \texttt{\glb@settings} is triggered by \texttt{\mathversion} which must be done outside of math mode.

Alas, with things such as \$...\hbox{...$..$..}...$ \texttt{mathastext} will have already made the parenthesis (mathematically) active. And \texttt{\boldsymbol} from \texttt{amsbsy} disables the \texttt{@nomath} switch and executes \texttt{\mathversion{bold}} directly in math mode. So we have a problem with \texttt{\resetMathstrut@}.

\texttt{lualatex-math} replaces \texttt{\resetMathstrut@} with its own version (which also looks at ) and no error is signaled when \texttt{mathastext} has done \texttt{\mathcode}=('8000, but the \texttt{\Mathstrutbox@} created by \texttt{mathastext} is then wrong.

The replacement macro avoids a potentially math active (\texttt{\resetMathstrut@}). It assumes that there is still some appropriate glyph in slot 40 of \texttt{operators} and it sets the height and depth of \texttt{\Mathstrutbox@} to be large enough to accomodate both this glyph and the one from the mathastext font (both in the current math version). If option \texttt{noparenthesis} was used, we leave everything untouched.

In 1.3a, 2013/09/04, the modification is done only at the time of \texttt{\MTnonlettersobeymathxx}. It is canceled by \texttt{\MTnonlettersdonotobeymathxx}. So the code has been moved to these macros and we just store at the begin document the then meaning of \texttt{\resetMathstrut@}, and check also if \texttt{\MTnonlettersobeymathxx} has been invoked in the preamble.

1.3f 2015/09/12 issues only an Info message not a Warning, as I am becoming aware from another context (etoc) that Warnings are stressful to users, in some integrated environments for editing and compiling \texttt{\LaTeX} source files.

\begin{verbatim}
259 \ifmst@noparen\else
260 \AtBeginDocument{%
261 \@ifundefined{resetMathstrut@}{{ nothing to do, no amsmath
262 }{% amsmath loaded, and possibly patched by things such as lualatex-math
263 \let\mst@savedresetMathstrut@\resetMathstrut@
264 \PackageInfo{mathastext}{current `at begin document\string' meaning of
265 \string\resetMathstrut@\space preserved for for
266 \string\future use\gobble}\%
267 \ifx\mst@the\the \% means that \MTnonlettersobeymathxx was used in preamble
268 \let\mst@the\gobble\MTnonlettersobeymathxx
269 \fi}
270 \fi}
271 \fi
\end{verbatim}

1.2 2012/12/20 does some rather daring \textit{math} activation of ; , ! ? + - = < > ( ) [ ] in math mode to achieve something I wanted to do since a long time: overcome the mutually excluding relation between the variable-family concept and the automatic spacing concept. After loading \texttt{mathastext}, these characters now obey the math alphabets commands but still have the automatic spacing. The use as delimiters for those concerned is also ok.

The activation is done via setting the \texttt{\mathcode} to \texttt{"8000} through the macro \texttt{\mst@mathactivate} which in turn is put into the \texttt{\everymath} and \texttt{\everydisplay} token lists. No character is made active in the sense of the \texttt{\catcode} (the issues with catcode active characters at the entrance of the math mode are discussed later),

\begin{quote}
but the concerned characters will now expand in math mode to \textit{two tokens}.
\end{quote}
1.2c 2012/12/31: hence, this current implementation puts constraints on the input: $x^{-?}$ or $x\backslash\mathrel{?}y$ now create errors. They must be input $x^{?-?}$, respectively $x\mathrel{?}y$.

The disactivating macro \MTnonlettersdonotobeymathxx is made the default.

The mechanism is (even more) off by default for \{ and \} as this is not compatible with their use as delimiters (\lbrace and \rbrace should be used instead) but it can be activated for them too.

\mst@mathactivate 1.2b 2012/12/30: there were bad oversights in the 1.2 code for \mst@mathactivate related to the possibility for some characters to have been made active (in the sense of the catcode) elsewhere (something which often is done by language definition files of the babel system). The code from v1.2b tried to provide correct behavior using a prefix called \mst@fork (its definition and its use has since been modified) which let the active character expand to the mathtext re-definition only in math mode and only if \protect was \@typeset\protect. This indeed took care of situations such as $\hbox{?}$ with an active ? or $\label{eq:1}$ with an active : (assuming for the latter that things would have worked ok before the twiddling by mathtext).

1.2e 2013/01/09: alas $\ref{eq:1}$ still was a problem. Indeed in that case the mathtext prefix had no means to know it was inside a \ref so it made the character expand to its mathtext redefinition, which is not acceptable inside a \csname...\endcsname. What happens with Babel is that it patches things such as \ref, \newlabel,... we can test the \if@safe@actives flag to detect it in that case, but this is Babel specific. After having thought hard about this I see no general solution except patching all macros such as \ref...(in an imitation of what Babel does). So the final decision is to not do anything when the character is catcode active except it it seems that Babel is behind the scenes.

Incidently, Babel and TikZ are buggy with characters which are mathcode actives. For example the combination of [french]{babel} and mathtools with its centercolon turns $:$ into an infinite loop !

In the case of Babel the reason is that, generally (but not always, the right quote ’ is an exception), the \normal@char〈char〉 fall-back is \string〈char〉. But this is wrong if the mathcode is 32768! The fall-back becomes the default if the user switches to a language where 〈char〉 is ‘normal’ and then an infinite loop arises.

As a further example (I am not familiar with other languages from the Babel system) with frenchb the active !?: expand in math mode to \string! or ? or ; or :. This creates an infinite loop if the mathcode is 32768.

For the special case of the right quote ‘ when it is made active by Babel, its fall-back does not invoke \string‘ so being still of mathcode 32768 is not a problem.

I have posted online how Babel should possibly modify its definitions and I use this here. I simplify a bit my proposed replacement of \normal@char〈char〉 as the check for \protect is superfluous, I think, having been done already at the level of the Babel prefix.

Replacing \user@active〈char〉 is indeed not enough, and \normal@char〈char〉 also must be changed, because when the user switches back to a language where the character is ‘normal’ it remains catcode active. The crucial thing is the test of \if@safe@actives in the replacement of the \normal@char〈char〉, besides of course the test for math mode in both replacements.

When the character is not catcode active, then mathtext uses the math activation method. As the mathcode is not looked at in \edef, \write or inside \csname...\endcsname nothing special needs to be done, I think, in terms of protection against premature expansion. (I did not know that initially).

So, to recapitulate, mathtext will use the mechanism of the active mathcode if the character is not catcode active, and in the opposite case will do something only in the context of Babel, modifying directly its \user@active〈char〉 and its \normal@char〈char〉 macros and
it does NOT then set the mathcode to 32768!! rather it makes sure the character is not mathematically active.

As 1.2e is a bit paranoid it takes precautions against the possibility of characters it treats being active at the time of its loading. Excepted from the scope of the paranoia are the latin letters (that would be crazy!) and also *, " and the left quote `.  

1.2f 2013/01/21 with earlier versions (*) it was important not to do twice the business of `\mst@mathactivate` (think `\hbox{$\hbox{??}$}`), so I used (this was a bit wasteful) some sort of boolean macro for each character. But now that there are the `\mst@the..` prefixes, let’s just use them! (don’t know why I did not think of that earlier; perhaps I had in mind some more general character per character customization initially, which I just dropped.) (*) it is still important to not do twice the thing when the character is active, in which case the babel macros are patched.

As an aside, `\hbox{\catcode` ?\active $\hbox{??}$}` for an ? which was unactive at the first $ will just make `mathastext` overwrite the definition (assumed here to have been done earlier) of an active ?, but the result is that the inner ? can not be used in `\label` or `\ref`. So testing for active characters should be done always... many things should be done always... I leave as is.

1.3i 2016/01/06 removes a spurious end of line space in `\mst@mathactivate` (did not show as anyhow done in math mode).

\MTmath-activeletters 1.2b 2012/12/28 now that we understand the great advantages of "8000 we do it also for all letters a-z and A-Z to insert automatically the italic corrections. See the discussion in the user manual. Ironically I wrote the code initially for the italic option only to realize later it was more suitable to using an upright text font in math mode! So this mathematical activation of the letters is not done if the font shape is detected to be it or sl; to bypass this the command `\MTicinmath` is provided.

1.2e 2013/01/10 corrects a bad oversight of 1.2b in `\mst@mathactivate` which made the reproduction of the user manual illustrations with `$f_i^i$` impossible. As `\mst@mathactivate` was originally used also to get the non-letters obey math alphabet while maintaining the TeX spacings, it added no extra braces. The braces should however be added for expansion of math active letters, in order of things like $x^y$ to work as expected. (the group braces do not prevent ligatures when the letters are arguments to the math alphabet commands, the added macros `\mst@itcorr` and `\mst@before<letter>` expanding to nothing).

Added note 2016/01/08: it should be explicitly said that the extra `{..}` in `\mst@mathactivate` for letters end up creating `\hbox`es around each letter with its extra skips and explicit italic correction, when present. These skips are thus set at natural width and do not add any break point.

\MTmath-standardletters 1.3 2013/09/02 extends the use of mathematically active letters to allow the user to specify muglue before and after the letter itself (see `\MTsetmathskips`, below). Mathematically active letters were previously used only to add the italic correction; the math activation has now been separated and put in `\MTmathactiveletters`. There is also `\MTmathactiveLetters` to allow math activation only for the uppercase letters. To cancel the (now default, even with option italic) math activation of letters, there is `\MTmathstandardletters`. Version 1.3a removes some silly `\string`s from the code, which prevented to pass macros as first argument to the command.

\MTnonletters-obeymathxx \MTnonletters-donotobeymathxx \resetMathstrut@ These macros are modified in version 1.3a 2013/09/04 in order to cleverly adjust, or not, the amsmath `\resetMathstrut@`. When used in the preamble, they just modify `\mst@the`. And there is code at begin document to check the status there of `\mst@the` and if its meaning is
\the, then $\MTnonlettersobeymathxx$ is activated again to do the patch. When used in the body they adjust $\resetMathstrut@$.

Notice that the saved meaning is the one at begin document (thus, possibly patched by lualatex-math — not anymore since 1.5 of March 2016, as amsmath.sty now maintained by LaTeX team has modified $\resetMathStrut@$ to make it compatible to Unicode engines) but modifications done after that would not be seen in $\mst@savedresetMathstrut@$.

The new version of $\resetMathStrut@$ from LaTeX team release 2016/03/03 v2.15a of amsmath.sty is still not compatible with a math active opening parenthesis. Hence my patch here is still needed.

At 1.3u $\MTnonlettersobeymathxx$ and $\MTeasynonlettersobeymathxx$ are made no-ops under subdued mode. This fixes some bug if for example the former was used in preamble or immediately after $\begin{document}$ making the minus sign math active although the mathastext action was supposedly subdued. Similarly $\MTmathactiveletters$ is now a no-op if issued under subdued mode in the normal or bold math versions.

\begin{verbatim}
272 \newtoks\mst@do@nonletters
273 \newtoks\mst@do@easynonletters
274 \newtoks\mst@do@az
275 \newtoks\mst@do@AZ
276 \let\mst@the\@gobble
277 \newcommand*\MTnonlettersdonotobeymathxx{%
278   \ifx\mst@the\the\@gobble
279     \else
280       \@ifundefined{mst@savedresetMathstrut@}{}{%
281         \PackageInfo{mathastext}{restoring for this group or environment
282             the original\MessageBreak
283             amsmath \protect\resetMathstrut@}%
284         \let\resetMathstrut@\mst@savedresetMathstrut@%
285       \fi
286   \fi\mst@the\@gobble
287  }%
1.3u adds this check that we are not in a subdued normal or bold math version. No need for expandable coding.
288 \def\mst@OnlyIfNotSubdued#1{%
289   \ifmst@subdued
290     \ifx\math@version\mst@normalversionname
291       \else
292         \ifx\math@version\mst@boldversionname
293           \else
294             #1%
295           \fi
296         \fi
297       \else
298         #1%
299       \fi
300   \fi
301 \def\mst@nonlettersobeymathxx{%
302   \ifx\mst@the\the\%  80
\end{verbatim}

\ifundefined{mst@savedresetMathstrut@}{%\ifmst@symboldelimiters\def\resetMathstrut@{%\setbox\z@\hbox{\the\textfont\symmtpsymbol\char40\the\textfont\symmtoperatorfont\char40\the\textfont\symoperators\char40}\ht\Mathstrutbox@\ht\z@ \dp\Mathstrutbox@\dp\z@}%\else\def\resetMathstrut@{%\setbox\z@\hbox{\the\textfont\symmtoperatorfont\char40\the\textfont\symoperators\char40}\ht\Mathstrutbox@\ht\z@ \dp\Mathstrutbox@\dp\z@}%\fi\PackageInfo{mathastext}{\string\resetMathstrut@ from amsmath replaced for this\MessageBreak group or environment}}%\fi\let\mst@the\the}\newcommand*{\MTnonlettersobeymathxx}{\mst@OnlyIfNotSubdued\mst@nonlettersobeymathxx}\newcommand*{\MTeasynonlettersdonotobeymathxx}{\let\mst@theeasy\@gobble}\newcommand*{\MTeasynonlettersobeymathxx}{\let\mst@theeasy\the}\newcommand*{\MTeasynonlettersobeymathxx}{\mst@OnlyIfNotSubdued\mst@easynonlettersobeymathxx}\MTnonlettersobeymathxx \MTTeasynonlettersobeymathxx \MTeasynonlettersobeymathxx % no-op here if subdued mode\def\mst@mathactiveletters{\let\mst@thef\the \let\mst@theF\the}\newcommand*{\MTmathactiveletters}{\mst@OnlyIfNotSubdued\mst@mathactiveletters}\MTmathactiveletters % no-op here if subdued mode\def\mst@mathactiveLetters{\let\mst@theF\the}\newcommand*{\MTmathactiveLetters}{\mst@OnlyIfNotSubdued\mst@mathactiveLetters}\MTmathactiveLetters \MTmathstandardletters{\let\mst@thef\@gobble \let\mst@theF\@gobble}\MTmathstandardletters\MTicinmath\MTICinmath\MTnoicinmath\MTicalsoinmathxx\MTnoicinmath\MTicinmath\MTicalsoinmathxx\MTnoicinmath\MTicinmath\MTnoicinmath\MTicalsoinmathxx\MTnoicinmath\MTicinmath\MTnoicinmath\MTicalsoinmathxx\MTnoicinmath\MTicinmath\MTnoicinmath\MTicalsoinmathxx\MTnoicinmath

\MTicinmath\MTnoicinmath can also be used from inside math mode.
\MTICinmath\MTnoicinmath\MTicalsoinmathxx can also be used inside math mode, to revert an earlier \MTnoicinmath from inside the same math group: the math mode had to be entered with the math activation of letters allowed.

1.3i 2016/01/06: For some reason which I have now forgotten I did until now:

% \def\mst@itcorr{\ifnum\fam=\m@ne/\else\ifnum\fam=\symmtletterfont/\fi\fi}%
% hence italic corrections were also applied inside \mathnormal (for upright fonts; \mathnormalbold math alphabet was not treated like \mathnormal). I now drop this to be more in sync
with the handling of the extra skips around letters. Everything gets suppressed inside all math alphabets, allowing ligatures, even for \textit{\textnormal{}}.

\newcommand{\MTicinmath}{% \MTmathactiveletters \def\mst@itcorr{\ifnum\fam=\m@ne/\fi} \let\mst@ITcorr\mst@itcorr}
\newcommand{\MTICinmath}{% \MTmathactiveLetters \def\mst@ITcorr{\ifnum\fam=\m@ne/\fi}}
\newcommand{\MTnoicinmath}{\let\mst@itcorr\@empty\let\mst@ITcorr\@empty}
\newcommand{\MTnoICinmath}{\let\mst@ITcorr\@empty}
\newcommand{\MTicalsoinmathxx}{% \ifx\mst@itcorr\@empty\else\def\mst@itcorr{/}\fi \ifx\mst@ITcorr\@empty\else\def\mst@ITcorr{/}\fi}

\MTsetmathskips
\MTunsetmathskips

1.3 2013/09/02: user level command to specify extra spaces in math mode around the letters (only the 7bit a,b,...,z and A,B,...,Z). First parameter is the letter, second is the math skip to be inserted before, and third the skip to be inserted after; for example \texttt{\textbackslash thickmuskip} or explicitly 0.1mu.

For this, letters are made mathematically active. This is now the package default (version 1.2 did this only in the absence of option italic, or more precisely when the font used was not of shape \textit{it} or \textit{sl}). But if \MTsetmathskips has not been used for that letter, the only effect of the math activation is, as in 1.2, to add the italic correction automatically, except when the font shape is detected to be \textit{it} or \textit{sl}; in these latter cases, although mathematically active, the letter acts in the standard way.

The command \texttt{\MTmathstandardletters} turns off math activation and its effects for all letters.

Ligatures within the argument of a math alphabet command are impeached by skips; so \MTsetmathskips is provided to cancel the skips for one specific letter (f for example).

1.3a 2013/09/04: I strangely had \texttt{\string #1} inside \MTsetmathskips. Phobic of catcode active letters... but with \texttt{\string} one needs some \texttt{\expandafter} to use \MTsetmathskips in an \texttt{\@for} loop for example. It is better to allow the first argument to be a macro or anything expanding to a letter, and to not be paranoid about improbable catcode active letters (the user just has to tame them at the time of the \MTsetmathskip) so I take out these \texttt{\string}'s.

1.3i 2016/01/06: the extra skips are suppressed for the arguments of math alphabet commands. This applies in particular for amsmath's \texttt{\DeclareMathOperator}.

\newcommand{\MTsetmathskips}[3]{% \@namedef{mst@before#1}{\ifnum\fam=\m@ne\mskip#2\relax\fi} \@namedef{mst@after#1}{\ifnum\fam=\m@ne\mskip#3\relax\fi} }
\newcommand{\MTunsetmathskips}[1]{% \@namedef{mst@before#1}{} \@namedef{mst@after#1}{} }

\def{\mst@magic@v #1#2#3#4#5}{{#1#3#4}}

\MTmathactivate
\MTmathstandardletters
\MTsetmathskips
\MTunsetmathskips

Added note 2016/01/06: Notice that the initially \texttt{\relax} tokens \texttt{\mst@[before|after]@<letter>} formed with \texttt{\csname...\endcsname} do not modify TeX's math layout: \texttt{\textbackslash relax f\textbackslash relax} is like \texttt{f} (also for ligatures inside \texttt{\textbackslash mathrm} for example).

\def{\mst@magic@v #1#2#3#4#5}{#1#3#4}
1.15d: the \newmcodes@ amsmath macro causes an error in Unicode engines as soon someone assigns a Unicode mathcode to the minus sign, and then makes a `DeclareMathOperator` declaration. Furthermore it hard-codes the font family 0 as being the one to be used. Moreover just putting the concerned signs -, ; , \, * inside braces emulates enough the behavior (although the tick will give a prime).

1.3: now tests if ‘basic’ option was used.

1.3d: I should re-examine the situation with \newmcodes@. In the meantime its relaxification will not be done if lualatex-math is loaded. And the whole thing is put at begin document.

1.3m: lualatex-math 1.5 n’a pas modifié son traitement de \newmcodes@ mais par contre a supprimé le patch de \resetMathstrut@. Mais la date de release est restée à 2015/09/22 (date de 1.4a) au lieu de quelque chose comme 2016/03/13 (date pour l’annonce sur CTAN). Il faudra suivre l’évolution future de amsmath.sty maintenant assurée par D.C.

1.3n 2016/04/22: there is no more a patch of \newmcodes@ by lualatex-math 1.6 (2016/04/16), as amsmath 2016/03/10 v2.15b has now a version compatible with LualatexX.

My very radical \let\newmcodes@\relax was only a temporary measure I adopted for lack of time on October 13, 2012, and apart from avoiding to do that in case lualatex-math was detected, I never came back... finally I handle it myself for 1.3n. The remaining problem of this macro (now that it does not anymore crash lualatex or vice versa) is that (also with amsmath version 2016/03/10 v2.15b) it hardcodes the font used. The aim of the macro is to modify the type of spacing affected to symbols ‘, *, -, /, ;, in case they are used in operator names.

- As I don’t want to monopolize a count register only for computations, let’s just be mean if ε-TeX not there.
- mathastext makes (or not, depending on commands issued by the user) these characters math active (the right tick already is), which complicates recovery of former mathcode. We have mathchar type macros, but then the complication is in diverging behaviours of the engines: \numexpr\text@varfam@\text@minus\relax works with LualatexX, not with XeTeX.
• the * must presumably really be the non-lowered text glyph.

• for the – I hesitated but do use the hyphen in the end.

• seems I simply don’t understand what the amsmath code does with `\std@minus`. It is used in `\relbar` and it escapes me why `\newmcodes@` would ever want to redefine it, and more importantly why on earth it tests the mathcode of – for that ? yes, `\std@minus` is defined (at begin document) using the mathcode of –, but what’s the connexion to `\newmcodes@` ?? Any way `mathastext` defines `\relbar` with `\mst@minus@sign`. Thus I just drop this conditional.

• things are complicated by the options such as `nominus`, `noparenthesis`.

• the `\newmcodes@` macro is anyhow assuming that if a new math font is used it occupies math groups 0 and 1 !! very bad; fixing it in passing if the character has not been handled by `mathastext` could be envisioned, but that’s not `mathastext`’s job.

• years go by, and I remain as baffled as ever about the story of “more than 16 math families”. I will not test again, but I am pretty sure that `\DeclareMathSymbol` does not work with more than 16 families, thus when I try to be a good boy and use `\mathcode` syntax with `\symmtoperatorfont` I am perhaps doing unnecessary efforts.

• I noticed that LuaLaTeX does not apply the “TeX Ligature” (bad name) regarding the right tick APOSTROPHE being transformed into RIGHT SINGLE QUOTATION MARK in math mode, but XeLaTeX does. From the point of view of `mathastext`, the behaviour of XeLaTeX is the coherent one. It appears that LuaLaTeX use in math mode of a text font does not obey the set features.I opened a ticket at https://github.com/wspr/fontspec/issues/238, but as usual it is hard to figure out the best place where to report font matters. *This item might be obsolete – not checked (1.3q).*

• Some hesitation about what to do under option `symboldelimiters`. I temporarily used `\symmtpsymbol`, except for the right quote and for the hyphen, but finally I drop that and use `\symmtoperatorfont` always. (after testing how it looked like).

All in all this is a great deal of trouble and I understand I postponed back in 2012! I spent some hours on this small thing, with consequent testing and for example this TeX Ligature issue with Unicode engines.

Since 1.3v we require e-TEx extensions, so a test for `\numexpr` has been dropped here.

```
428 \ifmst@basic
429 \else
430 \ifmst@XeOrLua
431 \AtBeginDocument {%
432 \exif\newmcodes@%@undefined\else
433 \edef\mst@newmcodes@{%
434 \mst@Umathcode `\noexpand` 0 \symmtoperatorfont 39\relax
435 \ifmst@asterisk
436 \mst@Umathcode `\noexpand`* 0 \symmtoperatorfont 42\relax
437 \else\mathcode `\noexpand`* 42
438 \fi
439 \ifmst@nopunct\mathcode `\noexpand`."613A \mathcode `\noexpand`: "603A
440 \else
441 \mst@Umathcode `\noexpand`. 6 \symmtoperatorfont 46\relax
442 \mst@Umathcode `\noexpand`: 6 \symmtoperatorfont 58\relax
```
mtoperatorfont Declaration of the current default font as our math font. The characteristics of the used font can be changed by a user call to the macros \Mathastext or \Mathastextwilluse, which will be defined next. We will also make one internal call to \Mathastext to set up the normal and bold math versions, so we will also employ \SetSymbolFont later.

\newcommand{\MTresetnewmcodes}{\ifx\mst@originalnewmcodes@undefined\else
\let\newmcodes@\mst@originalnewmcodes@\fi}
\newcommand{\MTcustomizenewmcodes}{\ifx\mst@originalnewmcodes@undefined\else
\let\newmcodes@\mst@originalnewmcodes@\fi}

\DeclareSymbolFont{mtoperatorfont}{\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}
\operator@font We modify this LATEX internal variable in order for the predefined \cos, \sin, etc... to be typeset with the mathastext font. This will also work for things declared through the amsmath package command \DeclareMathOperator. The alternative would have been to redefine the ‘operators’
Math Symbol Font. Obviously people who expect that \texttt{\textbackslash operator\textbackslash font} will always refer to the ‘operators’ math font might be in for a surprise... well, we’ll see.

\MTmathoperators\-obeymathxx
\MTmathoperators\-donot\-obeymathxx

1.2: rather than just replacing \texttt{\textbackslash symoperators} by \texttt{\textbackslash symmtoperatorfont} I add a modification which makes the declared operator names sensitive to the math alphabets... ouh le vilain!

483 \newcommand*{\MTmathoperatorsobeymathxx}{\def\operator@font{\mathgroup\ifnum\fam=\m@ne\symmtoperatorfont\else\fam\fi}}
484 \newcommand*{\MTmathoperatorsdonotobeymathxx}{\def\operator@font{\mathgroup\symmtoperatorfont}}
487 \MTmathoperatorsobeymathxx

mtletterfont

At version 1.1, we add the possibility to mimic the standard behavior, that is to have italic letters and upright digits. Thanks to Tariq \textsc{Perwez} and Kevin \textsc{Klement} who asked for such a feature.

488 \DeclareSymbolFont{mtletterfont}{\mst@enc}{\mst@fam}{\mst@ser}{\mst@ltsh}

\MTfixmathfonts

There is a long-standing issue \url{https://github.com/lualatex/luaotfload/issues/204} on \texttt{\LaTeX} not applying OpenType features in math mode (this impacts \texttt{\url{\textbackslash m}}acro too, as it uses math mode.) \texttt{\LaTeX} has two modes for handling of OpenType fonts, the default in text being to use the \texttt{node} mode, and this mode is non-working in math, thus \texttt{\textbackslash mathastext} needs to force use of \texttt{base} mode. Else one sees old style figures where one does not expect them, or the opposite, depending on the default font feature.

Once we know the cause, the fix is relatively easy. I will go for the \texttt{everymath} way, because I don’t want to dwell at all with the details of \texttt{\LaTeX}'s handling of math fonts, of size changes, of math versions etc... perhaps in the future \texttt{\LaTeX} will fix the issue upstream by modifying \texttt{\textbackslash DeclareSymbolFont} under \texttt{\LaTeX} + \texttt{luaotfload} regime, then the present patch by \texttt{\textbackslash mathastext} will be unneeded. Naturally, here we care only about the two math fonts used by \texttt{\textbackslash mathastext}: \texttt{\textbackslash mtoperatorfont} and \texttt{\textbackslash mtletterfont}.

For the \texttt{\url} situation, I have posted online a patch.

Not all is resolved, as I comment online at \url{https://github.com/lualatex/luaotfload/issues/204#issuecomment-216465680} that with \texttt{\TeX} \texttt{Gyre Termes} for example I can not get simultaneously Old Style and Tabular Figures to work in math mode, although the font name as constructed by my patch (which is like the code below, only simpler as we only have to consider \texttt{\textbackslash textfont0}) is the correct one. Similarly with \texttt{\textbackslash Vollkorn}: I can then not get the two features \texttt{lnum} and \texttt{tnum} to work simultaneously when specified with \texttt{mode=base}. It does work with \texttt{mode=node} but this mode “does not work in math mode.”

Done for 1.3o of 2016/05/03.

1.3p renames the macro to \texttt{\textbackslash MTfixmathfonts} for public access.

87
\Mathnormal
\Mathrm
\Mathbf
\Mathit
\Mathsf
\Mathtt
\Mathnormalbold

We redefine the default normal, rm, bf, it, sf, and tt alphabets, but this will be done via \renewcommand*{\mathrm}{\Mathrm} etc... (not anymore, see comment below).

We follow the standard \LaTeX behavior for \mathbf, which is to pick up the bold series of the roman font (digits and operator names).

We will access (if no option is passed for Greek) the \omicron via \mathnormal. But unfortunately the fourier package with the upright option does not have an upright omicron obtainable by simply typing \mathnormal{o}. So if fourier is loaded we use \mathrm and not \mathnormal.

Actually math alphabet macros are created robust since \LaTeX from 2005, so at 1.3v 2019/09/19 I decided to modify the old \mathastext approach a bit. Indeed with the old approach a \Mathtt in a moving argument translates ultimately into \Mathtt but if for example the new context where it gets expanded is a subdued normal math version, this does not give the same as \Mathtt would have given there. This was a bug: imagine \section{X} issued in a math version, but the TOC is done in subdued normal version; the output in TOC will often differ (fontsize being put aside) from what direct usage of \Mathtt in the TOC would have given. I have no strong preference between the two possibilities (to be as in section title, or to be as if \Mathtt gets executed in TOC and obeys its local regime), but it is a bug if the result is still a third one. Thus I decided to follow \LaTeX2e and that \Mathtt had to remain \Mathtt when moving.

But a math alphabet command such as \Mathtt redefines its unprotected meaning on first use as well as the one of the math version macro, hence a \let\robustmacro{\Mathtt}{\Mathtt} of sorts is no good at all. I thus opted to not hack into the math \LaTeX font support across math versions and to simply use \protected\def in place of obeying strictly \LaTeX2e robustness (except of
course in the subdued math versions as there the math alphabets acquire back their original robust meanings.)

Potential breaking change at 1.3za, the defaultbf etc... options do not prevent the package declaring \mathbf etc... commands.

1.3za defines a \Mathnormalbold and then defines \mathnormalbold in terms of it in place of defining directly \mathnormalbold as a math alphabet. This is in relation to implementation of the LGRgreek+ option. There was some hesitation though to restrict this change to that option only or not.

\let\mst@alph@omicron\mathnormal
\ifpackageloaded{fourier}\{\ifsloped\else\let\mst@alph@omicron\mathrm\fi\}\{\let\mst@alph@omicron\mathnormal\\}
\DeclareSymbolFontAlphabet{\Mathnormal}{mtletterfont}
\DeclareSymbolFontAlphabet{\Mathrm}{mtoperatorfont}
\DeclareMathAlphabet{\Mathnormalbold}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@ltsh}
\protected\def\mathnormalbold{\Mathnormalbold}
\DeclareMathAlphabet{\Mathbf}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@opsh}
\DeclareMathAlphabet{\Mathit}{\mst@enc}{\mst@fam}{\mst@ser}{\itdefault}
\DeclareMathAlphabet{\Mathsf}{\mst@enc}{\sfdefault}{\mst@ser}{\mst@opsh}
\DeclareMathAlphabet{\Mathtt}{\mst@enc}{\ttdefault}{\mst@ser}{\mst@opsh}

The \mathxx macros being \LaTeX2e robust, or course the meanings here are known, and «original» macros are sort of superfluous but well it works.

\let\mst@original@normal\mathnormal
\let\mst@original@rm\mathrm
\let\mst@original@bf\mathbf
\let\mst@original@it\mathit
\let\mst@original@sf\mathsf
\let\mst@original@tt\mathtt
\def\mst@restorealphabets{% for subdued
\let\mathnormal\mst@original@normal
\let\mathrm\mst@original@rm
\let\mathbf\mst@original@bf
\let\mathit\mst@original@it
\let\mathsf\mst@original@sf
\let\mathtt\mst@original@tt
}
\ifmst@greekplus
\mst@mathalph
\fi

\mst@mathalph% The \LaTeX kernel code regarding math fonts is too complex and rigid for there to be a robust and easy way to know when one is in the argument of \mathrm or \mathbf, and the code is spread on various chapters of source2e.pdf, and the comments are often not up-to-date. So I did not try a too complex hack and decided for 1.3za to add a numeric indicator to let Greek letters react to it. It incorporates a space to be self-delimiting in an \ifcase to maintain expandability in numeric context of the to-be-defined Greek control sequences.
\def\mst@mathalph{-1}
\def\mst@setalphabets{%
\protected\def\mst@mathalph{\#1}\{\def\mst@mathalph{\#1}\}\%
\ifmst@defaultnormal\%
\else
\}
\fi
\mst@mathalph
1.14b: We cannot move the \DeclareSymbolFont to the \Mathastext macro because it resets the font family in *all* math versions, and some could have been defined by the user with previous calls to \Mathastext. So we have to have them here. The problem is that at this stage it is impossible to know if we really need (in the case of LGRgreek) two separate shapes for upper and lowercase, and (in the case of selfGreek) a shape distinct from the one used in \mtoperatorfont. So I opted in the end for declaring possibly one too many font. To achieve more economy the only way would be to keep cumulative track of all previously declared math versions and to redeclare appropriately the LGR or self Greek fonts at each call to \Mathastext (with no optional argument): a bit painful, and as I am possibly the sole user in the world of this possibility of multiple math versions with this package. Also the advantage to systematically allocate a font for the selfGreek option is that we can force the use of the OT1 encoding.

First we establish the cumulative effect of the greek related options. 1.15c introduces some possibilities to change the shapes of Greek letters in each math versions, and even the Greek font (in LGR encoding). The commands \Mtgitgreek etc... will be used in-between calls to \Mathastext and re-adjust the shapes. And the command \Mtgreekontfont changes the Greek font family.

Note that \mst@ltsh expands to \shapedefault or \itdefault at this location.

Note added 2022/11/02: using \Mtgitgreek etc... once implies that from then on, for subsequent \mathastext-math versions, the shape of Greek letters will not be kept in sync with the shape and lettershape version parameters, but only react to the configuration decided by these
commands (and italic/frenchmath options).

Note 2022/10/29: for some time \updefault was made into up by \LaTeX (since 2020-02-02 now that I check this out). As a result this triggered Font Warnings in the log about the replacement of up by n.

1.3x refactors completely the handling of Greek letter shapes under the LGRgreek() options (and only under them). Under these options we don’t use one font for lowercase Greek and another one for uppercase Greek (some above code comments have not been updated) but one math font mtgreekit for italic Greek and one math font mtgreekup for upright Greek. What ‘italic’ and ‘upright’ mean is decided by the expansion of \MTgreekitdefault and \MTgreekupdefault, which give respectively it and n per default.

If no itgreek et al. options or \MTitgreek et al. commands have been used, we need to map \mst@ltsh (which was used for lowercase Greek, except under frenchmath option) and \mst@opsht to either ‘italic’ or ‘upright’. This is done by testing if they hold ‘it’ or ‘sl’. If yes we map to ‘italic’ by setting to false an ‘up’ Boolean, if not we leave the ‘up’ Boolean to true.

In order to maintain perfect identical code for non-LGRgreek, the LGRgreek related code is simply added to previously shared constructions. The LGRgreek behaviour will remain identical in most documents, but for example those who used some adventurous ‘sc’ for the main shape (the one used per default for operator names) need to adjust \MTgreekupdefault to be ‘sc’, for the math version being defined, or the default one if this is followed by usage of \Mathastext.

The new LGRgreek-specific commands \MTgreekupdefault and \MTgreekitdefault are the only ones in the package which can possibly be defined previously to loading it. (Perhaps some other macros could be also converted to being modifiable prior to loading \Mathastext, thus avoiding potential need to use \Mathastext at least once after loading the package; to be examined next time — which may be a long time in future!).

Unfortunately the 1.3x/1.3y did some internal renamings here (using \@lgr@ in macro names in place of \@greek@ which were not everywhere followed up, and this broke the selfGreek option. Fixed at 1.3z.

\providecommand*{\MTgreekupdefault}{n}
\providecommand*{\MTgreekitdefault}{it}
\newif\ifmst@greek@lower@up
\newif\ifmst@greek@upper@up
\def\mst@update@greeksh{
  \def\mst@greek@lsh{\mst@ltsh}
  \def\mst@greek@ush{\mst@opsh}
  \ifmst@itgreek
    \def\mst@greek@lsh{\MTgreekitdefault}
    \def\mst@greek@ush{\MTgreekitdefault}
  \fi
  \ifmst@upgreek
    \def\mst@greek@lsh{\MTgreekupdefault}
    \def\mst@greek@ush{\MTgreekupdefault}
  \fi
}
The 1.3x refactoring was done in order to be able to define \alphaup, etc... control sequences (\mathchar's), as well as the italic ones. Formerly two math fonts were created but to be used respectively with lowercase or uppercase Greek. Now we have two fonts indexed by their shape, and we take advantage to create two math alphabets mapping to the two defined symbol fonts mtgreekup and mtgreekit.

1.3za adds \mathgreekupbold and \mathgreekitbold.

---

In case we need the Euler font, we declare it here. It will use uzeur.fd from the eulervm package of Walter Schmidt.

---

mtselfGreekfont
\DeclareSymbolFont{mtselfGreekfont}{OT1}{\mst@fam}{\mst@ser}{\mst@greek@ush}
\fi\fi
\ifcase\mst@greek@select
\or
\def\mst@greek@ush{\MTgreekitdefault}
\mst@greek@upper@upfalse
\or
\def\mst@greek@ush{\MTgreekupdefault}
\mst@greek@upper@uptrue
\fi
\fi
\fi} \mst@update@greeksh
\begin{thebibliography}{92}
\input{mtselfGreeks8}
\end{thebibliography}
\[t\]EX \(\epsilon\) has a strange initial configuration where the capital Greek letters are of type \texttt{mathalpha}, but the lower Greek letters of type \texttt{mathord}, so that \texttt{\textbf{mathbf}} does not act on them, although lowercase Greek letters and Latin letters are from the same font. This is because \texttt{\textbf{mathbf}} is set up to be like a bold version of \texttt{\textbf{mathrm}}, and \texttt{\textbf{mathrm}} uses the ‘operators’ font, by default \texttt{cmr}, where there are NO lowercase greek letters. This set-up is ok for the Capital Greek letters which are together with the Latin letters in both \texttt{cmmi} and \texttt{cmr}.

The package eulervm sets the lowercase Greek letters to be of type \texttt{mathalpha}, the default \texttt{\textbf{mathbf}} and \texttt{\textbf{mathrm}} will act wierdly on them, but a \texttt{\textbf{mathbold}} is defined which will use the bold series of the Euler roman font, it gives something coherent for Latin and Greek lowercase letters, and this is possible because the same font contains upright forms for them all.

Here in \texttt{mathastext}, Latin letters and Greek letters (lower and uppercase) must be (generally) assumed to come from two different fonts, as a result the standard \texttt{\textbf{mathbf}} (and \texttt{\textbf{mathrm}}) will give weird results when used for Greek letters. We could coerce \texttt{\textbf{mathbf}} to do something reasonable (cf \url{http://tug.org/pipermail/texhax/2011-January/016605.html}) but at this time 30-01-2011 09:42:27 CET I decided I would not try to implement it here. I prefer to respect the default things.

I followed the simpler idea of the \texttt{eulervm} package and defined \texttt{\MathEuler} and \texttt{\MathEulerBold} alphabet commands (the \texttt{eulervm} package does this only for the bold font).

\begin{verbatim}
mtsymbol
\MathPSymbol

I did not choose for name \texttt{\MathSymbol} as this may be defined somewhere for another thing. There is no bold for the postscript Symbol font distributed with the \texttt{t}EX \(\epsilon\) \texttt{psnffs} core package.

\pmvec

Definition of a poor man version of the \texttt{\vec} accent

\fouriervec

The glyph is taken from the Fourier font of Michel Bovani. Note: (oct 2012) I should not allocate an entire symbol font just for one glyph! But I have not given any serious thought to what one can do to simulate a math accent without doing such a wasteful thing.

\end{verbatim}
Some public macros to modify our private internals, and we will use them also ourself.

In version 1.1 we add the possibility to have two distinct font shapes for letters and digits. So in fact we could as well have two really unrelated fonts but this is really not the spirit of the package.

Note that using these macros in the preamble allows \Mathastext to set up math versions with a given font for math mode, and at the same time not modifying the \familydefault or \romandefault etc...

At time of 1.3za I considered letting LGRgreeks and selfGreeks support \MTgreekfont and this needed a change to \MTfamily here but I dropped the idea. Too wary of documentation changes.

At (long...) last we now change the font for the letters of the latin alphabet. In version 1.1, Latin letters have their own font (shape).
1.2b initiated the use of mathematically active letters to insert the italic corrections. With version 1.3 the use of math active letters is also for extra muglue added before and after the letters. Use of \texttt{@} for to shorten the code initiated with release 1.3.

\begin{verbatim}
\def\mstDeclareMathLetter #1{\
  \DeclareMathSymbol {#1}{\mathalpha}{mtletterfont}{'#1}\
  \expandafter
  \DeclareMathSymbol \csname mst@#1\endcsname{\mathalpha}{mtletterfont}{'#1}\
  \expandafter\mst@addtodo@az\expandafter #1\csname mst@#1\endcsname
}
\@tfor\mst@tmp:=abcdefghijklmnopqrstuvwxyz\do\
{\expandafter\mstDeclareMathLetter\mst@tmp}
\ifmst@frenchmath \def\mst@font@tbu{mtoperatorfont}\fi
\def\mstDeclareMathLetter #1{%
  \DeclareMathSymbol {#1}{\mathalpha}{\mst@font@tbu}{'#1}\
  \expandafter
  \DeclareMathSymbol \csname mst@#1\endcsname{\mathalpha}{\mst@font@tbu}{'#1}\
  \expandafter\mst@addtodo@AZ\expandafter #1\csname mst@#1\endcsname
}
\@tfor\mst@tmp:=ABCDEFGHIJKLMNOPQRSTUVWXYZ\do\
{\expandafter\mstDeclareMathLetter\mst@tmp}
\let\mstDeclareMathLetter\relax
\ifmst@nodigits\else
\def\mst@font@tbu{mtoperatorfont}\fi
\ifmst@symboldigits \def\mst@font@tbu{mtpsymbol} \fi
\ifmst@eulerdigits \def\mst@font@tbu{mteulervm} \fi
\DeclareMathSymbol{0}{\mathalpha}{\mst@font@tbu}{'0}\
\DeclareMathSymbol{1}{\mathalpha}{\mst@font@tbu}{'1}\
\DeclareMathSymbol{2}{\mathalpha}{\mst@font@tbu}{'2}\
\DeclareMathSymbol{3}{\mathalpha}{\mst@font@tbu}{'3}\
\DeclareMathSymbol{4}{\mathalpha}{\mst@font@tbu}{'4}\
\DeclareMathSymbol{5}{\mathalpha}{\mst@font@tbu}{'5}\
\DeclareMathSymbol{6}{\mathalpha}{\mst@font@tbu}{'6}\
\DeclareMathSymbol{7}{\mathalpha}{\mst@font@tbu}{'7}\
\DeclareMathSymbol{8}{\mathalpha}{\mst@font@tbu}{'8}\
\DeclareMathSymbol{9}{\mathalpha}{\mst@font@tbu}{'9}\
\fi
\end{verbatim}

In version 1.1, we have now separated digits from letters, so paradoxically it is less problematic to give them the \texttt{mathalpha} type.

\begin{verbatim}
\ifmst@symboldelimiters \def\mst@font@tbu{mtpsymbol}\fi
\ifmst@eulerdelimiters \def\mst@font@tbu{mteulervm}\fi
\DeclareMathSymbol{0}{\mathalpha}{\mst@font@tbu}{'0}\
\DeclareMathSymbol{1}{\mathalpha}{\mst@font@tbu}{'1}\
\DeclareMathSymbol{2}{\mathalpha}{\mst@font@tbu}{'2}\
\DeclareMathSymbol{3}{\mathalpha}{\mst@font@tbu}{'3}\
\DeclareMathSymbol{4}{\mathalpha}{\mst@font@tbu}{'4}\
\DeclareMathSymbol{5}{\mathalpha}{\mst@font@tbu}{'5}\
\DeclareMathSymbol{6}{\mathalpha}{\mst@font@tbu}{'6}\
\DeclareMathSymbol{7}{\mathalpha}{\mst@font@tbu}{'7}\
\DeclareMathSymbol{8}{\mathalpha}{\mst@font@tbu}{'8}\
\DeclareMathSymbol{9}{\mathalpha}{\mst@font@tbu}{'9}\
\fi
\end{verbatim}

When \texttt{symboldelimiters} is passed as an option, we use the Symbol font for the printable characters other than letters and digits.
1.2 adds the tricks to let non letters/digits obey math alphabets. We have to double the definitions for easy switch on-off of the mechanism, via a token list which is put into \everymath and \everydisplay.

\MTlowerast
\mst@doasterisk
\MTnormalasterisk
\MTactiveasterisk

1.12d The \ast or * is defined in fontmath.ltx as a binary operator from the symbols font. Usually the asterisk from the text font is in a raised position. Previous versions of mathastext did nothing with \ast but strangely defined * to be the one from the text font, with type \mathalpha. The package now leaves by default both * and \ast untouched, and if passed option asterisk replaces both of them with a lowered text asterisk (or the one from the Symbol font), and of type \mathbin. A trick is used to optionally get both * and \ast obey the math alphabets.

The user macro \MTlowerast sets the amount of lowering to be applied to the text asterisk.

nota bene: it is assumed that * is of type other when mathastext is loaded... it should neither be active, nor of type letter!

1.3i adds \MTnormalasterisk and \MTactiveasterisk. They do nothing without option asterisk.

\def\mst@doasterisk{\let\ast\mst@ast\mst@mathactivate*\relax}
\newcommand*\MTnormalasterisk {\let\mst@doasterisk\relax}
\newcommand*\MTactiveasterisk {\let\mst@doasterisk\mst@@doasterisk}
\ifmst@asterisk\mst@infoline{asterisk: \string\ast\space and *}
\AtBeginDocument{\everymath\expandafter{\the\everymath \mst@doasterisk \MTnormalasterisk}}
\everydisplay\expandafter{\the\everydisplay \mst@doasterisk \MTnormalasterisk}
\ifmst@symbolmisc
\def\mst@bin@ast{\mathbin{\mathchoice{\def\mst@doasterisk{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}}
\else
\def\mst@bin@ast{\mathbin{\mathchoice{\def\mst@doasterisk{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}{\let\ast\mst@ast}\mst@mathactivate*\relax}}
\fi
(2011) I renounced to try to do things with all the various dots, they are defined in many
different ways, and there is the amsmath also. Dealing with this issue would mean a lot a time
for a minuscule result. Better to leave the user use the \texttt{mathdots}
package and accept that we
can not avoid the default fonts in that case. So here I just treat.

\[(Dec. 2012) should I reexamine these definitive sounding remarks?\]

1.3x of 2022/11/03 adds support for \texttt{ncccomma} option.

Some non-obvious hack is needed for compatibility with our home-made mechanism of non-
letters obeying math alphabet commands. Alternative would have been to not load at all \texttt{ncc-
comma} (or since 1.3zb \texttt{decimalcomma}) and provide the functionality purely by our own means;
because here in order to support \texttt{MTnonlettersobeymathxx} we are almost contrived to override
quasi entirely the contents of these tiny packages.

1.3zb adds support for the \texttt{decimalcomma} option. This was mandatory to keep a compatibility
layer with \texttt{frenchmath} after its 2.7 release of 2023/12/23.

Hesitation if I should also make it shadow the \texttt{ncccomma} option if both are used at same time,
or let the older option have priority. Well, let’s give priority to the new one so that one can do
\texttt{\PassOptionsToPackage} and recycle old documents compiled via \texttt{\input} to tell them to use the
new option.

Much ado about these tiny packages!
\RequirePackage{decimalcomma}[2023/12/28]% 1.3 or later

Attention that the breaking behavior of \AtBeginDocument at October 2020 \LaTeX{} release means that, taking into account that mathastext already has employed some \AtBeginDocument prior to loading decimalcomma, any code here will be executed BEFORE the \AtBeginDocument material from decimalcomma!

https://github.com/latex3/latex2e/issues/1226

So we definitely should not do here some \texttt{`\mathcode`,="8000\relax} in the \AtBeginDocument, and by the way I don’t even recall why I had this line at some point which ended up causing me some much suffering and pain and lost sleep. It seems to have been a silly copy-paste from the ncccomma branch next, and that I started experimenting before having even re-read the code I copied pasted and whether it was needed.

decimalcomma is a rewrite of icomma and it loads the latter for which babel-french has a detection mechanism, which as a result avoids the bad interactions with numprint plus its autolanguage option, which are mentioned below in the ncccomma branch. So we don’t need here the workaround done below is in the ncccomma branch. Notice though that in both cases, the ‘intelligent’ comma feature will be applied to the whole document, even inside those portions where the user has switched to another language such as English. This is to be expected here as nothing is done in a language specific manner, but if we wanted to do so, we might then be confronted with the babel issue mentioned next in the ncccomma branch.

\let\mathcomma\relax
\DeclareMathSymbol{\mathcomma}{\mathpunct}{\mst@font@tbu}{"2C}

Due to package decimalcomma internals, the hack here, which has to do with the “non letters obey math alphabets” optional mathastext feature, has to be done differently than the one we apply below for ncccomma. One can not really talk of a hack, as we basically have to redo the whole thing to insert an \texttt{\aftergroup} trick.

\def\mst@sm@rtcomma{\begingroup\@tfor\@tempa:=0123456789% \do{\expandafter\ifx\@tempa\@let@token \aftergroup\mathord \aftergroup\@gobble \@break@tfor\fi}\endgroup\mathpunct\mathcomma}

\mst@do@nonletters\expandafter{\the\mst@do@nonletters}

\else % end of decimalcomma branch
\fi
\else % end if ncccomma
\fi
\else
\fi
\let\mathcomma\mst@varfam@comma
\let\sm@rtcomma\mst@sm@rtcomma

Work around some bad interaction of ncccomma, numprint with autolanguage and babel-french. See https://github.com/latex3/babel/issues/190 for background. Some hesitation whether I should use the \texttt{\noextrasfrench} to work around babel-french code influencing non-French sections in the document. Update: I think the last sentence means I was hesitating at time of 1.3x whether to insert some extra code inside the \texttt{\noextrasfrench}.
Complications for compatibility with the \texttt{MTnonlettersobeymathxx} mechanism. No fix done here for usage by \texttt{ncccomma} of \texttt{@tempb} with no restoration of its meaning.

\edef\mst@NCC@comma{\let\noexpand\@empty\mathpunct\unexpanded\expandafter{\NCC@comma}\let\noexpand\@empty\noexpand\empty}
\mst@do@nonletters\expandafter{\the\mst@do@nonletters\let\mathcomma\mst@varfam@comma\let\NCC@comma\mst@NCC@comma}
\else % neither ncccomma nor decimalcomma
\expandafter\mst@addtodo@nonletters\string,\mathpunct\mst@varfam@comma
\fi
\DeclareMathSymbol{.}{\mathord}{\mst@font@tbu}{\string.\mathpunct\mst@varfam@dot}
\mst@addtodo@easynonletters\.
\mst@varfam@dot
\DeclareMathSymbol{:}{\mathrel}{\mst@font@tbu}{\string:}\mst@varfam@colon
\@ifpackageloaded{amsmath}{} % \colon defined in amsmath.sty in terms of : with some enlarged explicit % spacing. No need to intervene.
\else % no amssymb, use standard punctuation spacing
% the reason is if some package has redefined \colon which then % can not be used in \cs{DeclareMathSymbol} anymore (we % shamelessly overwrite...)\DeclareMathSymbol{\string\colon}{\mathbin}{\mst@font@tbu}{\string\colon}\mst@varfam@colon
\mst@do@nonletters\expandafter{\the\mst@do@nonletters\string,\mathrel\mst@varfam@colon}
1.3v uses \texttt{\protected\def\colon}{\mathpunct{\mst@varfam@colon}}\%
1.3x adds binarysemicolon option.
Due to the way = and - are used by \LaTeX in arrows, we will have to redefine \textbackslash Relbar and \textbackslash relbar in order for them to preserve their original meanings.

1.15d: Oct 13, 2012. Belated amendment of the code to be compatible with Unicode engines in case someone changed the mathcode of -. However, for the time being I can do it in an easy way only for Xe\LaTeX, not for Lua\LaTeX. Also I do my modifications to \textbackslash relbar in a manner testing for the presence of amsmath.

1.3v 2019/09/19: \LaTeX of 2019-10-01 defines \textbackslash leftarrowfill and \textbackslash rightarrowfill as robust macros, so we do the same.

I need to put amsmath under surveillance to check if it decides to robustify \textbackslash relbar at some point, now that the \LaTeX team has taken over maintenance.

2019/09/16 Use \textbackslash protected for \textbackslash right|leftarrowfill in the non \textbackslash DeclareRobustCommand branch?


\endash

1.1 2011/01/29: Producing this next piece of code was not a piece of cake for a novice like myself!

1.11 2011/02/05: Compatibility with Unicode (via use of fonts\spec encodeings EU1 and EU2)

1.12 2011/02/07: Improved dealing of Unicode possibility.

1.14b 2011/04/02: Corrected some very irresponsible bug in the Unicode part which caused a problem when 10 or more math families have been allocated.

1.15 2012/09/24: Added AtBeginDocument to circumvent some amsmath problem with unicode engines.

1.31 2016/01/29: anticipating TL2016 fonts\spec's switch to TU.
1.3t 2018/08/22: fix to very ancient (2012/12/20) bug with \DeclareMathSymbol lacking last argument if encoding not T1, OT1 or LY1 when setting up math mode to use the en-dash character as minus sign (PDFTeX engine).

Further, new macros \mst@subduedminus and \mst@nonsubduedminus, for the good functioning of the subdued option also in case of presence of fontspec. This is the only character for which subdued option works (now) by setting the mathcode on each math version change. Indeed, a typical issue is when the Unicode EN DASH or MINUS is used, but the actual font in subdued normal math version is originally in OT1 or T1 encoding. The only reasonable way to address this is by actually modifying the assigned mathcode at each version change. This means also that \MTversion and not \mathversion must be used for good functioning.

1.3u improves the handling of the minus sign by letting it be compatible with math versions (and not only with the with subdued mechanism but all math versions) having varying font encodings, even possibly classic 8bit font encoding mixed with TU encoding for Unicode engines. For this it is needed to work around a feature of XeTeX/LuaLaTeX, here is original comment:

afaict it is impossible to use straightforwardly in extended mathcode assignments a control sequence as created by \Umathchardef. This is counter-intuitive and breaks expectations.

But the 1.3u mechanism with \mst@UmathchardefWorkAround01 introduced a bug which showed under option noendash (hence also symboldelimiters) with Unicode engines. Fixed at 1.3w.

879 \let\mst@subduedminus\empty
880 \let\mst@nonsubduedminus\empty
881 \def\mst@dothe@endashstuff#1#2#3{%
882 \edef\mst@tmp@enc{#3}%
883 \if1\mst@OneifUniEnc
884 \mst@Umathchardef#1=2 \symmtoperatorfont "\mst@unicodeminus\relax
885 \mst@Umathchardef#2=7 \symmtoperatorfont "\mst@unicodeminus\relax
886 \else
887 \DeclareMathSymbol{#1}{\mathbin}{mtoperatorfont}{"
888 {\csname\mst@tmp@enc\string\textendash\endcsname}
889 \DeclareMathSymbol{#2}{\mathalpha}{mtoperatorfont}{"
890 {\csname\mst@tmp@enc\string\textendash\endcsname}
891 \fi}% mst@dothe@endashstuff
892 \def\mst@dothe@emdashstuff#1#2#3{%
893 \edef\mst@tmp@enc{#3}%
894 \if1\mst@OneifUniEnc
895 \mst@Umathchardef#1=2 \symmtoperatorfont "2014\relax
896 \mst@Umathchardef#2=7 \symmtoperatorfont "2014\relax
897 \else
898 \DeclareMathSymbol{#1}{\mathbin}{mtoperatorfont}{"
899 {\csname\mst@tmp@enc\string\textemdash\endcsname}
900 \DeclareMathSymbol{#2}{\mathalpha}{mtoperatorfont}{"
901 {\csname\mst@tmp@enc\string\textemdash\endcsname}
902 \fi}% mst@dothe@emdashstuff
903 \def\mst@dothe@hyphenstuff#1#2{%
904 \DeclareMathSymbol{#1}{\mathbin}{\mst@font@tbu}{"2D}%
905 \DeclareMathSymbol{#2}{\mathalpha}{\mst@font@tbu}{"2D}%

101
The above works only if the \texttt{\mst@minus@mv<name>} was really defined via \texttt{\Umathchardef}. If it was defined via \texttt{\DeclareMathSymbol} then it is a \texttt{\mathchar}, not a \texttt{\Umathchar}. At least currently (2019). So we need to correct the definition of \texttt{\mst@nonsubduedminus}.

\begin{verbatim}
\def\mst@UmathchardefWorkAround@i {\expandafter\mst@UmathchardefWorkAround@ii \meaning}\
\def\mst@UmathchardefWorkAround@ii #1"}\
\mst@hbar@mvnormal\
\mst@ltbar@mvnormal
\end{verbatim}

I decide to settle the question of the \texttt{\hbar}. The \LaTeX{} definition is \texttt{\def\hbar{{\mathchar'26\mkern-9muh}}} and its advantage is that \texttt{h} is in the correct font. But of course not the macron character (\texttt{\=, \bar}). And anyway \texttt{amsfonts} uses a \texttt{\DeclareMathSymbol}. Also there is the kern whose length depends on \texttt{cmsy} (18mu=1em and em taken from info in \texttt{cmsy}).

I will need an \texttt{rlap} adapted to math mode, and this is provided by code from Alexander R. Perlis in his TugBoat article 22 (2001), 350–352, which I found by googling \texttt{rlap}. (as an aside, I am only now (April 2, 2011) aware that the package \texttt{mathtools} provides the \texttt{mathrlap} etc... )

1.31 2016/01/29: anticipating TL2016 fontspec’s switch to TU.

1.3u 2019/08/20: encoding (8bits) agnostic construct for \texttt{hbar}, using same method as for \texttt{mathaccent} option. I should add some way to adjust the vertical positioning.

On this occasion I replace \texttt{h} by \texttt{\mst@h} because the mechanism for before and after skips does not interact well with the \texttt{rlap} construct.

1.3v 2019/09/19 adapts to maintain the robustness of \texttt{\hbar} which now applies with \texttt{\LaTeX{} 2019-10-01}.

1.3w works around \texttt{https://github.com/latex3/latex2e/issues/216} via \texttt{\mst@DeclareMathAccent}. The upstream bug affected the definition of \texttt{\mst@ltbar@mvnormal} and broke usage of \texttt{\Mathastext} in preamble.

1.3w also fixes oversight that \texttt{\hbar} may have been redefined via \texttt{\DeclareMathSymbol} by some package (e.g. \texttt{amsfonts}) and with \texttt{\LaTeX{} 2019-10-01} this means \texttt{\hbar<space>} is now
undefined. Modifying it changed nothing to \hbar behaviour in such circumstances. Finally we opt for a 'protected \hbar and choose to ignore completely if there is a \hbar<space> or not. To avoid extra steps we do not undefine it if it exists, because we would need to restore it in subdued math versions.

\let\mst@subduedhbar\@empty
\let\mst@nonsubduedhbar\@empty
\ifmst@nohbar\else
  \def\mst@subduedhbar{\let\hbar\mst@original@hbar}\%  
  \def\mst@nonsubduedhbar{\expandafter\hbar\csname mst@hbar@mv\math@version\endcsname}\%  
\fi
\def\mst@mathrlap{\mathpalette\mst@mathrlapinternal}
\def\mst@mathrlapinternal#1#2{\rlap{$\mathsurround=0pt#1{#2}$}}
\def\mst@dothe@hbarstuff#1#2#3{\edef\mst@tmp@enc{#3}\if1\mst@OneifUniEnc % Unicode engine and font \mst@Umathchardef#1=7 \symmtletterfont "0127 \relax \% or 210F? \else \begingroup \def\@text@composite##1\@text@composite##2{##2}\let\add@accent\@firstoftwo \mst@DeclareMathAccent{#2}{\mathalpha}{mtletterfont}{}\endgroup \protected\def#1{\mst@mathrlap{#2{\ }}\mst@h}\fi\fi}

1.15d: Oct 13, 2012. The \mathcode thing with = is (belatedly, sorry!) made Unicode compatible.

+,=,\Relbar

\ifmst@noplus\else\mst@infoline{\string+ and \string=}\fi
\DeclareMathSymbol{+}{\mathbin}{\mst@font@tbu}{"2B}
\DeclareMathSymbol{=}\mathrel\mst@font@tbu{"3D}

\def\Relbar{\mathrel\mst@equal@sign}
\DeclareRobustCommand\Relbar{\mathrel{\mst@equal@sign}}

\fss@catcodes 2012/12/18: Activating = (only in math mode actually) seems very bad but surprisingly works
well. However I had a problem with eu2lmtt.fd which should not be loaded with an active =.
2012/12/25: Since then I had switched to only math activation. And in fact the problematic = from eu2lmtt.fd end up in \csname...\endcsname and I have learnt since that TeX does not look at the mathcode inside a \csname...\endcsname. Example:

\%
\mathcode`x="8000
\begingroup
\%\catcode`x=\active
\%
\global\everymath{\defx{Hello}}
\%
\endgroup
\%\def\foox{World!}
\%
\$\text{x \csname ffoox\endcsname}$
\%

We need nevertheless to inactivate the =, for the following reason. Imagine someone did \catcode`-=\active\def=\string=, or another definition which would not lead to a tragedy in a \csname...\endcsname. Then the = is active and the re-definition done by \texttt{mathastext} will not be compatible with loading eu2lmtt.fd (for the first time) from math mode, as this re-definition can not be expanded inside a \csname...\endcsname.

2012/12/28: to be on the safe side, I add also ; and + and do it without discriminating between engines

\begin{verbatim}
969 \mst@infoline{adding \string= \string; and \string+ to \string\nfss@catcodes}
970 \g@addto@macro\nfss@catcodes{\
971 \e@makeother\=%
972 \e@makeother\;\%
973 \e@makeother\+\%
974 }
975 \expandafter\mst@addtodo@nonletters\string=\mathrel\mst@varfam@equal
976 \fi
\end{verbatim}

\texttt{noparenthesis} \lbrack and \rbrack are defined in latex.ltx by \texttt{\def\lbrack}{\def\rbrack[]} so this fits well with what we do here. \texttt{\lparen} and \texttt{\rparen} are similarly defined in mathtools. On the other hand in latex.ltx with \{ and \} are defined (in math mode) in terms of the control sequences \texttt{\lbrace} and \texttt{\rbrace}. Such control sequences can not be simultaneously math symbols and math delimiters, thus, this complicates things for the mathastextification.

\begin{verbatim}
977 \ifmst@noparen\else\mst@infoline{parentheses \string( \string) \string[ \string] and slash \string/}
978 \fi
\end{verbatim}

\texttt{noparenthesis} \lbrack and \rbrack are defined in latex.ltx by \texttt{\def\lbrack}{\def\rbrack[]} so this fits well with what we do here. \texttt{\lparen} and \texttt{\rparen} are similarly defined in mathtools. On the other hand in latex.ltx with \{ and \} are defined (in math mode) in terms of the control sequences \texttt{\lbrace} and \texttt{\rbrace}. Such control sequences can not be simultaneously math symbols and math delimiters, thus, this complicates things for the mathastextification.

\begin{verbatim}
977 \ifmst@noparen\else\mst@infoline{parentheses \string( \string) \string[ \string] and slash \string/}
978 \fi
\end{verbatim}

\texttt{noparenthesis} \lbrack and \rbrack are defined in latex.ltx by \texttt{\def\lbrack}{\def\rbrack[]} so this fits well with what we do here. \texttt{\lparen} and \texttt{\rparen} are similarly defined in mathtools. On the other hand in latex.ltx with \{ and \} are defined (in math mode) in terms of the control sequences \texttt{\lbrace} and \texttt{\rbrace}. Such control sequences can not be simultaneously math symbols and math delimiters, thus, this complicates things for the mathastextification.
Dec 18, 2012. We then want `\backslash` to do nothing when the `\backslash` is used as a delimiter. So here the original definition from `latex.ltx` is copied, generally speaking when people use other math symbol fonts they do respect the encoding of the CM symbols and largesymbols, so this is 90% safe. But in truth I should extract from the meaning of `\backslash` the delcode.

```
1.3v adds a `\protected` here for `\setminus`.
```

```
\models 1.15d: 13 oct 2012. Before modifying | we must preserve \models.
```

```
\ifmst\XeOrLua
\fi
```
(2011) I did not do anything then to try to emulate $\Vert$ with the vertical bar from the text font... and now (2012) \texttt{mathastext} is not as radical as it used to be anyhow, so it is too late. Or not (2019)? maybe I should do something here...

1.3v 2019/09/19: I discover this rather radical legacy \texttt{\def\vert{|}}, which is done here once in the preamble, but I leave it unmodified apart from prefixing it with \texttt{protected}. I also add a \texttt{protected} for the definition of \texttt{\mid} (which applies only under \texttt{\MTnonlettersobeymathxx} regime).

For 1.3 I make \texttt{\lbrace} and \texttt{\rbrace} undefined first, else problems may arise with some packages.

1.3e suppresses under option \texttt{nosmalldelims} the definitions of \texttt{\lbrace} and \texttt{\rbrace} as math symbols as this made \texttt{\left\lbrace} cause an error, it was a bug.

\LaTeX{} defines \texttt{\{} and \texttt{\} as robust commands for a long time (I don’t know since when). The \texttt{mathastext} redefinition is done only if user has executed \texttt{\MTexplicitbracesobeymathxx}, and it is done only when entering math mode, but there could be some \texttt{hbox} inside math, hence it has to be careful to be valid in text too.

1.3v maintains strict \LaTeX{}e robustness for \texttt{\{} and \texttt{\}}. This assumes no one fiddled with \texttt{\{} and \texttt{\}} proper (without space in the name).

\LaTeX{}e defines \texttt{\{} and \texttt{\} as robust commands for a long time (I don’t know since when). The \texttt{mathastext} redefinition is done only if user has executed \texttt{\MTexplicitbracesobeymathxx}, and it is done only when entering math mode, but there could be some \texttt{hbox} inside math, hence it has to be careful to be valid in text too.
specials

1.14b 2011/04/02: the redefinitions of #, $, \% and & were buggy (this showed up when 10 or more math families had been created).

1.15f 2012/10/23: the code, although working, was perhaps a bit insane and had definitions which could surprise other packages. For example, it did:

\renewcommand{\%}{\ifmmode\mt@mmode@percent\else\char37\relax\fi}

But it seems this provokes a problem with \texttt{microtype}. Perhaps the problem was that the command was not declared robust? For the dollar \LaTeX{} itself does

\DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}

So here I just modify \mathdollar. Then we have in \texttt{latex.ltx} the same definitions as in \texttt{plain.tex}: \texttt{\chardef\%=\%}, \texttt{\chardef\&=\&}, and \texttt{\chardef\#=\#}. It turns out that we can just adjust the mathcodes of these characters and achieve exactly what is wanted for the corresponding one char control sequences. In math mode the control sequence will use the specified mathcode. So here it is \texttt{not} a redefinition of the control sequences, purely an adjustment of mathcodes.

1.2d 2013/01/01: previous versions imposed the variable family type. I hereby make it possible to de-activate this feature with the macro \texttt{\MTeasynonlettersdonotobeymath}x. Besides, I have absolutely no idea why I had different looking code depending on the engine \texttt{Xe\LaTeX}, \texttt{Lua\LaTeX} or default. Removed.

1.3c 2013/12/14: I have absolutely no idea why I removed the \texttt{Xe\LaTeX} and \texttt{Lua\LaTeX} code at the time of 1.2d! the code for \texttt{tex/pdftex} engine could not accommodate more than 16 math families. Code for \texttt{Xe\LaTeX} and \texttt{Lua\LaTeX} again added. (and since TL2013 no more problems with \texttt{\luatexUmathcode}.)

1069 \ifmst@nospecials
1070 \else
1071 \mst@infoline{\string\#\space\string\mathdollar\space
1072 \string\&\space\string\&\space}
1073 \ifmst@XeOrLu\a
1074 \mst@Umathcode\#=0 \Symmtoperatorfont "23 \relax
1075 \mst@Umathchardef\mathdollar=0 \Symmtoperatorfont "24 \relax
1076 \mst@Umathcode\%=0 \Symmtoperatorfont "25 \relax
1077 \mst@Umathcode\&=0 \Symmtoperatorfont "26 \relax
1078 \mst@do@easynonletters\expandafter-codedata\%
1079 \the\mst@do@easynonletters
1080 \mst@Umathcode\#=7 \Symmtoperatorfont "23 \relax
1081 \mst@Umathchardef\mathdollar=7 \Symmtoperatorfont "24 \relax

107
symbolmisc We construct (with some effort) some long arrows from the Symbol glyphs, of almost the same lengths as the standard ones. By the way, I always found the \iff to be too wide, but I follow here the default. Also, although there is a \longmapsto in standard \LaTeX, if I am not mistaken, there is no \longto. So I define one here. I could not construct in the same manner \Longrightarrow etc. . . as the = sign from Symbol does not combine easily with the logical arrows, well, I could have done some box manipulations, but well, life is finite.

\prod
\sum
1.13b: I correct the brutal re-definitions of \prod and \sum from the earlier versions of the package; most of the time the Symbol glyphs do appear to be too small in display mode. The new redefinitions do have some defects: \$\textstyle\prod_1^2$ changes the position of limits but not the glyph itself, and $$\textstyle\prod_1^2$$ change the limits but switches to the CM inline math glyph. So I tried \renewcommand{\prod}{\mathchoice{\mst@prod}{\prodpsy}{\prodpsy}{\prodpsy}} but this did not go well with subscripts and exponents.

October 2012: maybe I should re-examine what I did?

1.3c (2013/12/14) renames \defaultprod to \MToriginalprod and \defaultsum to \MToriginalsum. 
1.3v hesitates about making robust here \prod and \sum. Finally I use \protected for them.
\ifmst@symbolmisc\mst@infoline{symbolmisc: miscellaneous math symbols from Symbol font}
\let\mst@prod\prod
\let\MToriginalprod\prod
\DeclareMathSymbol{\prodpsy}{\mathop}{mtpsymbol}{213}
\protected\def\prod{\ifinner\prodpsy\else\mst@prod\fi}
\let\mst@sum\sum
\let\MToriginalsum\sum
\DeclareMathSymbol{\sumpsy}{\mathop}{mtpsymbol}{229}
\protected\def\sum{\ifinner\sumpsy\else\mst@sum\fi}
\DeclareMathSymbol{\mst@implies}{\mathrel}{mtpsymbol}{222}
\DeclareRobustCommand*{\implies}{\;\mst@implies\;}
\DeclareMathSymbol{\mst@impliedby}{\mathrel}{mtpsymbol}{220}
\DeclareRobustCommand*{\impliedby}{\;\mst@impliedby\;}
\DeclareRobustCommand*{\iff}{\;\mst@impliedby\mathrel{\mkern-3mu}\mst@implies\;}
\DeclareMathSymbol{\mst@iff}{\mathrel}{mtpsymbol}{219}
\DeclareRobustCommand*{\shortiff}{\;\mst@iff\;}
\DeclareMathSymbol{\mst@to}{\mathrel}{mtpsymbol}{174}
\DeclareMathSymbol{\mst@trait}{\mathrel}{mtpsymbol}{190}
\DeclareRobustCommand*{\to}{\mst@to}
\DeclareRobustCommand*{\longto}{\mkern2mu\mst@trait\mathrel{\mkern-10mu}\mst@to}
\DeclareRobustCommand*{\mapsto}{\mapstochar\mathrel{\mkern0.2mu}\mst@to}
\DeclareRobustCommand*{\longmapsto}{\mapstochar\mathrel{\mkern2mu}\mst@trait\mathrel{\mkern-10mu}\mst@to}
\DeclareMathSymbol{\aleph}{\mathord}{mtpsymbol}{192}
\DeclareMathSymbol{\inftypsy}{\mathord}{mtpsymbol}{165}
\DeclareMathSymbol{\emptyset}{\mathord}{mtpsymbol}{198}
\let\varnothing\emptyset
\DeclareMathSymbol{\nabla}{\mathord}{mtpsymbol}{209}
\DeclareMathSymbol{\surd}{\mathop}{mtpsymbol}{214}
\let\angle\undefined
\DeclareMathSymbol{\angle}{\mathord}{mtpsymbol}{208}
\DeclareMathSymbol{\forall}{\mathord}{mtpsymbol}{34}
\DeclareMathSymbol{\exists}{\mathord}{mtpsymbol}{36}
\DeclareMathSymbol{\neg}{\mathord}{mtpsymbol}{216}
\DeclareMathSymbol{\clubsuit}{\mathord}{mtpsymbol}{167}
\DeclareMathSymbol{\diamondsuit}{\mathord}{mtpsymbol}{168}
\DeclareMathSymbol{\heartsuit}{\mathord}{mtpsymbol}{169}
\DeclareMathSymbol{\spadesuit}{\mathord}{mtpsymbol}{170}
\DeclareMathSymbol{\smallint}{\mathop}{mtpsymbol}{242}
\DeclareMathSymbol{\wedge}{\mathbin}{mtpsymbol}{217}
\DeclareMathSymbol{\vee}{\mathbin}{mtpsymbol}{218}
\DeclareMathSymbol{\cap}{\mathbin}{mtpsymbol}{199}
\DeclareMathSymbol{\cup}{\mathbin}{mtpsymbol}{200}
\DeclareMathSymbol{\bullet}{\mathbin}{mtpsymbol}{183}
\DeclareMathSymbol{\div}{\mathbin}{mtpsymbol}{184}
\DeclareMathSymbol{\otimes}{\mathbin}{mtpsymbol}{196}
\DeclareMathSymbol{\oplus}{\mathbin}{mtpsymbol}{197}
symbolre  I like the \Re and \Im from Symbol, so I overwrite the CM ones.

Greek letters  LGRgreek > selfGreek > eulergreek > symbolgreek

1.11 I correct some bugs on how eulergreek and symbolgreek interacted.
1.12b more bug fixes.
1.13
* Option LGRgreek.
* Also, a behavior has been changed: it regards the selfGreek case, the default shape is now
  the one for letters, not for operator-names and digits. This complies to the ISO standard.
  * bugfix: version 1.12b did not define the \omicron in the case when no Greek-related option
    was passed to the package.
1.13d has new macros \MTstandardgreek and \MTcustomgreek. And in the subdued case
\MTstandardgreek is done when switching to the normal or bold math versions (previously
something like this was only done in case of LGRgreek option.)
The \omicron{} requires special treatment. By default we use the o from the (original) normal alphabet, if euler
greek or symbolgreek we adapt. There is also a special adjustment if the package fourier was loaded in its upright
variant: we then take \omicron{} from the (original) rm alphabet.

When we in fact use Symbol, we have to correct \Rho{} and \Chi{}. And \Digamma{} is non-existent
in fact (no F in Symbol, F codes a \Phi{}).
not symbolgreek but eulergreek or selfGreek. Note 2015/10/31 : apparemment à un moment
dans le passé je considérais eulergreek et selfGreek comme pouvant être utilisés simultanément
car j'avais ici 'or both'. Mais je laisse tomber tout effort réel de m'en préoccuper.

There are differences regarding Euler and Symbol with respect to the available var-letters. We
include one or two things like the wp and the partial.

The lower case Greek letters in default L\LaTeX{} are of type mathord. If we use the Euler font it
is perhaps better to have them be of type mathalpha.
\alphaupt etc... Completely refactored at 1.3x to define \Alphaup, \Alphait, \alphaup, \alphait, etc... and prepare templates \Alpha, ..., \alpha, ..., which when activating a math version will be submitted to an \expanded, whose behaviour will depend on version-specific conditionals.

1.3w and earlier had a bug regarding Digamma which was set up to use same font shape as for lowercase digamma.
\DeclareMathSymbol{\Xiup}{\mathalpha}{mtgreekup}{88}
\DeclareMathSymbol{\Piup}{\mathalpha}{mtgreekup}{80}
\DeclareMathSymbol{\Sigmaup}{\mathalpha}{mtgreekup}{83}
\DeclareMathSymbol{\Upsilonup}{\mathalpha}{mtgreekup}{85}
\DeclareMathSymbol{\Phiup}{\mathalpha}{mtgreekup}{70}
\DeclareMathSymbol{\Psiup}{\mathalpha}{mtgreekup}{89}
\DeclareMathSymbol{\Omegaup}{\mathalpha}{mtgreekup}{87}
% 
\DeclareMathSymbol{\Gammait}{\mathalpha}{mtgreekit}{71}
\DeclareMathSymbol{\Deltait}{\mathalpha}{mtgreekit}{68}
\DeclareMathSymbol{\Thetait}{\mathalpha}{mtgreekit}{74}
\DeclareMathSymbol{\Lambdait}{\mathalpha}{mtgreekit}{88}
\DeclareMathSymbol{\Piit}{\mathalpha}{mtgreekit}{80}
\DeclareMathSymbol{\Sigmait}{\mathalpha}{mtgreekit}{83}
\DeclareMathSymbol{\Upsilonit}{\mathalpha}{mtgreekit}{85}
\DeclareMathSymbol{\Phiit}{\mathalpha}{mtgreekit}{70}
\DeclareMathSymbol{\Psiit}{\mathalpha}{mtgreekit}{89}
\DeclareMathSymbol{\Omegait}{\mathalpha}{mtgreekit}{87}
% 
\def\mst@Alpha{\ifmst@greek@upper@up\Alphaup\else\Alphait\fi}%
\def\mst@Beta{\ifmst@greek@upper@up\Betaup\else\Betait\fi}%
\def\mst@Epsilon{\ifmst@greek@upper@up\Epsilonup\else\Epsilonit\fi}%
\def\mst@Zeta{\ifmst@greek@upper@up\Zetaup\else\Zetait\fi}%
\def\mst@Iota{\ifmst@greek@upper@up\Iotaup\else\Iotait\fi}%
\def\mst@Kappa{\ifmst@greek@upper@up\Kappaup\else\Kappait\fi}%
\def\mst@Mu{\ifmst@greek@upper@up\Muup\else\Muit\fi}%
\def\mst@Nu{\ifmst@greek@upper@up\Nuup\else\Nuit\fi}%
\def\mst@Omicron{\ifmst@greek@upper@up\Omicronup\else\Omicronit\fi}%
\def\mst@Rho{\ifmst@greek@upper@up\Rhoup\else\Rhoit\fi}%
\def\mst@Tau{\ifmst@greek@upper@up\Taup\else\Tauit\fi}%
\def\mst@Chi{\ifmst@greek@upper@up\Chiup\else\Chiit\fi}%
% 
\def\mst@Digamma{\ifmst@greek@upper@up\Digammaup\else\Digammait\fi}%
\def\mst@Gamma{\ifmst@greek@upper@up\Gammaup\else\Gammait\fi}%
\def\mst@Delta{\ifmst@greek@upper@up\Deltaup\else\Deltait\fi}%
\def\mst@Theta{\ifmst@greek@upper@up\Thetaup\else\Thetait\fi}%
\def\mst@Lambda{\ifmst@greek@upper@up\Lambdaup\else\Lambdait\fi}%
\def\mst@Xi{\ifmst@greek@upper@up\Xiup\else\Xit\fi}%
\def\mst@Pi{\ifmst@greek@upper@up\Piup\else\Pit\fi}%
\def\mst@Sigma{\ifmst@greek@upper@up\Sigmaup\else\Sigmat\fi}%
\def\mst@Upsilon{\ifmst@greek@upper@up\Upsilonup\else\Upsilonit\fi}%
\def\mst@Phi{\ifmst@greek@upper@up\Phiup\else\Phit\fi}%
\def\mst@Psi{\ifmst@greek@upper@up\Psiup\else\Psiit\fi}%
\def\mst@Omega{\ifmst@greek@upper@up\Omegaup\else\Omegait\fi}%
% 
\DeclareMathSymbol{\alphaup}{\mathalpha}{mtgreekup}{97}
\DeclareMathSymbol{\betaup}{\mathalpha}{mtgreekup}{98}
\DeclareMathSymbol{\gammaup}{\mathalpha}{mtgreekup}{103}
\DeclareMathSymbol{\deltaup}{\mathalpha}{mtgreekup}{100}
\DeclareMathSymbol{\epsilonup}{\mathalpha}{mtgreekup}{101}
\DeclareMathSymbol{\zetaup}{\mathalpha}{mtgreekup}{122}
\DeclareMathSymbol{\etaup}{\mathalpha}{mtgreekup}{104}
\DeclareMathSymbol{\thetaup}{\mathalpha}{mtgreekup}{106}
\DeclareMathSymbol{\iotaup}{\mathalpha}{mtgreekup}{105}
\DeclareMathSymbol{\kappaup}{\mathalpha}{mtgreekup}{107}
\DeclareMathSymbol{\lambdaup}{\mathalpha}{mtgreekup}{108}
\DeclareMathSymbol{\muup}{\mathalpha}{mtgreekup}{109}
\DeclareMathSymbol{\nuup}{\mathalpha}{mtgreekup}{110}
\DeclareMathSymbol{\xiup}{\mathalpha}{mtgreekup}{111}
\DeclareMathSymbol{\omicronup}{\mathalpha}{mtgreekup}{112}
\DeclareMathSymbol{\upsilonup}{\mathalpha}{mtgreekup}{113}
\DeclareMathSymbol{\phiup}{\mathalpha}{mtgreekup}{114}
\DeclareMathSymbol{\chiup}{\mathalpha}{mtgreekup}{115}
\DeclareMathSymbol{\psiup}{\mathalpha}{mtgreekup}{116}
\DeclareMathSymbol{\omegaup}{\mathalpha}{mtgreekup}{117}
\DeclareMathSymbol{\digammaup}{\mathalpha}{mtgreekup}{118}
\DeclareMathSymbol{\varsigmaup}{\mathalpha}{mtgreekup}{119}
\DeclareMathSymbol{\alphait}{\mathalpha}{mtgreekit}{97}
\DeclareMathSymbol{\betait}{\mathalpha}{mtgreekit}{98}
\DeclareMathSymbol{\gammait}{\mathalpha}{mtgreekit}{103}
\DeclareMathSymbol{\deltait}{\mathalpha}{mtgreekit}{100}
\DeclareMathSymbol{\epsilonit}{\mathalpha}{mtgreekit}{101}
\DeclareMathSymbol{\zetait}{\mathalpha}{mtgreekit}{122}
\DeclareMathSymbol{\etait}{\mathalpha}{mtgreekit}{104}
\DeclareMathSymbol{\thetait}{\mathalpha}{mtgreekit}{106}
\DeclareMathSymbol{\iotait}{\mathalpha}{mtgreekit}{105}
\DeclareMathSymbol{\kappait}{\mathalpha}{mtgreekit}{107}
\DeclareMathSymbol{\lambdait}{\mathalpha}{mtgreekit}{108}
\DeclareMathSymbol{\muit}{\mathalpha}{mtgreekit}{109}
\DeclareMathSymbol{\nuit}{\mathalpha}{mtgreekit}{110}
\DeclareMathSymbol{\xiit}{\mathalpha}{mtgreekit}{111}
\DeclareMathSymbol{\omicronit}{\mathalpha}{mtgreekit}{112}
\DeclareMathSymbol{\piit}{\mathalpha}{mtgreekit}{113}
\DeclareMathSymbol{\rhoit}{\mathalpha}{mtgreekit}{114}
\DeclareMathSymbol{\sigmait}{\mathalpha}{mtgreekit}{115}
\DeclareMathSymbol{\tauit}{\mathalpha}{mtgreekit}{116}
\DeclareMathSymbol{\upsilonit}{\mathalpha}{mtgreekit}{117}
\DeclareMathSymbol{\phiit}{\mathalpha}{mtgreekit}{118}
\DeclareMathSymbol{\digammait}{\mathalpha}{mtgreekit}{119}
\DeclareMathSymbol{\varsigmain}{\mathalpha}{mtgreekit}{120}
\DeclareMathSymbol{\alphadigamma}{\mathalpha}{mtgreekit}{121}
\DeclareMathSymbol{\betadigamma}{\mathalpha}{mtgreekit}{122}
\DeclareMathSymbol{\gammadigamma}{\mathalpha}{mtgreekit}{123}
\DeclareMathSymbol{\deltadigamma}{\mathalpha}{mtgreekit}{124}
\DeclareMathSymbol{\epiddigamma}{\mathalpha}{mtgreekit}{125}
\DeclareMathSymbol{\zetadigamma}{\mathalpha}{mtgreekit}{126}
\DeclareMathSymbol{\etadigamma}{\mathalpha}{mtgreekit}{127}
\DeclareMathSymbol{\thetadigamma}{\mathalpha}{mtgreekit}{128}
\DeclareMathSymbol{\iota}{\mathalpha}{mtgreekit}{129}
\DeclareMathSymbol{\kappadigamma}{\mathalpha}{mtgreekit}{130}
\DeclareMathSymbol{\lambdadi}{\mathalpha}{mtgreekit}{131}
\DeclareMathSymbol{\mudigamma}{\mathalpha}{mtgreekit}{132}
\DeclareMathSymbol{\nudi}{\mathalpha}{mtgreekit}{133}
\DeclareMathSymbol{\xid}{\mathalpha}{mtgreekit}{134}
\DeclareMathSymbol{\omicron}{\mathalpha}{mtgreekit}{135}
\DeclareMathSymbol{\pid}{\mathalpha}{mtgreekit}{136}
\DeclareMathSymbol{\rhoid}{\mathalpha}{mtgreekit}{137}
\DeclareMathSymbol{\sigmadigamma}{\mathalpha}{mtgreekit}{138}
\DeclareMathSymbol{\tauid}{\mathalpha}{mtgreekit}{139}
\DeclareMathSymbol{\upsilondigamma}{\mathalpha}{mtgreekit}{140}
\DeclareMathSymbol{\phidi}{\mathalpha}{mtgreekit}{141}
\DeclareMathSymbol{\digammadigamma}{\mathalpha}{mtgreekit}{142}
\ DeclareMathSymbol{\varsigmain}{\mathalpha}{mtgreekit}{143}
\ DeclareMathSymbol{\alphadigamma}{\mathalpha}{mtgreekit}{144}
\ DeclareMathSymbol{\betadigamma}{\mathalpha}{mtgreekit}{145}
\ DeclareMathSymbol{\gammadigamma}{\mathalpha}{mtgreekit}{146}
\ DeclareMathSymbol{\deltadigamma}{\mathalpha}{mtgreekit}{147}
\ DeclareMathSymbol{\epiddigamma}{\mathalpha}{mtgreekit}{148}
\ DeclareMathSymbol{\zetadigamma}{\mathalpha}{mtgreekit}{149}
\ DeclareMathSymbol{\etadigamma}{\mathalpha}{mtgreekit}{150}
\ DeclareMathSymbol{\thetadigamma}{\mathalpha}{mtgreekit}{151}
\ DeclareMathSymbol{\iota}{\mathalpha}{mtgreekit}{152}
\ DeclareMathSymbol{\kappadigamma}{\mathalpha}{mtgreekit}{153}
\ DeclareMathSymbol{\lambdadi}{\mathalpha}{mtgreekit}{154}
\ DeclareMathSymbol{\mudigamma}{\mathalpha}{mtgreekit}{155}
\ DeclareMathSymbol{\nudi}{\mathalpha}{mtgreekit}{156}
\ DeclareMathSymbol{\xid}{\mathalpha}{mtgreekit}{157}
\ DeclareMathSymbol{\omicron}{\mathalpha}{mtgreekit}{158}
\ DeclareMathSymbol{\pid}{\mathalpha}{mtgreekit}{159}
\ DeclareMathSymbol{\rhoid}{\mathalpha}{mtgreekit}{160}
\ DeclareMathSymbol{\sigmadigamma}{\mathalpha}{mtgreekit}{161}
\ DeclareMathSymbol{\tauid}{\mathalpha}{mtgreekit}{162}
\ DeclareMathSymbol{\upsilondigamma}{\mathalpha}{mtgreekit}{163}
\ DeclareMathSymbol{\phidi}{\mathalpha}{mtgreekit}{164}
\ DeclareMathSymbol{\digammadigamma}{\mathalpha}{mtgreekit}{165}
\DeclareMathSymbol{\chiit}{\mathalpha}{mtgreekit}{113}
\DeclareMathSymbol{\psiit}{\mathalpha}{mtgreekit}{121}
\DeclareMathSymbol{\omegait}{\mathalpha}{mtgreekit}{119}
\DeclareMathSymbol{\digammait}{\mathalpha}{mtgreekit}{147}
\DeclareMathSymbol{\varsigmait}{\mathalpha}{mtgreekit}{99}
\def\mst@alpha{\ifmst@greek@lower@up\alphaup\else\alphait\fi}\%
\def\mst@beta{\ifmst@greek@lower@up\betaup\else\betait\fi}\%
\def\mst@gamma{\ifmst@greek@lower@up\gammaup\else\gammait\fi}\%
\def\mst@delta{\ifmst@greek@lower@up\deltaup\else\deltait\fi}\%
\def\mst@epsilon{\ifmst@greek@lower@up\epsilonup\else\epsilonit\fi}\%
\def\mst@zeta{\ifmst@greek@lower@up\zetaup\else\zetait\fi}\%
\def\mst@eta{\ifmst@greek@lower@up\etaup\else\etait\fi}\%
\def\mst@theta{\ifmst@greek@lower@up\thetaup\else\thetait\fi}\%
\def\mst@iota{\ifmst@greek@lower@up\iotaup\else\iotait\fi}\%
\def\mst@kappa{\ifmst@greek@lower@up\kappaup\else\kappait\fi}\%
\def\mst@lambda{\ifmst@greek@lower@up\lambdaup\else\lambdait\fi}\%
\def\mst@mu{\ifmst@greek@lower@up\muup\else\muit\fi}\%
\def\mst@nu{\ifmst@greek@lower@up\nuup\else
uit\fi}\%
\def\mst@xi{\ifmst@greek@lower@up\xiusp\else\xiit\fi}\%
\def\mst@omicron{\ifmst@greek@lower@up\omicronup\else\omicronit\fi}\%
\def\mst@pi{\ifmst@greek@lower@up\piup\else\piit\fi}\%
\def\mst@rho{\ifmst@greek@lower@up\rhoup\else\rhoit\fi}\%
\def\mst@sigma{\ifmst@greek@lower@up\sigmaup\else\sigmait\fi}\%
\def\mst@tau{\ifmst@greek@lower@up\taup\else\tauit\fi}\%
\def\mst@upsilon{\ifmst@greek@lower@up\upsilonup\else\upsilonit\fi}\%
\def\mst@phi{\ifmst@greek@lower@up\phiup\else\phiit\fi}\%
\def\mst@chi{\ifmst@greek@lower@up\chiup\else\chiit\fi}\%
\def\mst@psi{\ifmst@greek@lower@up\psiusp\else\psiit\fi}\%
\def\mst@omega{\ifmst@greek@lower@up\omegaup\else\omegait\fi}\%
\def\mst@digamma{\ifmst@greek@lower@up\digammaup\else\digammait\fi}\%
\def\mst@varsigma{\ifmst@greek@lower@up\varsigmaup\else\varsigmait\fi}\%
\newcommand*{\MTstandardgreek}{}
\newcommand*{\MTcustomgreek}{}
\newcommand*{\MTrecordstandardgreek}{}
\ifmst@customgreek\renewcommand*{\MTrecordstandardgreek}{%}
\let\mst@origAlpha\Alpha\%
\let\mst@origBeta\Beta

\MTstandardgreek 1.3d 2014/05/23 defines the commands \MTstandardgreek and \MTcustomgreek for package and user. I leave \MTrecordstandardgreek undocumented as I don’t want to encourage people to load math packages after mathastext.
\MTcustomgreek 1.3h 2015/10/31: corrected \MTcustomgreek as it caused \ell to become undefined under option symbolgreek and, much more catastrophic, caused \alpha, etc. to become undefined under option selfGreek!
\MTrecordstandardgreek 1.3d 2014/05/23 defines the commands \MTstandardgreek and \MTcustomgreek for package and user. I leave \MTrecordstandardgreek undocumented as I don’t want to encourage people to load math packages after mathastext.
\MTcustomgreek 1.3h 2015/10/31: corrected \MTcustomgreek as it caused \ell to become undefined under option symbolgreek and, much more catastrophic, caused \alpha, etc. to become undefined under option selfGreek!
\let\mst@origGamma\Gamma
\let\mst@origDelta\Delta
\let\mst@origEpsilon\Epsilon
\let\mst@origZeta\Zeta
\let\mst@origEta\Eta
\let\mst@origTheta\Theta
\let\mst@origIota\Iota
\let\mst@origKappa\Kappa
\let\mst@origLambda\Lambda
\let\mst@origMu\Mu
\let\mst@origNu\Nu
\let\mst@origXi\Xi
\let\mst@origOmicron\Omicron
\let\mst@origPi\Pi
\let\mst@origRho\Rho
\let\mst@origSigma\Sigma
\let\mst@origTau\Tau
\let\mst@origUpsilon\Upsilon
\let\mst@origPhi\Phi
\let\mst@origChi\Chi
\let\mst@origPsi\Psi
\let\mst@origOmega\Omega
\let\mst@origalpha\alpha
\let\mst@origbeta\beta
\let\mst@origgamma\gamma
\let\mst@origdelta\delta
\let\mst@origepsilon\epsilon
\let\mst@origvarepsilon\varepsilon
\let\mst@origzeta\zeta
\let\mst@origeta\eta
\let\mst@origtheta\theta
\let\mst@origvartheta\vartheta
\let\mst@origiota\iota
\let\mst@origkappa\kappa
\let\mst@origlambda\lambda
\let\mst@origmu\mu
\let\mst@orignu\nu
\let\mst@origxi\xi
\let\mst@origomicron\omicron
\let\mst@origvarpi\varpi
\let\mst@origvarrho\varrho
\let\mst@origvarsigma\varsigma
\let\mst@origtau\tau
\let\mst@origupsilon\upsilon
\let\mst@origphi\phi
\let\mst@origvarphi\varphi
\let\mst@origchi\chi
\let\mst@origpsi\psi
\let\mst@origomega\omega
\let\mst@origDigamma\Digamma
\let\mst@origdigamma\digamma
\let\mst@origpartial\partial
\let\mst@origwp\wp
\let\mst@origell\ell }% \MTrecordstandardgreek
\renewcommand*{\MTstandardgreek}{%
\let\Alpha\mst@origAlpha
\let\Beta\mst@origBeta
\let\Gamma\mst@origGamma
\let\Delta\mst@origDelta
\let\Epsilon\mst@origEpsilon
\let\Zeta\mst@origZeta
\let\Eta\mst@origEta
\let\Theta\mst@origTheta
\let\Iota\mst@origIota
\let\Kappa\mst@origKappa
\let\Lambda\mst@origLambda
\let\Mu\mst@origMu
\let\Nu\mst@origNu
\let\Xi\mst@origXi
\let\Omicron\mst@origOmicron
\let\Pi\mst@origPi
\let\Rho\mst@origRho
\let\Sigma\mst@origSigma
\let\Tau\mst@origTau
\let\Upsilon\mst@origUpsilon
\let\Phi\mst@origPhi
\let\Chi\mst@origChi
\let\Psi\mst@origPsi
\let\Omega\mst@origOmega
%}
\renewcommand*{\MTstandardgreek}{%
1.3a implementation of \texttt{LGRgreek+} option. It is not exactly clear what we should do for \texttt{\mathnormal} and \texttt{\mathnormalbold}.

This definition allows usage of \texttt{\alpha} for example in numerical context. To be completely clean perhaps we should get rid of final \texttt{\fi}, but old-fashioned \LaTeX{} does not have built-in conveniences, were it not for the nested if’s simple \texttt{\expandafter} would do, but here we would need three in four places. Or simply wrap the whole in \texttt{\expanded}. Anyway, not really important.

1605 \let\lambda\mst@origlambda
1606 \let\mu\mst@origmu
1607 \let\nu\mst@orignu
1608 \let\xi\mst@origxi
1609 \let\omicron\mst@origomicron
1610 \let\pi\mst@origpi
1611 \let\varpi\mst@origvarpi
1612 \let\rho\mst@origrho
1613 \let\varrho\mst@origvarrho
1614 \let\sigma\mst@origsigma
1615 \let\varsigma\mst@origvarsigma
1616 \let\tau\mst@origtau
1617 \let\upsilon\mst@origupsilon
1618 \let\phi\mst@origphi
1619 \let\varphi\mst@origvarphi
1620 \let\chi\mst@origchi
1621 \let\psi\mst@origpsi
1622 \let\omega\mst@origomega
1623 \let\Digamma\mst@origDigamma
1624 \let\partial\mst@origpartial
1625 \let\wp\mst@origwp
1626 \let\ell\mst@origell
1627 \% \MTstandardgreek
1628 \ifmst@greekplus
1629 )\% MTstandardgreek
1630 \def\mst@define@lowergreekletter#1#2{%
1631 \protected\def\ifcase\mst@mathalph
1632 \ifmst@greek@lower@up\mathgreekup{#2}\else\mathgreekit{#2}\fi
1633 \or % \texttt{\rm}
1634 \mathgreekup{#2}\%
1635 \or % \texttt{\bf}
1636 \mathgreekupbold{#2}\%
1637 \or % \texttt{\it}
1638 \mathgreekit{#2}\%
1639 \or % \texttt{\normalbold}
1640 \ifmst@greek@lower@up\mathgreekupbold{#2}\else\mathgreekitbold{#2}\fi
1641 \else \#2\fi)\%
1642 }
1643 \def\mst@define@uppergreekletter#1#2{%
1644 \protected\def\ifcase\mst@mathalph
1645 \ifmst@greek@lower@up\mathgreekup{#2}\else\mathgreekit{#2}\fi
1646 \or % \texttt{\rm}
1647 }
\textgreekup{#2}\%\n\\%bf
\textgreekupbold{#2}\%\n\\%it
\textgreekit{#2}\%\n\\%mathnormalbold
\ifmst@greek@upperup\textgreekupbold{#2}\else\textgreekitbold{#2}\fi\
\else #2\fi\%
\renewcommand*{\MTcustomgreek}{%\n\mst@define@uppergreekletter\Alpha\mst@Alpha
\mst@define@uppergreekletter\Beta\mst@Beta
\mst@define@uppergreekletter\Epsilon\mst@Epsilon
\mst@define@uppergreekletter\Zeta\mst@Zeta
\mst@define@uppergreekletter\Eta\mst@Eta
\mst@define@uppergreekletter\Iota\mst@Iota
\mst@define@uppergreekletter\Kappa\mst@Kappa
\mst@define@uppergreekletter\Mu\mst@Mu
\mst@define@uppergreekletter\Nu\mst@Nu
\mst@define@uppergreekletter\omicron\mst@omicron
\mst@define@uppergreekletter\Rho\mst@Rho
\mst@define@uppergreekletter\Tau\mst@Tau
\mst@define@uppergreekletter\Chi\mst@Chi
\mst@define@uppergreekletter\Digamma\mst@Digamma
\mst@define@uppergreekletter\Gamma\mst@Gamma
\mst@define@uppergreekletter\Delta\mst@Delta
\mst@define@uppergreekletter\Theta\mst@Theta
\mst@define@uppergreekletter\Lambda\mst@Lambda
\mst@define@uppergreekletter\Xi\mst@Xi
\mst@define@uppergreekletter\Pi\mst@Pi
\mst@define@uppergreekletter\Sigma\mst@Sigma
\mst@define@uppergreekletter\Upsilon\mst@Upsilon
\mst@define@uppergreekletter\Phi\mst@Phi
\mst@define@uppergreekletter\Psi\mst@Psi
\mst@define@uppergreekletter\Omega\mst@Omega
\mst@define@lowergreekletter\alpha\mst@alpha
\mst@define@lowergreekletter\beta\mst@beta
\mst@define@lowergreekletter\gamma\mst@gamma
\mst@define@lowergreekletter\Delta\mst@Delta
\mst@define@lowergreekletter\epsilon\mst@epsilon
\mst@define@lowergreekletter\zeta\mst@zeta
\mst@define@lowergreekletter\eta\mst@eta
\mst@define@lowergreekletter\theta\mst@theta
\mst@define@lowergreekletter\iota\mst@iota
\mst@define@lowergreekletter\kappa\mst@kappa
\mst@define@lowergreekletter\lambda\mst@lambda
\mst@define@lowergreekletter\mu\mst@mu
\mst@define@lowergreekletter\nu\mst@nu
\mst@define@lowergreekletter\Xi\mst@Xi
Under selfGreek or other Greek option but not LGRgreek, these Greek letter control sequences are already \mathchar’s, but under LGRgreek they need (well not really, but I feel it is cleaner) expansion which will react to the Boolean saying if using ‘upright’ or ‘italic’. This Boolean setting is recorded when declaring a math version and reenacted when \MTversion is encountered in the document body. We must be careful not to contaminate things in the principal mode from math version declarations but I think my (now quite old) code is globally designed to achieve this protection see how \MTDeclareVersion is done. The \MTcustomgreek will always be executed in preamble at least once, except under subdued option.

The \expanded’s act on unexpanding tokens if not used under LGRgreek regimen.
1.3h 2015/10/31 adds this conditional to correct the bug in 1.3d 2014/05/23 which caused \alpha etc... to become undefined under option selfGreek.

\ifmst@selfGreek\else
\expanded{\let
oexpand\alpha\mst@alpha
\let
oexpand\beta\mst@beta
\let
oexpand\gamma\mst@gamma
\let
oexpand\delta\mst@delta
\let
oexpand\epsilon\mst@epsilon
\let
oexpand\zeta\mst@zeta
\let
oexpand\eta\mst@eta
\let
oexpand\theta\mst@theta
\let
oexpand\iota\mst@iota
\let
oexpand\kappa\mst@kappa
\let
oexpand\lambda\mst@lambda
\let
oexpand\mu\mst@mu
\let
oexpand\nu\mst@nu
\let
oexpand\xi\mst@xi
\let
oexpand\omicron\mst@omicron
\let
oexpand\pi\mst@pi
\let
oexpand\rho\mst@rho
\let
oexpand\sigma\mst@sigma
\let
oexpand\tau\mst@tau
\let
oexpand\upsilon\mst@upsilon
\let
oexpand\phi\mst@phi
\let
oexpand\chi\mst@chi
\let
oexpand\psi\mst@psi
\let
oexpand\omega\mst@omega
\let
oexpand\varsigma\mst@varsigma}
\fi

\ifmst@LGRgreek\else
\let\varepsilon\mst@varepsilon
\let\vartheta\mst@vartheta
\let\varpi\mst@varpi
\let\varrho\mst@varrho
\let\varphi\mst@varphi
\fi

\ifmst@LGRgreek\else
\let\digamma\mst@digamma
\fi

\ifmst@LGRgreek\else
\expanded{\let\varepsilon\mst@varepsilon
\let\vartheta\mst@vartheta
\let\varpi\mst@varpi
\let\varrho\mst@varrho
\let\varphi\mst@varphi}
\fi

1.3h: digamma only defined with option LGRgreek:
\ifmst@LGRgreek
\expanded{\let\varepsilon\mst@varepsilon
\let\digamma\mst@digamma}
\fi

\ifmst@LGRgreek\else
\let\varepsilon\mst@varepsilon
\let\digamma\mst@digamma
\fi

% conditional added 1.3h 2015/10/31
In 1.0, I had them of type mathord, here I choose mathalpha. If I used \i and \j from the text font the problem would be with the fontsize, if in scriptstyle. The amsmath \text would do the trick.

1.14b 2011/04/02: again this bug in the EU1/EU2 encoding part, as in the code redefining $ etc in math mode (see above). Fixed.

1.31 2016/01/29: anticipating TL2016 fontspec's switch to TU.

1.3t 2018/08/22 removes the definitions done of \i and \j since 1.12 (as robust commands usable both in text and math mode).

1.3u lets the \imath and \jmath react to the font encoding at each math version.

1.3v lets the redefined \imath and \jmath be \protected.
Obsolet comments relative to the 2011 code:

I don’t know how to get from the encoding to the slot positions of the accents (apart from going to look at all possible encodings definition files and putting this info here). In standard \TeX, the math accents are taken from the ‘operators’ font. So we do the same here. Of course there is the problem that the user can define math versions with different encodings. Here I take T1 if it was the default at the time of loading the package, else OT1. 1.12b: I add LY1 which is quasi like OT1.

At 1.3u 2019/08/20 I decide to remove the hard-coded slot positions for OT1, T1 and LY1, and replace them with some hack which assumes \LaTeXe way of handling text accents got executed by the encoding definition file. If not, some breakage on package loading could occur, but this whole thing is conditional on the \texttt{mathaccents} option anyway, which per default is not executed.

The \texttt{vec} accent is not considered here because it has no suitable available glyph in a standard 8bits text font encodings.

Also at 1.3u the math accents adapt to the font encoding at each math version.

1.3v adapts to \LaTeX 2019-10-01 which now comes with robust math accent macros. The «original»-named macros are without the robustifying space (NOT true anymore, see 1.3w next), as they only serve as meaning holders.

On the other hand the macros indexed by math version names are (in the pdflatex branch) always defined via \texttt{\DeclareMathAccent} hence they will be robust with 2019-10-01 or later and we must use the \texttt{\mst@robustifyingspace} with them to access their real meaning (this thus differs from the situation with \texttt{\bar}).

1.3w The above was a bit optimistic as \texttt{amsmath} for example modifies \LaTeX internals and handles math accents differently.

We thus needed to double our \texttt{\let}'s as, if \texttt{amsmath} is loaded, the cs with space will exist but not be paired in expected way with the original cs. This breaks things by the way if some math accent is written to an external file under a certain context and executed in another context. The new context will be probably ignored if \texttt{amsmath} is loaded, as the external file will have an already expanded-once meaning.

Some macros with space in name might thus be created as \texttt{\relax}. Should I rather create \texttt{\protected} macros for the math accents with Unicode engines? Anyway, the construct does give good result with the few OpenType text fonts I tested.
\expandafter\xdef\csname mst@dot@mv#1\endcsname\relax\% \
" -> \ddot \\
\expandafter\xdef\csname mst@ddot@mv#1\endcsname\relax\% \\
% " -> \mathring \\
\expandafter\xdef\csname mst@mathring@mv#1\endcsname\relax\% \\
% ^ -> \hat \\
\expandafter\xdef\csname mst@hat@mv#1\endcsname\relax\% \\
% ~ -> \tilde \\
\expandafter\xdef\csname mst@tilde@mv#1\endcsname\relax\% \\
\else \% false branch of ifmst@unimathaccents \\
\AtEndOfPackage{\AtBeginDocument{% \\
\@tfor\@tempa:={grave}{acute}{check}{breve}{bar}\
{dot}{ddot}{mathring}{hat}{tilde}\
\do\expandafter\let\127
\endcsname

1.3u used some \def but this made the accent macro meanings look slightly different depending on whether the math version being set-up was with an 8bit encoding or TU encoding. For the sake of uniform treatment we modify this at 1.3v, but this is a bit complicated regarding timing: we need, in absence of unimathaccents option, in math versions with an OpenType font, to let the \acute etc... acquire back some prior non-mathastext meanings. To allow maximal flexibility, these original meaning get stored at begin document only. But \mst@nonsubduedmathaccents assigns to \acute etc... (in the robust sense with \LaTeX\ 2019-10-01 or later) the meaning stored in the macros with the math version in their names. Such \mst@acute@mnormal etc... must thus be ready before \mst@nonsubduedmathaccents (or at least before the last such) execution: the code here must get executed after the definition of the «original»-named macros but prior to the (last one, if multiple) \mst@nonsubduedmathaccents.

Hence 1.3v delayed a bit the initial execution of this macro (see further down in the code) compared to what happened in 1.3u.

We are in a group but \AtEndOfPackage does the right thing.
This is needed because the pdflatex engine branch will use `\DeclareMathAccent` and it creates robust macros with \LaTeX\ 2019-10-01 or later. As we want elsewhere in the package code not to have to check if under Unicode engine or not, we need to handle here also some definition of robust macros.

But wouldn't it be simpler to manage `\protected` macros?

`\ifmst@robust@obsessed@LaTeX@era`

`\@tfor\@tempa:={grave}{acute}{check}{breve}{bar}{dot}{ddot}{mathring}{hat}{tilde} do`

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string\endcsname} {}` % \v -> \check

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@bar\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string\endcsname} {}` % \. -> \dot

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@grave\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string`\endcsname} {}` % \` -> \grave

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@acute\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string'\endcsname} {}` % \' -> \acute

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@check\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string\v\endcsname} {}` % \v -> \check

`\expandafter\mst@DeclareMathAccent\expandafter`

`\csname mst@breve\@tempa\endcsname{\mathalpha}{mtoperatorfont}{\csname#2\string\u\endcsname} {}` % \u -> \breve

% \= -> \bar

% \. -> \dot
The `\MTDeclareVersion` command is to be used in the preamble to declare a math version. A more complicated variant would also specify a choice of series for the Euler and Symbol font: anyhow Symbol only has the medium series, and Euler has medium and bold, so what is lacking is the possibility to create a version with a bold Euler. There is already one such version: the default bold one. And there is always the possibility to add to the preamble `\SetSymbolFont{mteulervm}{versionname}{U}{zeur}{bx}{n}` if one wants to have a math version with bold Euler characters.

For version 1.1 we add an optional parameter specifying the shape to be used for letters. Note: (2012/10/24) I really should check whether the user attempts to redefine the ‘normal’ and ‘bold’ versions and issue a warning in that case! Finally done at 1.3w 2019/11/16! Better late than never...

1.3c (2013/12/14) adds an extra optional parameter after all previous ones, to inherit the settings from another version. Typically to be used with [bold]. I take this opportunity to sanitize a bit some line endings to avoid generating (in the preamble, document macros were already careful of course) too many space tokens, at least inside macros. And I modify (correct? perhaps it was on purpose) the strange way I used `\@onlypreamble` in earlier version.

1.3u adds storage of macros holding the needed meanings for `\imath`, `\hbar`, math accents, and the minus symbol, version wise.

1.3w adds the check to forbid normal and bold as version names.
\MTDoNotDeclareVersion\expandafter\gobblefour
\fi
\relax\DeclareMathVersion{mst@version}\MTDeclareVersion
\newcommand*\MTDoNotDeclareVersion@[1][{}%
\PackageWarningNoLine{mathastext}{You asked to declare a version with name
``\mst@version''.~"J%\@spaces Sorry but you are not allowed to do that.~"J%
\@spaces \ifmst@subdued Anyway the `subdued' option is in force\else
Use rather \string\Mathastext\space macro (with no optional argument)\fi
\MTDoNotDeclareVersion@%}
\newcommand*\MTDeclareVersion@[1][]{%
\edef\mst@tmp{#1}%
\ifx\mst@tmp\empty
\else
\global\expandafter\let\csname mv@\mst@version\expandafter\endcsname\csname mv@#1\endcsname
\PackageInfo{mathastext}{Math version `\mst@version'\MessageBreak
declared\on@line\MessageBreak
inherits from `#1'\@gobble}\fi
\fi
\expandafter\MTDeclareVersion@@\mst@declareversionargs%}
\newcommand*\MTDeclareVersion@@[6][]{%
\expandafter\edef\csname mst@encoding@\mst@version\endcsname{#3}%
\expandafter\edef\csname mst@family@\mst@version\endcsname{#4}%
\expandafter\edef\csname mst@series@\mst@version\endcsname{#5}%
\expandafter\edef\csname mst@shape@\mst@version\endcsname{#6}%
\expandafter\edef\csname mst@boldvariant@\mst@version\endcsname{\mst@bold}%
\expandafter\edef\csname mst@itdefault@\mst@version\endcsname{\itdefault}%
\expandafter\edef\csname mst@rmdefault@\mst@version\endcsname{\rmdefault}%
\expandafter\edef\csname mst@sfdefault@\mst@version\endcsname{\sfdefault}%
\expandafter\edef\csname mst@ttdefault@\mst@version\endcsname{\ttdefault}%
\expandafter\edef\csname mst@exists@skip@\mst@version\endcsname{\mst@exists@skip}%
\expandafter\edef\csname mst@forall@skip@\mst@version\endcsname{\mst@forall@skip}%
\expandafter\edef\csname mst@prime@skip@\mst@version\endcsname{\mst@prime@skip}%
\def\mst@tmp{#1}%
\ifx\mst@tmp\empty
\ifmst@italic
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{\mst@ltsh}%
\immediate\write\m@ne{}%
\PackageInfo{mathastext}{Latin letters in math version `#2'\string'\MessageBreak
will use the font #3/#4/#5/\mst@ltsh}\fi%
\immediate\write\m@ne{}%
\PackageInfo{mathastext}{Latin letters in math version `#2'\string'\MessageBreak
will use the font #3/#4/#5/\mst@ltsh}\fi%
\else
\UseSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{\mst@ltsh}%
\fi
\fi\mst@bold
\ifmst@frenchmath\space(uppercase: #6)\fi
\endinput
will be in \texttt{#6}\string shape\@gobble}\
\immediate\write\m@ne{}
\else
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#6}\
\immediate\write\m@ne{}
\PackageInfo{mathastext}{Latin letters in math version \texttt{#2}\string\MessageBreak}
\message{on@line}{\MessageBreak}
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}\
\immediate\write\m@ne{}
\PackageInfo{mathastext}{Latin letters in math version \texttt{#2}\string\MessageBreak}
\message{on@line}{\MessageBreak}
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}\fi
\else % #1 not empty
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}\
\immediate\write\m@ne{}
\PackageInfo{mathastext}{Latin letters in math version \texttt{#2}\string\MessageBreak}
\message{on@line}{\MessageBreak}
\SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}\
\ifmst@frenchmath\space(uppercase: #6)\fi\MessageBreak
Other characters (digits, ...) and\MessageBreak
\protect\log-like names\MessageBreak
will be in \texttt{#6}\string shape\@gobble}\
\immediate\write\m@ne{}
\expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#6}\fi
\SetMathAlphabet{\Mathnormalbold}{#2}{#3}{#4}{\mst@bold}\
\SetMathAlphabet{\Mathbf}{#2}{#3}{#4}{\mst@bold}{#6}\
\SetMathAlphabet{\Mathit}{#2}{#3}{#4}{\itdefault}\
\SetMathAlphabet{\Mathsf}{#2}{#3}{#5}{#6}\
\ifmst@needeuler
\SetMathAlphabet{\MathEulerBold}{#2}{U}{zeur}{\mst@bold}{n}\fi
In the case of option LGRgreek (selfGreeks), it is expected that the fonts used in each math versions exist in LGR (OT1) encoding. We first recalculate the shapes to be used for lowercase and uppercase Greek letters depending on the frenchmath and [it/up][g/G]reek options as well as on the (local to this version) shapes for letters and digits.

1.3x replaces \updefault by \MTgreekupdefault and \itdefault by \MTgreekitdefault. It also prepares to store two Boolean settings saying whether lowercase respectively uppercase Greek letters will use ‘upright’ or ‘italic’ (LGRgreek(s) only).

The 1.3x refactoring of LGRgreek is done via a decoupling, thus things are done here under selfGreek or other Greek options which ultimately serve nothing and conversely things are done here for LGRgreek which are superfluous.
1.3za refactoring to reduce code duplication; I briefly considered trying to enhance \MTgreek\-font to work also with LGRgreeks and selfGreeks but I have dropped the idea for now.

\ifmst@LGrgreeks \def\mst@greekfont{#4}\fi
\ifmst@selfGreeks\def\mst@greekfont{#4}\fi
\ifmst@LGrgreek
\SetSymbolFont{mtgreekit}{#2}{LGR}{\mst@greekfont}{#5}{\MTgreekitdefault}%
\SetSymbolFont{mtgreekit}{#2}{LGR}{\mst@greekfont}{#5}{\MTgreekitdefault}%
\SetMathAlphabet{\mathgreekitbold}{#2}{LGR}{\mst@greekfont}{#5}{\MTgreekitdefault}%
\SetMathAlphabet{\mathgreekitbold}{#2}{LGR}{\mst@greekfont}{#5}{\MTgreekitdefault}%
This is where the shape of uppercase/lowercase Greek letters is recorded, for \MTversion’s triggered \MTcustomgreek to do the right thing.

\let\csname ifmst@greek@\mst@version @upper@up\expandafter\endcsname\csname ifmst@greek@upper@up\endcsname
\let\csname ifmst@greek@\mst@version @lower@up\expandafter\endcsname\csname ifmst@greek@lower@up\endcsname
\immediate\write\m@ne{}
\PackageInfo{mathastext}{Greek letters (upper: `\ifmst@greek@upper@up\MTgreekupdefault\else\MTgreekitdefault\fi\string', lower: `\ifmst@greek@lower@up\MTgreekupdefault\else\MTgreekitdefault\fi\string')\MessageBreak
will use font family `\ifmst@greekfont\string' (LGR)\MessageBreak
in mathastext\MessageBreak
math version `\mst@version\string'\MessageBreak
declared}%
\immediate\write\m@ne{}
\else
\ifmst@sselfGreek
\SetSymbolFont{mselfGreekfont}{#2}{OT1}{\mst@greekfont}{#5}{\mst@greek@ush@loc}%
\PackageInfo{mathastext}{Capital Greek letters (shape `\ifmst@greek@ush@loc\string'
will use the font\MessageBreak
family `\ifmst@greekfont\string' (OT1) in mathastext\MessageBreak
math version `\mst@version\string' declared}%
\immediate\write\m@ne{}
\else
\fi
\edef\mst@tmp{\expandafter\noexpand\csname mst@hbar@mv#2\endcsname\expandafter\noexpand\csname mst@ltbar@mv#2\endcsname}
\expandafter\mst@dothe@hbarstuff\mst@tmp{#3}
\edef\mst@tmp{\expandafter\noexpand\csname mst@inodot@mv#2\endcsname\expandafter\noexpand\csname mst@jnodot@mv#2\endcsname}
\expandafter\mst@dothe@inodotstuff\mst@tmp{#3}
\ifmst@mathaccents
\mst@dothe@mathaccentsstuff{#2}{#3}%
\fi
\edef\mst@tmp{\expandafter\noexpand\csname mst@minus@mv#2\endcsname\expandafter\noexpand\csname mst@varfam@minus@mv#2\endcsname}
\ifmst@endash
\expandafter\mst@dothe@endashstuff\mst@tmp{#3}
\else
\ifmst@emdash
\expandafter\mst@dothe@emdashstuff\mst@tmp{#3}
\else
\expandafter\mst@dothe@hyphenstuff\mst@tmp
\fi
\fi
\fi
\edef\mst@tmp{\expandafter\noexpand\csname mst@odot@mathaccentsstuff\endcsname}
\expandafter\mst@dothe@mathaccentsstuff{#2}{#3}%
\fi
\edef\mst@tmp{\expandafter\noexpand\csname mst@odot@minus@mv\endcsname\expandafter\noexpand\csname mst@varfam@minus@mv\endcsname}
\ifmst@endash
\expandafter\mst@dothe@endashstuff\mst@tmp{#3}%
\else
\ifmst@emdash
\expandafter\mst@dothe@emdashstuff\mst@tmp{#3}%
\else
\expandafter\mst@dothe@hyphenstuff\mst@tmp
\fi
\fi
\fi
\MTDeclareVersion@@
\let\MathastextDeclareVersion\MTDeclareVersion
This is a wrapper around \LaTeX's \texttt{\textbackslash mathversion}: here we have an optional argument allowing a quick and easy change of the text fonts additionally to the math fonts. Present already in the initial version of the package (January 2011.)

1.15: some modifications for the subdued option vs LGRgreek and for the math muskips after \texttt{\textbackslash exists} and \texttt{\textbackslash forall}.

1.2: with the subdued option sets the math alphabets in the normal and bold math versions do not apply to operator names and non-alphabetical symbols. The switch for braces is left as it is.

1.2b: with the subdued option, the italic corrections are not added. Else, we check the shape of letters in this version. Also, there was a bug since 1.15: the values of the math skips were taken not from the settings for the math version (#2) but from those of the optional argument (#1), if present...

1.3: activation of italic corrections is now separated from actual math activation of letters.

1.3c: a starred variant is added which does not modify the text fonts, only the math set-up.

1.3d: replaced in \texttt{\textbackslash MTversion} things like \texttt{\textbackslashedef\mst@encoding{...}} and \texttt{\textbackslash renewcommand{\textbackslash encodingdefault}{...}} etc... All those \texttt{\mst@@...} things were useless. I also redefine \texttt{\textbackslash seriesdefault} rather than \texttt{\textbackslash mdefault}.

1.3d: mechanism of restoration of Greek in subdued normal and bold versions has been to all cases, and not only for the LGRgreek option.

1.3u: version savvy (i.e. font-encoding savvy) minus sign, \texttt{\textbackslash hbar}, \texttt{\textbackslash imath}, math accents.

1.3x: Booleans recovered from stored data in the math version will configure the things \texttt{\textbackslash MTcustomgreek} do, under LGRgreek option.

2166 \texttt{\newcommand*{\MTversion}{\ifstar\MTversion@s\MTversion@}}}
2167 \texttt{\newcommand*{\MTversion@s}{[1]{{\mathversion{#1}}\MTversion@@}}}
2168 \texttt{\newcommand*{\MTversion@}{[2][]{{%}}}
2169 \texttt{\mathversion{#2}}\% defines \texttt{\mathversion} as expanded #2
2170 \texttt{\edef\mst@tmp{#1}\%}
2171 \texttt{\ifx\mst@tmp\empty}
2172 \texttt{\let\mst@tmp\math@version}
2173 \texttt{\else}
2174 \texttt{\let\mst@tmp\mst@tmpa}
2175 \texttt{\fi}
2176 \texttt{\edef\encodingdefault{\csname mst@encoding@\mst@tmp\endcsname}\%}
2177 \texttt{\edef\familydefault{\csname mst@family@\mst@tmp\endcsname}\%}
2178 \texttt{\edef\seriesdefault{\csname mst@series@\mst@tmp\endcsname}\%}
2179 \texttt{\edef\shapedefault{\csname mst@shape@\mst@tmp\endcsname}\%}
2180 \texttt{\edef\bfdefault{\csname mst@boldvariant@\mst@tmp\endcsname}\%}
2181 \texttt{\edef\itdefault{\csname mst@itdefault@\mst@tmp\endcsname}\%}
2182 \texttt{\edef\rmdefault{\csname mst@rmdefault@\mst@tmp\endcsname}\%}
2183 \texttt{\edef\sfdefault{\csname mst@sfdefault@\mst@tmp\endcsname}\%}
2184 \texttt{\edef\ttdefault{\csname mst@ttdefault@\mst@tmp\endcsname}\%}
2185 \texttt{\usefont{\encodingdefault}{\familydefault}{\seriesdefault}{\shapedefault}\%}
2186 \texttt{\MTversion@@\% \MTversion@}
2187 \texttt{\MTversion@}

1.3j has a stronger subdued which does \texttt{\MTnormalprime}, \texttt{\MTnormalexists}, \texttt{\MTnormalforall} rather than setting the skips to 0mu. Hence \texttt{\MTversion} by default should do \texttt{\MTprimedoesskip}, \texttt{\MTexistsdoesskip}, \texttt{\MTforalldoesskip}.

1.3u drops the argument, as the info is in \texttt{\mathversion} from \LaTeX\texttt{2e} code.
\newcommand*{\MTversion@@}{% 
\MTexistsdoesskip
\MTforalldoesskip
\MTprimoesskip
v1.15e: muskips.
\mst@exists@muskip\csname mst@exists@skip@\math@version\endcsname\relax
\mst@forall@muskip\csname mst@forall@skip@\math@version\endcsname\relax
v1.2: muskip for \prime.
\mst@prime@muskip\csname mst@prime@skip@\math@version\endcsname\relax
v1.2b: italic corrections except for italic/slanted (sic) letters, and of course except in the subdued normal and bold math versions.
v1.3: by default, letters are made mathematically active, even if italic corrections are not used, to allow the action of \MTsetmathskips.
\edef\mst@tmpa{\csname mst@ltshape@\math@version\endcsname}\
\edef\mst@tmpb{\csname mst@shape@\math@version\endcsname}\
v1.15c: extending subdued to LGRgreek.
v1.15f: subduing math alphabets in a simpler way than in 1.15e.
v1.2b: subduing the activation of characters in math mode.
v1.2d: special treatment of the asterisk.
v1.3d: extended LGRgreek mechanism of activation/restoration of Greek to all cases.
v1.3j: use of \MTeverymathdefault, which includes \MTicinmath, but must be corrected then according to shape of letters and presence or absence of option frenchmath. We do only \def\mst@ITcorr{\ifnum\fam=\m@ne\slash\fi} and not \MTICinmath to not overwrite some user-defined \MTeverymathdefault. Code for italic corrections or not according to letter shape is executed after \MTeverymathdefault which limits a bit user customizing possibilities, but if I moved it later, I would possibly have to put inside the \MTicinmath the check for it or sl. Similary the \MTcustomgreek always executed (if not subdued).
\MTmathoperatorsobeymathxx
\MTeverymathdefault
\MTcustomizenewmcodes
\@for\mst@tmpc:=it,sl\do{\ifx\mst@tmpc\mst@tmpa\MTnoicinmath\fi}\
\ifmst@frenchmath
\@for\mst@tmpc:=it,sl\do{\ifx\mst@tmpc\mst@tmpb\MTnoICinmath\fi}\
\fi
1.3j has a stronger subdued which does \MTnormalprime, \MTnormalexists, \MTnormalforall rather than simply setting the skips to 0mu. Note: \MTnormalprime is done as part of \MTeverymathoff.
The subdued mode does not undo the effect of the frenchmath option on uppercase Latin letters: they will use the same shape as digits and operator names! (This should have been made more prominent in user manual more than ten years ago, but is done only today 2023/12/28...).
\@ifmt@subdued
\@ifx\math@version\mst@normalversionname
\mst@restorealphabets
\MTstandardgreek
\MTmathoperatorsdonotobeymathxx
\MTnormalexists
\MTnormalexists
1.3t adds better compatibility with subdued mode for \textit{imath}/\textjmath and perfect compatibility for the minus sign.

1.3u extends this further to allow per-math-version meanings for them.

\begin{verbatim}
2214 \mst@subduedhbar
2215 \mst@subduedinodot
2216 \mst@subduedmathaccents
2217 \mst@subduedminus
2218 \else
2219 \ifx\math@version\mst@boldversionname
2220 \mst@restorealphabets
2221 \MTstandardgreek
2222 \MTmathoperatorsdonotobeymathxx
2223 \MTnormalexists
2224 \MTnormalforall
2225 \MTeverymathoff
2226 \MTresetnewmcodes
2227 \mst@subduedhbar
2228 \mst@subduedinodot
2229 \mst@subduedmathaccents
2230 \mst@subduedminus
2231 \else
2232 \mst@setalphabets
2233 \endverbatim

1.3x addition for \textsl{MTcustomgreek} under LGRgreek option.

\begin{verbatim}
2233 \expandafter\let\csname ifmst@greek@upper@up\expandafter\endcsname
2234 \csname ifmst@greek@math@version@upper@up\endcsname
2235 \expandafter\let\csname ifmst@greek@lower@up\expandafter\endcsname
2236 \csname ifmst@greek@math@version@lower@up\endcsname
2237 \MTcustomgreek
2238 \mst@nonsubduedhbar
2239 \mst@nonsubduedinodot
2240 \mst@nonsubduedmathaccents
2241 \mst@nonsubduedminus
2242 \fi
2243 \fi
2244 \else
2245 \endverbatim

1.3x addition for \textsl{MTcustomgreek} under LGRgreek option.

\begin{verbatim}
2245 \expandafter\let\csname ifmst@greek@upper@up\expandafter\endcsname
2246 \csname ifmst@greek@math@version@upper@up\endcsname
2247 \expandafter\let\csname ifmst@greek@lower@up\expandafter\endcsname
2248 \csname ifmst@greek@math@version@lower@up\endcsname
2249 \MTcustomgreek % new with 1.3d
2250 \mst@nonsubduedhbar
2251 \mst@nonsubduedinodot
2252 \mst@nonsubduedmathaccents
2253 \mst@nonsubduedminus
\end{verbatim}

136
\texttt{\MTWillUse} This is a preamble-only command, which can be used more than once, only the latest one counts. Sets up the math fonts in the normal and bold versions, as does \texttt{\Mathastext}.

\texttt{\MTWillUse} \begin{Verbatim}
\newcommand*{\MTWillUse}[5]{
  \MTencoding{#2}
  \MTfamily{#3}
  \MTseries{#4}
  \MTshape{#5}
  \ifmst@italic\MTlettershape{\itdefault}\fi % was missing in v 1.14 and prior
  \edef\mst@tmp{#1}
  \ifx\mst@tmp\empty\else\MTlettershape{#1}\fi
  \Mathastext}
\end{Verbatim}

\texttt{\Mathastext} \begin{Verbatim}
The command \texttt{\Mathastext} can be used anywhere in the preamble and any number of time, the last one is the one that counts.

In version 1.1 we have two fonts: they only differ in shape. The \texttt{mtletterfont} is for letters, and the \texttt{mtoperatorfont} for digits and log-like operator names. The default is that both are upright.

Starting with version 1.12, an optional argument makes \texttt{\Mathastext} act as the declaration of a math version, to be later used in the document.

Versions 1.15x brought some adaptations related to the subdued option.

1.3c adds a second optional parameter to inherit previous settings from another version; mostly done to inherit the bold version fonts for symbols and large symbols. This is done in \texttt{\MTDeclareVersion}.

1.3j moves the code related to \texttt{\MTicinmath} from \texttt{\Mathastext@} to \texttt{\AtBeginDocument} (code depending on whether \texttt{subdued} option in use). But we omit for this from \texttt{\MTicinmath} the \texttt{\MTmathactiveletters} and issue the latter during loading of package, hence allowing \texttt{\MTmathstandardletters} to be effective in the preamble.

I forgot to document that under \texttt{subduced} option the \texttt{\Mathastext} command without optional parameter does not any \texttt{\SetSymbolFont} etc... but it has a few other tasks to complete nevertheless.

1.3u fixes some long-standing bug that \texttt{\Mathastext} did not repeat some font-encoding dependent things: they got done only once during package loading (things regarding the \texttt{\hbar}, \texttt{\imath}, the math accents and the minus sign). They are now part of the contents of \texttt{\Mathastext} macro itself (which is executed during package loading).

1.3x has refactored the \texttt{LGRgreek} associated math fonts.

\end{Verbatim}
\else\expandafter\@secondoftwo
\fi
\Mathastext@
\{\MTDeclareVersion[\mst@ltsh]{#1}{\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}\%
\Mathastext@declare
\def\Mathastext@ {%
\mst@update@greeksh
\edef\mst@encoding@normal{\mst@enc}%
\edef\mst@family@normal{\mst@fam}%
\edef\mst@series@normal{\mst@ser}%
\edef\mst@shape@normal{\mst@opsh}%
\edef\mst@ltshape@normal{\mst@ltsh}%
\edef\mst@itdefault@normal{\itdefault}%
\edef\mst@rmdefault@normal{\rmdefault}%
\edef\mst@sfdefault@normal{\sfdefault}%
\edef\mst@ttdefault@normal{\ttdefault}%
\edef\mst@boldvariant@normal{\mst@bold}%
\edef\mst@exists@skip@normal{0mu}%
\edef\mst@forall@skip@normal{0mu}%
\edef\mst@prime@skip@normal{0mu}%
\edef\mst@exists@skip@bold{0mu}%
\edef\mst@forall@skip@bold{0mu}%
\edef\mst@prime@skip@bold{0mu}%
\else % not subdued
\ifmst@italic
\ifmst@frenchmath
\mst@exists@muskip\mst@exists@skip\relax
\mst@forall@muskip\mst@forall@skip\relax
\mst@prime@muskip\mst@prime@skip\relax
\else % not subdued
\ifmst@italic
\ifmst@frenchmath
\mst@exists@muskip\mst@exists@skip\relax
\mst@forall@muskip\mst@forall@skip\relax
\mst@prime@muskip\mst@prime@skip\relax
\else % not subdued

Here and elsewhere 1.3za has removed usage of an \texttt{\ifmst@nonormalbold} conditional which was added at 1.15f.

\SetMathAlphabet{\mathnormalbold}{normal}{\mst@encoding@normal}%
\{\mst@family@normal\%
\{\mst@boldvariant@normal\%
\{\mst@ltshape@normal\%
\SetMathAlphabet{\mathnormalbold}{bold}{\mst@encoding@bold}%
\{\mst@family@bold\%
\{\mst@boldvariant@bold\%
\{\mst@ltshape@bold\%
\SetSymbolFont{mtletterfont}{normal}{\mst@encoding@normal}%
\{\mst@family@normal\%
\{\mst@series@normal\%
\{\mst@ltshape@normal\%
\SetSymbolFont{mtletterfont}{bold}{\mst@encoding@bold}%
\{\mst@family@bold\%
\{\mst@series@bold\%
\{\mst@ltshape@bold\%
\SetSymbolFont{mtoperatorfont}{normal}{\mst@encoding@normal}%
\{\mst@family@normal\%
\{\mst@series@normal\%
\{\mst@shape@normal\%
\SetSymbolFont{mtoperatorfont}{bold}{\mst@encoding@bold}%
\{\mst@family@bold\%
\{\mst@series@bold\%
\{\mst@shape@bold\%

1.3za removes the 1.15f added conditional checks.
\SetMathAlphabet{\Mathit}{normal}{\mst@encoding@normal}\%
\SetMathAlphabet{\Mathit}{bold}{\mst@encoding@bold}\%
\SetMathAlphabet{\Mathsf}{normal}{\mst@encoding@normal}\%
\SetMathAlphabet{\Mathsf}{bold}{\mst@encoding@bold}\%
\SetMathAlphabet{\Mathtt}{normal}{\mst@encoding@normal}\%
\SetMathAlphabet{\Mathtt}{bold}{\mst@encoding@bold}\%
\fi % de \ifmst@subdued
\MathEulerBold 1.14c: We reset mteulervm and \MathEulerBold here as the variant for bold may have been changed by the user via \Mathastextboldvariant{m}; and we should keep this local to math versions.
\ifmst@need euler
\SetSymbolFont{mteulervm}{bold}{U}{zeur}{\mst@boldvariant@normal}{n}\%
\SetMathAlphabet{\MathEulerBold}{normal}{U}{zeur}{\mst@boldvariant@bold}{n}\%
\fi
\ifmst@needsymbol\SetSymbolFont{mtpsymbol}{bold}{U}{psy}{\mst@boldvariant@bold}{n}\%
\fi

1.3x has refactored the LGRgreek associated math fonts.
1.3za adds the math alphabets \mathgreekitbold and \mathgreekupbold. And it executes this code also in subdued mode, because anyhow the symbolfonts mtgreekit and mtgreekup and associated alphabets have been declared also, at time of loading the package, so not doing it here means that effect of \MTgreekfont would be ignored; which was probably a bug. And

\LGRgreek\ LGRgreek, LGRgreeks, selfGreek, and selfGreeks options.
\selfGreek* 1.3x has refactored the LGRgreek associated math fonts.
by the way, documentation says $\text{\textbackslash MTgreekfont}$ has no effect under LGRgreek and selfGreeks option so we need to enforce it here (for time being).

2404 \ifmst@LGRgreek \edef\mst@greekfont{\mst@fam}\fi
2405 \ifmst@selfGreeks\edef\mst@greekfont{\mst@fam}\fi
2406 \ifmst@LGRgreek
2407 \SetSymbolFont{mtgreekup}{normal}{LGR} %
2408 \SetSymbolFont{mtgreekup}{bold}{LGR} %
2409 \SetSymbolFont{mtgreekit}{normal}{LGR} %
2410 \SetSymbolFont{mtgreekit}{bold}{LGR} %
2411 \SetMathAlphabet{\mathgreekup}{normal}{LGR} %
2412 \SetMathAlphabet{\mathgreekup}{bold}{LGR} %
2413 \else
2414 \ifmst@selfGreek %
2415 \SetSymbolFont{mtselfGreekfont}{normal}{OT1} %
2416 \SetSymbolFont{mtselfGreekfont}{bold}{OT1} %
2417 \fi
2418 \fi
2419 \fi
2420 \fi
2421 \fi
2422 \else
2423 \ifmst@subdued %
2424 \fi
2425 \else
2426 \ifmst@frenchmath %
2427 \fi
2428 \fi
2429 \fi
2430 \else
2431 \\mst@infoline{Latin letters in the `normal\textquoteleft, resp. `bold\textquoteleft,}\%
2432 \\mst@infoline{math versions are now set up to use the fonts}\%
2433 \\mst@infoline{\mst@encoding@normal/\mst@family@normal/\mst@series@normal\%
2434 \\mst@encoding@normal/\mst@family@normal/\mst@boldvariant@normal\%
2435 \\mst@encoding@normal/\mst@family@normal/\mst@ltshape@normal, resp.\%
2436 \\mst@encoding@normal/\mst@family@normal/\mst@ltshape@normal.}\%
2437 \\infoline{\textbackslash frenchmath\mst@infoline{\textbackslash uppercase: \mst@shape@normal}}}\fi
2438 \\ifmst@LGRgreek
2439 \\mst@infoline{Greek letters (upper:\%
2440 `\textbackslash ifmst@greek@upper\textbackslash up\textbackslash MTgreekupdefault\else\textbackslash MTgreekupdefault\textbackslash fi\textbackslash string',\%
2441 lower:\%
2442 `\textbackslash ifmst@greek@lower\textbackslash up\textbackslash MTgreekupdefault\else\textbackslash MTgreekupdefault\textbackslash fi\textbackslash string')\%
2443 will use font\%}
2444 \\mst@infoline{family `\textbackslash mst@greekfont\textbackslash string' (LGR).}}\%
2445 \\else
2446 \\mst@infoline{\textbackslash family `\textbackslash mst@greekfont\textbackslash string' (LGR).}}\%
2447 \\else
2448 \\mst@infoline{\textbackslash self\textbackslash Greek
2449
141
\mst@infoline{Capital Greek letters (shape `\mst@greek\string') will use font}\%
\mst@infoline{family `\mst@greekfont\string' (OT1).}\%
\fi
\fi
\ifmst@nodigits\else
\mst@infoline{Other characters (digits, ...) and \string\log-like names will be}\
\mst@infoline{typeset with the \mst@shape@normal\space shape.}\%
\fi
\fi
\% end of else branch \ifmst@subdued
\ifmst@nohbar\else
\mst@infoline{\string\hbar}\%
\mst@dothe@hbarstuff
\mst@hbar@mvnormal\mst@ltbar@mvnormal\mst@encoding@normal
\let\mst@hbar@mvbold\mst@hbar@mvnormal
\fi
\mst@dothe@inodotstuff\inodot\jnodot\mst@encoding@normal
\let\mst@inodot@mvnormal\inodot
\let\mst@inodot@mvbold\inodot
\let\mst@jnodot@mvnormal\jnodot
\let\mst@jnodot@mvbold\jnodot
\ifmst@mathaccents
\mst@infoline{math accents}\%
\mst@dothe@mathaccentsstuff\normal\mst@encoding@normal
\fi
\ifmst@nominus\else
\mst@infoline{minus as endash}\%
\ifmst@endash
\mst@dothe@endashstuff\mst@minus@mvnormal\mst@varfam@minus@mvnormal
\mst@encoding@normal
\mst@dothe@endashstuff\mst@minus@mvbold\mst@varfam@minus@mvbold
\mst@encoding@normal
\else
\ifmst@emdash
\mst@dothe@emdashstuff\mst@minus@mvnormal\mst@varfam@minus@mvnormal
\mst@encoding@normal
\mst@dothe@emdashstuff\mst@minus@mvbold\mst@varfam@minus@mvbold
\mst@encoding@normal
\else
\mst@dothe@hyphenstuff\mst@minus@mvnormal\mst@varfam@minus@mvnormal
\let\mst@minus@mvbold\mst@minus@mvnormal
\let\mst@varfam@minus@mvbold\mst@varfam@minus@mvnormal
\fi
\fi
\fi
\fi
\% minus

1.3zb moves this info line last and also explicitly mentions italic or frenchmath (here and at some other locations above).

\ifmst@subdued
\mst@infoline{Subdued `normal\string' and `bold\string' math versions.}\%
\fi
Additional appropriate messages to the terminal and the log.

Math sizes

I took the code for \Huge and \HUGE from the moresize package of Christian Cornelissen

Math sizes

I chose rather big subscripts.
\DeclareMathSizes{\@xviipt}{\@xviipt}{\@xivpt}{\@xiipt}
\DeclareMathSizes{\@xxpt}{\@xxpt}{\@xviipt}{\@xivpt}
\DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xviipt}
\DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxvpt}{\@xxpt}
\DeclareMathSizes{\@xxxvipt}{\@xxxvipt}{\@xxxpt}{\@xxvpt}
\mst@infoline{mathastext has declared larger sizes for subscripts.}
\mst@infoline{To keep LaTeX defaults, use option `defaultmathsizes\string'.}
\fi
\MTeverymathoff
\newcommand*{\MTeverymathoff}{%}
\\MTnormalasterisk
\\MTnormalprime
\\MTnonlettersdonotobeymathxx
\\MTeasynonlettersdonotobeymathxx
\\MTmathstandardletters
\\MTdonotfixfonts
\AtBeginDocument{\%}
\\ifpackageloaded{hyperref}{\def\Hurl{\begingroup\MTeverymathoff\Url}}\%
\\iffalse\declareurlcommand\url{\MTeverymathoff}\fi\%
\AtBeginDocument{\%}
\\\ifpackageloaded{url}{\DeclareUrlCommand\url{\MTeverymathoff}}\%
\\AtBeginDocument{\%}
\MTeverymathdefault
\newcommand*{\MTeverymathdefault}{%}
\\MTactiveasterisk
\\MTprimedoesskip
\\MTeasynonlettersobeymathxx
\\MTicinmath
\\MTfixfonts
1.3i 2016/01/06 Compatibility patch with \url from url.sty and \url/nolinkurl from hyperref.sty.
1.3j 2016/01/15 renamed the macro from \MTactivemathoff to \MTeverymathoff, as it is not exclusively a matter of math active characters due to \MTeasynonlettersdonotobeymathxx.
1.3o 2016/05/03 adds \MTdonotfixfonts. Operant with LuaLaTeX only.
\newcommand*{\MTeverymathdefault}{%}
\\MTactiveasterisk
\\MTprimedoesskip
\\MTeasynonlettersobeymathxx
\\MTicinmath
\\MTfixfonts
1.3i 2016/01/06 Customizable command which gets executed by \MTversion except when switching to normal/bold if option subdued. The included \MTicinmath does \MTmathactiveletters which will also activate the math skips around letters.
The \MTeverymathdefault does not include \MTmathoperatorsobeymathxx as the latter does not correspond to something done during execution of \the\everymath.
Should I put \let\newmcodes@\mst@newmcodes@ here too? No, it is not done at everymath. During the loading, the (non subdued) package does \MTactiveasterisk (if option asterisk), \MTprimedoesskip, \MTeasynonlettersobeymathxx and \MTmathactiveletters. There is some code at begin document for decisions about italic corrections, this code does not emit again \MTmathactiveletters, hence a \MTmathstandardletters in the preamble is not overruled. Furthermore the at begin document code will not overrule user emitted \MTnoicinmath etc... commands in the preamble.
And user can employ \MTnormalexists, etc..., from inside the preamble, it will not be overruled (as it is delayed at begin document to after mathastext dealings).
1.3o 2016/05/03 adds \MTfixfonts. Operant with LuaLaTeX only.
Things to do last "at begin document"

\AtBeginDocument{\everymath\expandafter{\the\everymath
\mst@the\mst@do@nonletters \let\mst@the\@gobble
\mst@theeasy\mst@do@easynonletters \let\mst@theeasy\@gobble
\mst@the\mst@do@az \let\mst@the\@gobble
\mst@the\mst@do@AZ \let\mst@the\@gobble}
\everydisplay\expandafter{\the\everydisplay
\mst@the\mst@do@nonletters \let\mst@the\@gobble
\mst@theeasy\mst@do@easynonletters \let\mst@theeasy\@gobble
\mst@the\mst@do@az \let\mst@the\@gobble
\mst@the\mst@do@AZ \let\mst@the\@gobble}

1.3j: moved here to be executed at begin document (and not from inside \Mathastext@.)

The \MTeverymathoff does: \MTnormalasterisk, \MTnormalprime, \MTnonlettersdonotbeymathx, \MTeasynonlettersdonotbeymathx, \MTmathstandardletters.

1.3m: doing \MTmathactiveletters in subdued mode immediately after \begin{document} resulted in errors because \mst@itcorr had been left undefined. We thus add \MTnoicinmath to the subdued initialization.

Since 1.3n there is \MTresetnewmcodes which needs \mst@originalnewmcodes@, itself defined at begin document. Thus we have wrapped the whole thing in \AtEndOfPackage (at 1.3u whole code directly moved at end of package).

And 1.3p adds here \MTcustomizenewmcodes which had been regrettably forgotten by 1.3n.

1.3t adds some extras to handle correctly the minus sign and dotless i and j in subdued mode, even in case of usage with fontspec.

1.3u similarly lets math accents be correctly subdued.

1.3v adapts to \hbar and math accents now being robust with \TeX\ 2019-10-01 or later.

1.3w pays attention to the fact that \hbar may well be a \mathchar and not a robust macro!
And no need to worry about \hbar<space> finally in revised code.

\MTcustomizenewmcodes
\let\mst@original@hbar\hbar
\let\mst@original@imath\imath
\let\mst@original@jmath\jmath
\@tfor\@tempa:={grave}{acute}{check}{breve}{bar}
{dot}{ddot}{mathring}{hat}{tilde}{
\do
\expandafter\let\csname mst@original@\@tempa\expandafter\endcsname
\csname \@tempa\endcsname
\expandafter\let\csname mst@original@\@tempa\space\endcsname
\csname \@tempa\space\endcsname
\)\}
\ifmst@XeOrLua
\edef\mst@subduedminus{
\ifmst@subdued
\edef\mst@subduedminus{
\mst@Umathcodenum\noexpand=\the\mst@Umathcodenum\relax}
\else
\edef\mst@subduedminus{\noexpand=\the\mst@Umathcodenum\relax}
\fi
\fi
\MTeverymathoff
\MTresetnewmcodes
\MTnoicinmath
\MTmathoperatorsdonotobeymathxx
\%\mst@subduedhbar
\let\inodot\imath
\let\jnodot\jmath
\%\mst@subduedmathaccents
\mst@subduedminus
\else
\mst@nonsubduedhbar
\mst@nonsubduedmathaccents to get executed later (see code comments for \mst@dothe@mathaccentsstuff).
\% \mst@nonsubduedmathaccents \% will get executed later
\mst@nonsubduedminus
\mst@nonsubduedminus
1.3v needs this \mst@nonsubduedmathaccents to get executed later (see code comments for \mst@dothe@mathaccentsstuff).
\mst@nonsubduedmathaccents \mst@nonsubduedmathaccents % will get executed later
1.3j: an earlier version of this code was earlier part of \Mathastext@. As we are now in \AtBeginDocument we try to be careful not to overwrite \MTicinmath, \MTnoicinmath, \MTi-
calcoinmathxx, ... if issued by the user in the preamble, though. And we do not execute \MTmathactiveletters, it is issued by the package at loading time in order to allow user to
cancel it if desired from inside the preamble.
\ifx\mst@itcorr\@undefined
\def\mst@itcorr{\ifnum\fam=\m@ne/\fi}\
\@for\mst@tmp:=it,sl\do
{\ifx\mst@tmp\mst@ltshape@normal\let\mst@itcorr\@empty\fi }\%
\fi
\ifx\mst@ITcorr\undefined
\let\mst@ITcorr\mst@itcorr
\ifmst@frenchmath
\def\mst@ITcorr{\ifnum\fam=\m@ne/\fi}\
\@for\mst@tmp:=it,sl\do
{\ifx\mst@tmp\mst@shape@normal\let\mst@ITcorr\@empty\fi }\%
\fi
\fi
\fi
\mathcode thing has to be used with care under Unicode en-
gines. Unfortunately the \luatexUmathcode macro is helpless as it is not possible to know if
it will return a legacy mathcode or a Unicode mathcode. On the other hand the much saner
\XeTeXmathcodenum always return a Unicode mathcode.

1.15: The subdued code was initiated in May 2011. I returned to mathastext on Sep 24, 2012,
and decided to complete what I had started then, but in the mean time I had forgotten almost
all of the little I knew about L\TeX macro programming.
The point was to extract the data about how are 'letters' and 'operators' in the normal and
bold versions, through obtaining the math families of 'a' and '1', respectively\footnote{but the euler package for example assigns the digits to the letters symbol font...}. Due to the
reassignements done for characters by mathastext I also had decided in 2011 that the OT1
encoding, if detected, should be replaced by T1
\footnote{but the euler package for example assigns the digits to the \texttt{letters} symbol font...}
1.15d: Oct 13, 2012. The \mathcode thing has to be used with care under Unicode en-
gines. Unfortunately the \luatexUmathcode macro is helpless as it is not possible to know if
it will return a legacy mathcode or a Unicode mathcode. On the other hand the much saner
\XeTeXmathcodenum always return a Unicode mathcode.
UPDATE for mathastext 1.3 (2013/09/02): since the release of lualatex as included in TL2013, \texttt{\luatexmathcodenum} behaves as \texttt{\XeTeXmathcodenum} so mathastext 1.3 treats identically under both unicode engines the equal and minus signs (and the vertical bar).

1.15e: Oct 22, 2012. I add the necessary things to also subdue the \texttt{\mathbf}, \texttt{\mathit}, \texttt{\mathsf} and \texttt{\mathtt} macros (previous version only took care of the symbol alphabets \texttt{\mathnormal} and \texttt{\mathrm}) [update: 1.15f does that in a completely different and much simpler way] Notice that the package defines a \texttt{\mathnormalbold} macro, but it will not be subdued in the normal and bold math versions.

1.15f: Oct 23, 2012. The previous version of the code queried the math family of a, respectively 1, to guess and then extract the fonts to be reassigned to \texttt{mtletterfont} and \texttt{mtoperatorfont} (which is done at the end of this .sty file). The present code simply directly uses letters and operators (so mathastext could not subdue itself... if it was somehow cloned), but obtains indeed the corresponding font specifications in normal and bold in a cleaner manner. But it is so much shorter (and avoids the \texttt{\luatex} problem with \texttt{\luatexmathcode}). Anyhow, for example the euler package puts the digits in the letters math family! so the previous method was also error prone. In fact there is no way to do this subdued mechanism on the basis of the legacy code of mathastext. The only way is to rewrite entirely the package to query all mathcodes of things it changes in order to be able to revert these changes (and one would have to do even more hacking for \texttt{\mathversion{normal}} and not only \texttt{\MTversion{normal}} to work).

1.15f: and also I take this opportunity to do the subdued math alphabets things in a much much easier way, see below.

1.3s 2018/08/21: I have half-forgotten the reasons for modifying the font encoding to current \texttt{\encodingdefault}, but at any rate this should not be done in a \texttt{fontspec} context, encoding default being (now) TU it is very unlikely modifying from TU or to TU from something else will do any good. I add workaround here for case of \texttt{fontspec} being detected via the \texttt{\encodingdefault} setting.

1.3t 2018/08/22: the 1.3s fix erroneously removed the OT1->T1 replacement in TU context.

1.3u: the whole thing will only get executed At Begin Document.

I realize extremely late (2023/12/28) I never said explicitly anywhere it seems in the code comments that the \texttt{frenchmath} option effect is not subdued: the uppercase Latin letters \texttt{\mathcode}'s are not changed back to their defaults at start of a subdued document or when going to the subdued normal math version! Time to do so before the package enters resolutely dormant maintenance status soon... and I end up really forgetting anything and having wrong expectations on what is the behavior of the package.

2625 \ifmst@subdued
2626 \AtBeginDocument{%
2627 \def\mst@reserved\#1\getanddefine@fonts\symletters\#2\#3\@nil{%
2628 \def\mst@normalmv@letter{\#2}}%
2629 \expandafter\mst@reserved\mv@normal\@nil
2630 \def\mst@reserved\#1\getanddefine@fonts\symletters\#2\#3\@nil{%
2631 \def\mst@normalmv@letter{\#2}}%
2632 \expandafter\mst@reserved\mv@bold\@nil
2633 \def\mst@reserved\#1\getanddefine@fonts\symoperators\#2\#3\@nil{%
2634 \def\mst@normalmv@operator{\#2}}%
2635 \expandafter\mst@reserved\mv@normal\@nil
2636 \def\mst@reserved\#1\getanddefine@fonts\symoperators\#2\#3\@nil{%
2637 \def\mst@boldmv@operator{\#2}}%
2638 \expandafter\mst@reserved\mv@bold\@nil
2639 \edef\mst@tmpl@enc{\mst@encoding@normal}%
"Only preamble" restrictions. I was way too much obedient back in 2011, particularly taking into account how much of a pain it has been and still is that things such as`\DeclareMathSymbol` or`\DeclareMathAccent` are preamble-only. But keeping this for time being, however not using `\@onlypreamble` which breaks one’s heart when tracing to see how much place it takes, so we do it in one go.
\do\MTgreekfont
\do\Mathastextgreekfont
\do\MTgreekupdefault
\do\MTgreekitdefault
\do\MTDeclareVersion
\do\MathastextDeclareVersion
\do\MTWillUse
\do\MathastextWillUse
\do\Mathastextwilluse
\do\Mathastext
\do\mathastext
\immediate\write\m@ne{}
\PackageInfo{mathastext}{Loading is complete. You can now use \string\Mathastext\space to modify the normal and bold math versions. Use it with optional argument or use \string\MTDeclareVersion\space to declare additional math versions}@gobble}
\endinput