The mathastext package

Jean-François Burnol
jfbu (at) free (dot) fr

Package version: v1.2f (2013/01/21)
Documentation generated from the source file
with timestamp "23-01-2013 at 11:27:46 CET"

The 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, mathastext provides a simple mechanism in order to use more than one math-as-text font in the same document.

Contents

1 Installation instructions and recent changes

1.1 Installation

If you did not download this file from the internet chances are that `mathastext` is already in your TEX installation and that you are reading this documentation after having done something like `texdoc mathastext`. Still, read the following which explains how to produce some test files illustrating package features.

Download from the internet (from CTAN or from the package home page) `mathastext.dtx` (or copy the file already installed on your system, assuming it is up-to-date), put it in a temporary repertory and do `latex` (or `pdflatex`) on it. The compilation will not succeed if your installation is missing package `etoc` but even then it will have generated the package file `mathastext.sty`. You may then (if installing manually the package) move this up-to-date `mathastext.sty` to the appropriate location within your TEX installation (*i.e* to a repertory with name `mathastext` inside a repertory `latex` inside a repertory `tex` inside ...).

This `latex` run generated also a number of test files. They are provided to illustrate some features of the package. You can then run `latex`, `pdflatex`, `lualatex` or `xelatex` on them. Some test files are to be used only with XƎTEX and LuaTEX, as they use Unicode fonts.

1.2 Recent changes

The current version is `1.2f`: addition of the "change log" to this user manual, and some minor code improvements not changing neither features nor user interface.

Version `1.2e` made additions: in this user manual to the section 2.10 describing the compatibility issues, and to the test files illustrating various package features. Important changes to the source code were done to fix compatibility problems with active characters dating back to the `1.2` version. Also, an oversight in the implementation of the italic correction features from `1.2b` was corrected.

Version `1.2d` introduced the `asterisk` option (strangely the command `\MTlowerast` was not the one described in the manual; this has been corrected) and solved a problem of compatibility with `amsmath`.

The main new features in versions `1.2` and `1.2b` were the extended scope of the math alphabets and, respectively, added italic corrections in math mode. Both use mathematically active characters and some (thorny) technical problems related to globally active characters were finally solved to (almost) complete satisfaction (let's hope) only in the current `1.2e` version. These issues are commented upon in the compatibility section, in the test file `testmathastextalphabets.tex` and in the commented source code.

Earlier, version `1.15` introduced the concept of the subdued math versions.

2 What mathastext does

2.1 Examples

mathastext's basic aim is 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; mathastext may then help.

Here is an example with Latin Modern typewriter proportional:

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

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

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

$$\begin{aligned} \frac{d^2q}{db^2} = \frac{1}{2}\left\{\frac{1}{q}+\frac{1}{q-1}+\frac{1}{q-b}\right\}\left(\frac{dq}{db}\right)^2 - \left\{\frac{1}{b}+\frac{1}{b-1}+\frac{1}{q-b}\right\}\frac{dq}{db}\\ +\frac{q(q-1)(q-b)}{b^2(b-1)^2}\left\{\alpha+\frac{\beta b}{q^2}+\frac{\gamma(b-1)}{(q-1)^2}+\frac{\delta b(b-1)}{(q-b)^2}\right\} \end{aligned}$$

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

Notice that the Latin (and Greek letters) are in upright shape. But perhaps we insist on obeying the standardized habits:

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

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

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

$$\begin{aligned} \frac{d^2q}{db^2} = \frac{1}{2}\left\{\frac{1}{q}+\frac{1}{q-1}+\frac{1}{q-b}\right\}\left(\frac{dq}{db}\right)^2 - \left\{\frac{1}{b}+\frac{1}{b-1}+\frac{1}{q-b}\right\}\frac{dq}{db}\\ +\frac{q(q-1)(q-b)}{b^2(b-1)^2}\left\{\alpha+\frac{\beta b}{q^2}+\frac{\gamma(b-1)}{(q-1)^2}+\frac{\delta b(b-1)}{(q-b)^2}\right\} \end{aligned}$$

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

This was typeset using the Times font (available in any TEX distribution). Let us now 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.

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

$$a\frac{d}{da}X = \nu X - (1-X^2)\frac{2na}{1-a^2}\frac{aX+Y}{1+aXY}$$
$$a\frac{d}{da}Y = -(\nu+1)Y + (1-Y^2)\frac{2na}{1-a^2}\frac{X+aY}{1+aXY}$$

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

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

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

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 ...):

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

$$a\frac{d}{da}X = \nu X - (1-X^2)\frac{2na}{1-a^2}\frac{aX+Y}{1+aXY}$$
$$a\frac{d}{da}Y = -(\nu+1)Y + (1-Y^2)\frac{2na}{1-a^2}\frac{X+aY}{1+aXY}$$

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

Notice that the Greek letters also changed according to the *math version*: `mathastext` has indeed some (limited) capabilities to this effect, with its `LGRgreek` option. This document uses the `LGR` encoded fonts `cmtt`, `cmss`, and `txr`, which are part of standard TEX distributions.[1]

2.2 Overview

2.2.1 Basic use

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

[1] The first two are available (with no need to load explicitely any package in the document) via the combination `cbfonts` (`cbgreek-complete`) & `babel`, and the `LGR` encoded `txr` font (again no package loading is necessary) is part of the files of the `txfonstb` package.

As soon as one tries out other fonts for text than the Computer Modern ones one realizes how extremely "thin" are the default TEX fonts for mathematics: they definitely do not fit well visually with the majority of text fonts. With `mathastext` one can get one's (simple... or not) mathematics typeset in a manner more compatible with the text, without having to look for an especially designed font.

Here is a minimal example of what may go into the preamble:

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

All letters, digits, and punctuation signs inside math mode will then be typeset in Times.[2] The exact list of characters concerned by `mathastext` is a subset of the basic `ASCII` set:

$$\mathbf{abcdefghijklmnopqrstuvwxyz}$$
$$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$
$$\mathbf{0123456789}$$
$$\mathbf{!\,?\,*\,,\,.\,:\,;\,+\,-\,=\,(\,)\,[\,]\,/\,\#\,\$\,\%\,\&\,<\,>\,|\,\{\,\}}\text{ and }\backslash$$

As one can see, this is a very limited list! some possibilities are offered by `mathastext` for Greek letters and will be described later.

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

- the derivative sign $'$ is left to its default as the text font glyph ' is not, as a rule, a satisfying alternative.[3]
- for the minus sign `mathastext` uses the endash character –, if available, and not the hyphen character -,
- the option `asterisk` is necessary for `mathastext` to replace the binary math operator $*$ (and the equivalent control sequence `\ast`) with a version which uses the text asterisk * suitably lowered[4] (and with the correct spaces around it as binary operator). The reason for making it optional is that after this `$R^*$` or `$R^\ast$` do not work and have to 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.

[2] let's do as if we did not know the excellent `txfonts` package which employs Times for text and has a very complete math support, including many additional mathematical glyphs in comparison to the CM fonts.

[3] `v1.2` adds a customizable tiny space before $'$ to separate it from the previous letter, this is really needed when using upright letters in math mode with the CM derivative glyph.Compare f' with f'.

[4] the amount of lowering can be customized.

2.2.2 sans 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.

```
\renewcommand\familydefault\sfdefault
\usepackage{mathastext}
\renewcommand\familydefault\rmdefault
\begin{document}
```

2.2.3 always load mathastext last

As said already none of the "large" math symbols is modified in any way by mathastext. Only loading some math font packages such as fourier, kpfonts, mathabx, mathdesign, txfonts, newtxmath, etc... will change them. Think of loading these packages before mathastext, else they might undo what mathastext did. The more common symbols can be taken from the Symbol font (option symbolmisc, or symbolmax to get also the Greek letters from Symbol).

2.2.4 option LGRgreek

There is the issue of Greek letters. Sometimes the text font has Greek glyphs, in LGR encoding (this will be mentioned in the documentation of the font package). Then option LGRgreek tells mathastext to pick up these Greek letters. And it is possible to specify whether the Greek letters should be upright, or "italic".[5] Of course it is always possible to leave the responsability to set up Greek letters in math mode to packages loaded previously to mathastext.

2.2.5 avoid OT1 encoding

We specified in our minimal working example a T1 encoding (LY1 would have been ok, too) because the default OT1 does not have the $< > | \{ \}$ and $\backslash$ glyphs. If mathastext detects OT1 as the default encoding it will leave these characters to their defaults from the math fonts.[6]

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

[5] a more detailed discussion comes next. Note that the default CM and its replacement Latin Modern for european languages are (transparently to the user) extended with LGR encoded fonts from the cbfonts (cbgreek-complete) package.

[6] the subdued option, described next, acts a bit otherwise, it forces, contrarily to its usual low-key character, the replacement of OT1 by T1 for the fonts ultimately used with letters and digits in math mode.

2.3 Main options

2.3.1 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 `mathastext`.

Nevertheless, soon after I posted the initial version of the package to CTAN, I was overwhelmed by numerous[7] questions[8] 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 `italic` to have the Latin letters in math mode in italic shape.[9] There is also an option `frenchmath` to make the uppercase letters nevertheless upright, because this is the way of traditional French mathematical typography.[10]

2.3.2 The `defaultmathsizes` option

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

2.3.3 The `subdued` option

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

The previous description is in fact a bit optimistic: `mathastext` was not written initially in order to allow its action to be completely cancelled, and achieving this would require a complete rewrite of large portions of the code. In order to really have the displayed math (almost) as if `mathastext` had not been loaded, one must at a minimum also use the option `defaultmathsizes`. This does not quite suffice, because, 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, and although it will make a reasonable

[7] this means "more then one."

[8] I thank in particular Tariq Perwez and Kevin Klement for their kind remarks (chronological order).

[9] more precisely stated, the value of `\itdefault` is used.

[10] more precisely stated, the value of `\shapedefault` is used.

choice of this font, this is not a return to the previously prevailing situation. And then arbitrary packages could have done arbitrary things... so to be on the safe side one needs the `basic` option which limits the mathastextification to letters and digits (and should also be accompanied by `defaultimath` which prevents redefinition of the `\imath` macro, and `nohbar` which prevents redefinition of the `\hbar` macro...). And even then, in some circumstances, this will still not suffice; for example the *euler* package puts the digits in the same font as the Latin letters in math mode, but the subdued `mathastext` will pick them up in the same font as used in operator names, and in the case of the *euler* package, this is the main document font. So, even subdued, `mathastext` still kicks. But, as I think is illustrated by the examples given at the start of this document, the `subdued` option has its utility, and works reasonably well.

Furthermore, the `subdued` action does *not* extinguish package options `eulergreek`, `symbolgreek` or `symbol`. But `LGRgreek` is receptive to it.

2.4 Math versions

LaTeX has the concept of *math versions*, but most font packages do not define any such version beyond the default normal and bold (that they possibly customize to use such or such math font). The package `unicode-math` for unicode engines fruitfully uses this concept. `mathastext` uses 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. However the other math symbols (sums, products, integrals, logical signs, etc...) will be the same throughout the document as it is not in `mathastext` power to modify them. There are some possibilities to use different sets of fonts for the Greek letters, though.

The present document illustrated the use of various fonts, here is its preamble (slightly stripped-down):

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[subdued,italic,defaultmathsizes]{mathastext}
\MTDeclareVersion[n]{lmvtt}{T1}{lmvtt}{m}{n}
\usepackage{newcent}
\Mathastext[newcent]
\usepackage{times}
\Mathastext[times]
\usepackage[scaled]{helvet}
\renewcommand\familydefault\sfdefault
\Mathastext[helvet]
\begin{document}\MTversion{normal}
```

Let us examine this code: it uses once the command `\MTDeclareVersion` and three times the command `\Mathastext`, thus defining four `math versions`[11]: `lmvtt`,

[11] `math versions` are discussed in the document `fntguide.pdf` from your TeX distribution.

`newcent`, `times`, and `helvet`. The names can be taken arbitrarily (they only need to be suitable arguments to the LaTeX `\DeclareMathVersion` command which is invoked internally, so no spaces in the names). Two additional math versions pre-exist: the `normal` and `bold`, which, because there was the `subdued` option, were left untouched by `mathastext`.

Once these `math versions` are defined, `\Mathastextversion{name_of_version}`, or equivalently `\MTversion{name_of_version}` in the body of the document enacts the font switches. As is usual with LaTeX one can limit the scope to the inside of a group, or also switch back to the main set-up through issuing `\Mathastextversion{normal}`.

When `\Mathastext` is used in the preamble, it records the current font defaults and (except for the `normal` and `bold` versions under the `subdued` regime) sets up the math font to be used in that version to be the text font as found in `\familydefault`. But it is still possible for a `mathastext`-declared math version to have distinct fonts for text and math:

1. in the body of the TeX source, an optional argument (the name of a `mathastext`-declared math version) to `\MTversion` is allowed, and for example we used in the source of this document `\MTversion[newcent]{helvet}` meaning "New Century Schoolbook for the text and Helvetica for the math."

2. there are preamble-only commands `\MTencoding`, `\MTfamily`, `\MTseries`, `\MTshape`, `\MTlettershape` which tell `mathastext` what to do (for math *only*) in each math version declared *afterwards*, independently of the text fonts.

In the body of the document the LaTeX command `\mathversion{`⟨*version_name*⟩`}` will change only the fonts used in math mode. It is recommended to use instead the package command `\mathastextversion` (or its synonyms `\MTversion`, `\Mathastextversion`, `\MTVersion`), with a mandatory argument `{`⟨*version_name*⟩`}`. It

- checks in case the `subdued` option was specified whether the asked-for math version is ⟨*normal*⟩ or ⟨*bold*⟩, and adapts the following to that case,

- sets the font which will be used in math mode for letters (including math operator names), digits, punctuations and other ascii symbols,

- sets the font of the document text (if another version name is additionally passed as optional argument, it uses instead the corresponding font for text),

- resets the `\(family,rm,sf,tt)defaults` to their values as registered at the time of definition of the version,

- (see section 2.7) resets the user-defined extra spaces after the symbols ∃, ∀ and before the derivative ′ to the values as decided in the preamble on a *per version* basis,

- (see section 2.6) checks if the used font is upright and then activates the automatic insertion of italic corrections after each letter in math mode,
- (see section 2.5) makes the math operator names as well as the 'easy' non letter characters (and the asterisk) obey the math alphabets,
- does some additional set up for Greek letters when the LGRgreek option was used.

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

It is sometimes not compatible with 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 epigrafica package loads pxfonts. Hence it will interfere with mathastext if it is loaded after it.[12] But one can use instead \renewcommand{\rmdefault}{epigrafica},[13] followed with \Mathastext, or also \MTfamily{epigrafica}\Mathastext which will only change the font in math.

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

2.5 Extended scope of the math alphabets commands

Ever since the initial version of the package, some characters usually unaffected by the math alphabet commands \mathbf, \mathtt, \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 $\mathbf{\#}$ (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 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 amsmath \DeclareMathOperator or equivalent) operator names obey the math alphabet commands,[14]

[12] may typically give a 'too many math alphabets' error message.

[13] sometimes one needs to look in the .sty file of the font package to figure out the font name (it is rarely as epigrafica, the same as the package name), and, if one does not know the arcanes 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.

[14] contrarily to the next feature, this one is not likely to create incompatibilities with other packages, so it is activated by default.

2. and, *optionally*, all non alphabetical characters[15] treated by mathastext, *i.e.*, if not disabled by options, ! ? , : ; + − = ( ) [ ] < > { }, the asterisk ∗, and . / | \ # $ % &[16] will also obey the math alphabet commands (when not used as delimiters). The important thing is that the spaces added by TEX before and after are not modified.

Let us compare, for example, the new behavior of \mathtt and \mathbf

`(sin(n!) < cos(m - p)?)` $\qquad$ $\mathbf{[\sin(x+y) = \cos(z-t)]}$

with the traditional default behavior:

$(\sin(\mathtt{n!}) < \cos(\mathtt{m}-\mathtt{p})?)$ $\qquad$ $[\sin(\mathbf{x}+\mathbf{y}) = \cos(\mathbf{z}-\mathbf{t})]$

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 explicitely 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 relevant commands are

\MTmathoperatorsdonotobeymathxx
\MTnonlettersdonotobeymathxx
\MTeasynonlettersdonotobeymathxx

for deactivation and

\MTmathoperatorsobeymathxx
\MTnonlettersobeymathxx
\MTeasynonlettersobeymathxx

for activation.[17] [18]

[15] of course some of them are input preceded by a backslash, and the backslash itself is input as \backslash.

[16] # $ % & obey the math alphabets since the initial version of mathastext; the dot ., the slash /, the vertical bar | and the backslash \ do not have specific spacings inserted by TEX around them, and the procedure is then not a devilish one, this is why it is made the default for these characters which are listed apart. The math symbols \mid (which is | with type \mathrel) and \setminus (\ with type \mathbin) are counted among the 'difficult' cases, not the 'easy non-letters'.

[17] 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.

[18] when in subdued mode, the math alphabets are the default ones, not the ones modified by mathastext to use the document text fonts. As a result, matters of font encodings may then give unexpected results, for example for −. On the present document page we switched to a math version to escape from the subdued mode and avoid the problem with \mathbf{-} giving in the normal (subdued) math version ˘, when 'non-letters' are declared to obey math alphabets.

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 `\usepackage{mathastext}\MTnonlettersobeymathxx` 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 TEX if some precautions are not taken: `$x^?$`, `$R^+$` or `$\mathopen<A\mathclose>$` *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 **mathastext** does a 'mathematical activation' of the characters which could be unexpected and unchecked for by other packages. This is precisely the case with the `amsmath` package, and the problem goes away by just making sure that `amsmath` is loaded before **mathastext** (generally speaking, **mathastext** should be loaded last after all packages dealing with math things).

The brace control sequences `\{` and `\}` have their own (supplementary) switch, which is (even) less activated by default. The reason is that after `\MTexplicitbracesobeymathxx`, they regrettably can not be used anymore as delimiters: `\big\{` or `\big\}` cease to function and must be replaced by `\big\lbrace` and `\big\rbrace`. But we can now enjoy `{a, a > b}`, $\mathbf{\{a, a > b\}}$, $\{a, a > b\}$, or even $\boldsymbol{\{a, a > b\}}$.[19]

There is no such restriction with (,) or the brackets [,]. When used as delimiters though, they will become again unreactive to the math alphabets: the same applies to < and > or to any of the other characters such as the slashes /, \, when they are used as delimiters. This is a rather obvious restriction except possibly for the smallest size delimiters (the so-called 'small variants'), as all other sizes will be anyway absent from the text fonts. Even the small variants may not be really available in the text fonts: for example the standard < used as a delimiter is in LATEX a `\langle`: ⟨. With standard LATEX `$\left< x\right>$` gives $\langle x\rangle$.

Actually, **mathastext** does try to pick up most of the 'small variants' in the text font: `$\left<x\right>$` gives $<x>$ (but `$\left<b\right>$` gives $\langle b\rangle$.) If you don't like that use the option `nosmalldelims` (it is indeed perhaps a bit strange to have $<x>$ next to $\langle X\rangle$, again before blaming me, consider using `nosmalldelims`.) At any rate, whether 'small' or not, delimiters are not under the extended law of math alphabets, this is a general rule of the way **mathastext** manages these things.

Examples: `\mathbf{<a,b>}` gives $\mathbf{<a,b>}$. `\mathbf{\left<a,b\right>}` gives $\mathbf{\langle a,b\rangle}$. `\mathbf{\mathopen{<}a,b \mathclose{>}}` gives $\mathbf{<a,b>}$. The LATEX standard behavior for `\mathbf{\mathopen{<}a,b\mathclose{>}}` is $\mathbf{<a,b>}$.

[19] this last example uses the `\mathnormalbold` additional alphabet defined by **mathastext**.

2.6 Italic corrections

With the `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 *f* extends way out of its declared bounding box; let us compare the bounding boxes[20] for the letter *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 `$f(x)$` is wider with the standard math italic $f(x)$ (`21.31474pt`) than it is with the text italic font used in math:[21] $f(x)$ (`19.74986pt`) but we should be surprised that our text italic *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 *italic correction* for the letter *f*. The italic correction also exists and is used for the math italic font, it was inserted in `$f$` without us having to ask anything. Its value is `1.17865pt` for the math italic *f* and `1.8919pt` for the text italic *f*.[22] With the italic corrections included our bounding boxes are indeed more alike: f vs *f*.

Without the italic corrections[23] it is f vs *f*. I said that `$f$` 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:

abcdefghijklmnopqrstuvwxyz	`text italic in text`
abcdefghijklmnopqrstuvwxyz	`text italic in math`
$abcdefghijklmnopqrstuvwxyz$	`math italic in math`
$abcdefghijklmnopqrstuvwxyz$	`math italic in text`

Where are our italic corrections gone? the last line was done with `\usefont{OML}{lmm}{m}{it}` and confirms that italic corrections have been used for the math italic in math.

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 `\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:

$abcdefghijklmnopqrstuvwxyz$	`text italic in math`
$abcdefghijklmnopqrstuvwxyz$	`math italic in math`

[20] let's be honest, we are lying here about what exactly the first of these is bounding; this is explained later!

[21] we used simply `$\mathit{f(x)}$`.

[22] these values are for the Latin Modern fonts of course.

[23] here we give correctly the bounding box for the math italic f... without its italic correction!

We saw that the italic correction was taken into acount automatically (independently of the value of the interword space font parameter) in expressions such as `$f(x)$`. 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 f^j is very different from f^j... where the latter was coded with `$\hbox{\itshape f}^j$`. The inputs `$\mathit{\hbox{\itshape f\/}^j}$` and `$\mathit{f^j}$` give almost identical results: $\boxed{f^j}$ vs. $\boxed{f^j}$. 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 f was indeed used.

Subscripts are another matter: they do *not* take into account the italic correction. For example `$\mathit{f_i}$` gives the same horizontal positions as `$\mathit{\hbox{\itshape f}_i}$`: 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 the standard math italics f_i^j. In the next paragraphs we use f_i^i for more accurate comparison of the positioning of the sub- and superscript.

If we try something like this: `${f\/}_i^i$` we obtain f_i^i. Our overlapping game with `\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 be positioned exactly below the superscript![24] 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... **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[25] $\mathrm{f}_{\mathrm{i}}^{\mathrm{i}}$. We see the italic correction in action for the positioning of the superscript! Compare with `$\mathrm{\hbox{f}_i^i}$`: $\mathrm{f}_{\mathrm{i}}^{\mathrm{i}}$. Overlapping with `\rlap{$\mathrm{f_i^i}$}$\mathrm{\hbox{f}_i^i}$` gives $\mathrm{f}_{\mathrm{i}}^{\mathrm{i}}$ 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 `$\mathrm{{f\/}_i^i}$`: this gives $\mathrm{f}_{\mathrm{i}}^{\mathrm{i}}$ 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, `\rlap{$\mathrm{f_i^i}$}$\mathrm{{f\/}_i^i}$` gives $\mathrm{f}_{\mathrm{i}}^{\mathrm{i}}$.

[24] there are also some tiny vertical displacements of the sub- and superscripts.

[25] we just use `$\mathrm{f_i^i}$`.

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 `$\mathrm{ff}$`, `$\mathit{ff}$`, `$\mathbf{ff}$`, `$\mathsf{ff}$` all give ligatures ff, *ff*, **ff**, and ff and we would like to preserve this behavior).

> Starting with version `v1.2b`, `mathastext` adds the italic correction automatically after each letter of the Latin alphabet in math mode, *except* when these letters are italic or slanted.[26]
>
> These italic corrections are cancelled inside the arguments to the math alphabet commands others than `\mathnormal`, to allow the formation of ligatures as is expected in the standard default TEX font set-up in math.

The feature-implementing commands `\MTicinmath`, `\MTnoicinmath`, `\MTicalsoinmathxx` are described in section 3.3.2.

Note: *from brief testing on 2012/12/28, XeTeX seems not to create fake 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 XeTeX engine. On the other hand LuaLaTeX 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:*

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

Comment out the `fontspec` *line and use pdfLaTeX. All three outputs are different on my TEX installation. XeTeXdoes not have the italic corrections. LuaLaTeX does, but only for the italic font. pdfLaTeX has them for both the italic and the upright font.*

[26] the situation is rather ironical! by the way, the warnings in section 2.5 with `$x^?$` or similar are less of an issue here, because the letter is only *followed* by `\/` and anyhow the whole is put by `mathastext` within group braces, so no surprises with `$x^y$` or `$\mathbin x$`. Nevertheless it is still true that (in math mode only) the letter characters `a-z`, `A-Z`, expand to composite objects, something which could surprise other packages. The macro `\MTnoicinmath` cancels the mechanism.

2.7 Additional whitespaces

\MTforallskip, \MTexistsskip, and \MTprimeskip are three commands with each a mandatory argument like for example `3mu plus 1mu minus 1mu` or just `2.5mu`. 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{2mu}`,
- compare $\exists$N with $\exists$N, typeset after `\MTexistsskip{2mu}`,
- and finally compare f$'$ with f$'$, typeset after `\MTprimeskip{2mu}`.

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: `0.6667mu` for the skip after $\forall$, `1mu` for the skip after $\exists$, and `0.5mu` for the skip before the prime. The examples above become $\forall$B, $\exists$N and f$'$.[27]

With the `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 `\Mathastext[`⟨*version_name*⟩`]` (or `\MTDeclareVersion`) 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.

Also, when the `subdued` option has been used, the normal and bold math versions have by default zero length skips. Note though that there is no `\MTversion{normal}` done implicitely by the package when the document body starts[28] (*i.e.* at the level of `\begin{document}`), even when the `subdued` option is in force. As a result the last use in the preamble of the `\MT...skip` commands decides the skips which will be initially used; this is cancelled if `\begin{document}` is followed with `\MTversion{normal}`, in which case the skips used are the ones in force at the latest `\Mathastext` (without optional argument) preamble use, or just zero skips under the `subdued` regime.

2.8 Greek letters

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 `mathastext`:

[27] the derivative glyph from the `txfonts` math symbols adapts itself better to an upright letter, no skip seems to be needed then.

[28] when `\MTversion` is used, `mathastext` resets all font defaults, so for example a `\renewcommand\sfdefault` at the end of the preamble would be overruled if `\MTversion{normal}` was done automatically. Perhaps this behavior of `\MTversion` will change in a future version of the package.

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 `fourier` package with option 'upright', which gives upright Greek letters.

LGRgreek: this 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 txfontsb 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 `CM/LM` fonts (in serif, sans-serif and typewriter family) give the nice Greek letters in upright shape from the `cbfonts` package. To get these letters in your `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`.

eulergreek: the Greek letters will be taken from the Euler font (the document does not have to load the eulervm package, `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` which tells `mathastext` to pick up in each math version the letters from the `LGR` encoded font used in that version, and `selfGreeks` to tell `mathastext` to do as for `selfGreek` but separately in all math versions.

The `subdued` option interacts with the options for Greek letters in the following way:

1. in its presence, the LGRgreek and LGRgreeks options cease to have any effect in the normal and bold math versions,

2. selfGreek(s), eulergreek and symbolgreek act normally, they are not affected by the presence or absence of subdued.

2.8.1 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 LGRgreek and selfGreek.

The options frenchmath, itgreek, upgreek, itGreek and 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 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,upgreek: both lowercase and uppercase are in the `\itdefault`, respectively the `\updefault` shape (at the time of loading the package or at the time of a subsequent call to `\Mathastext` or `\MathastextWillUse`),

itGreek,upGreek: same as above, but only for the uppercase letters.

So, the default gives the classic TEX behavior when option italic was passed. Each call to `\Mathastext` (or `\MathastextWillUse`) macros (described in a later section) reinitializes the computation of the shapes.

As mentioned already the package allows to define various "math versions". In the case of eulergreek or symbolgreek they apply to all these versions. In the case of the options LGRgreeks or selfGreeks (notice the additional "s"), each math version is assumed to have its text font available in LGR (or OT1 encoding) and also the shapes will be local to the math version.

Finally version 1.15c of mathastext introduces new preamble-only commands to change the shapes, and even the font, used for Greek letters, in case of package options LGRgreek/selfGreek. They are \MTitgreek, \MTupgreek, \MTitGreek, \MTupGreek: these are used like the options and change only the shapes for the math versions which will be declared *next* in the preamble; and \MTgreekfont{name_of_font} will tell the *next* math versions to use that font family. To use this command you need to know the (little) name of a suitable font family available in LGR encoding: for example lmr, txr (needs txfontsb package on your system), DejaVuSerif-TLF (needs dejavu package on your system), etc...

2.9 Unicode engines

mathastext has been made minimally unicode-aware and can be used with XeTeX or LuaLaTeX, but the user is strongly advised to look first at the mathspec package, which is a far more powerful package designed for unicode (only for XeTeX), and to unicode-math (to use OpenType math fonts).

Of course mathastext is extremely far from being able to define a math font, as it applies basically only to a subset of the 32-127 ascii range, and in particular it does not know how to use a given Unicode font simultaneously for Latin and Greek letters. Again the user is strongly advised to look at mathspec and unicode-math.

When using mathastext with either XeTeX or LuaLaTeX it is recommended to use the fontspec package. Else, some of the encoding dependent things done by mathastext like using the en-dash character to get a minus sign in math mode will not be put in place. Furthermore, it is *necessary* to load fontspec with its no-math option, and this *must* happen before loading mathastext.

> Use fontspec with its *no-math* option, and load it *prior* to mathastext.

> The amsmath package, if used, *must* be loaded *prior* to mathastext.

A little piece of the functionality of mathastext is less fully realized under the LuaLaTeX engine than it is with XeTeX. This is a temporary situation as the needed feature of LuaLaTeX has been implemented in its latest development release. I will update mathastext when these binaries will have reached the distribution stage.

I already mentioned in the section 2.6 the fact that the italic corrections were not available for OpenType fonts under the XeTeX engine and only partially available for the LuaLaTeX 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 pdfetex engine (the OpenType fonts not being usable with the latter engine, this is not a criterion of choice anyhow).

To specify math versions using unicode fonts, use the fontspec \setmainfont command (with arbitrary optional features). This command can be issued before loading mathastext, or after and then will be followed by a \Mathastext command with the name of the version in square brackets.

It is possible to mix unicode fonts and classical TeX fonts. But this is definitely *not* recommended as mathastext decides once and for all what is the font slot of things such as the text endash (used for the minus sign) and this is encoding dependent. So it is best to have either only unicode fonts, or only old-fashioned TeX fonts in a fixed encoding (T1, or LY1 for example).

The package was not extensively tested with unicode engines. I include here two examples which compiled successfully with XeTeX and LuaLaTeX, the first one on

a Linux machine, the second one on a Mac OS X machine.[29]

```
\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]
\MTDeclareVersion{times}{T1}{ptm}{m}{n}
\setmainfont[Color=FF0000]{Andale Mono}  \Mathastext[Andale]
\begin{document}
\newcommand\TEST[1]{\MTversion{#1}%
\begin{multicols}{2}
\hbox to\columnwidth{\hbox to\columnwidth{\hfil
                $abcdefghijklmnopqrstuvwxyz$\hfil}\kern-2.5em{#1}}
   \centerline{ $ABCDEFGHIJKLMNOPQRSTUVWXYZ$ }
   \centerline{ $0123456789$ }
   \centerline{ $!\,?\,*\,,\,.\,:\,;\,+\,-\,=\,(\,)\,[\,]\,/\,\#\,%
   \$\,\%\,\&\,<\,>\,|\,\{\,\}\,\backslash$ }
\columnbreak
   \centerline{ abcdefghijklmnopqrstuvwxyz }
   \centerline{ ABCDEFGHIJKLMNOPQRSTUVWXYZ }
   \centerline{ 0123456789}
   \centerline{ !\,?\,*\,,\,.\,:\,;\,+\,-\,=\,(\,)\,[\,]\,/\,\#\,%
   \$\,\%\,\&\,<\,>\,|\,\{\,\}\,\char92 }
\end{multicols}}
\begin{multicols}{2}
   \centerline{\textbf{math mode}}
\columnbreak
   \centerline{ \textbf{text} }
\end{multicols}
\TEST{DejaVu}\TEST{Verdana}\TEST{times}\TEST{Andale}
\TEST{Arial}\TEST{bold}\TEST{normal}
\end{document}
```

And now the same thing with fonts available on Mac OS X:

```
\documentclass{article}
\usepackage[hscale=0.8]{geometry}
\usepackage{multicol}
\usepackage[no-math]{fontspec}
\usepackage{lmodern}
\usepackage[subdued,italic]{mathastext}
\setmainfont[Color=FF0000]{Hoefler Text}  \Mathastext[Hoefler]
```

[29] running `latex` (in a temporary repertory) on a copy of the file `mathastext.dtx` will extract extended versions of these examples as test files.

```
\setmainfont[Color=336633]{American Typewriter}\Mathastext[Typewriter]
\setmainfont[Color=0000FF]{Herculanum}    \Mathastext[Herculanum]
\setmainfont[Color=FF00FF]{Didot}         \Mathastext[Didot]
\setmainfont[Color=999999]{Comic Sans MS} \Mathastext[Comic]
\begin{document}
   --- copy here the code from the previous example ---
\TEST{Didot}\TEST{Comic}\TEST{normal}\TEST{Herculanum}
\TEST{Hoefler}\TEST{Typewriter}\TEST{bold}
\end{document}
```

The test files which will be produced by running `latex` on `mathastext.dtx` are more extent.

2.10 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.

In case of a 'too many math alphabets' message try the `defaultalphabets` option or one of its `defaultnormal`, `defaulttt`, etc. . . sub-options.

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[30] 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

[30] italic correction insertion for the latin letters, receptivity to the math alphabet action for the other characters.

inactive.[31]

Here is indeed some code that you should **ABSOLUTELY NOT** try at home:

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

DO NOT DO THIS AT HOME : it creates an infinite loop. This is due to the fact that the colon is simultaneously active (this is made by `babel` 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 `mathastext` will detect that Babel has activated the colon, and will cancel the mathematical activation (the precise definition done by `mathtools` was already lost at begin document due to overwriting by `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 `testmathastextalphabets.tex` for more explanations and illustrations, and the commented source code of the package). Pure mathematical activation revealed an incompatibility of another type with `amsmath`. To fix it, `mathastext` now replaces an inner macro of `amsmath` (`\resetMathstrut@`) with its own version.

Always load `amsmath` before `mathastext`.

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 `\DeclareMathOperator` macro from `amsmath`.

Important! As is mentioned in the section 2.5, with the "non letters obey math alphabets" mechanism, 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 `\MTnonlettersdonotobeymathxx` by default.

One thing to take note of is that this mechanism uses the `\everymath` and `\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.

[31] only the characters ; , : ! ? + − = < > ( ) [ ] * mentioned in section 2.5 as 'difficult non letters' (and the right quote ') and the latin letters are concerned here; it seems highly unprobable that a latin letter $\in \{a–z, A–Z\}$ will have been made globally active (only letters never being used in command names are possible candidates), but `mathastext` has been designed to cope with it, should it happen ...

Doing (after the `\begin{document}`) `\everymath={}` and `\everydisplay={}` (preemptively) annihilates all of the **mathastext** (evil) doing with math active characters but it annihilates also everything else some other package might have put in there, so it is much better, in case the need arises to preemptively cancel the action of **mathastext**, to use the package macros `\MTmathoperatorsdonotobeymathxx`, `\MTnonlettersdonotobeymathxx` (this is default), `\MTnoicinmath`, and `\MTnormalprime` (which cancels[32] the redefinition done by **mathastext** of the action of the right quote `'` in math mode).

An exception is with the modifications to ∃ and ∀ which are done by the package in the preamble; the old definitions can be saved before to be reestablished after loading the package. Just using `\MTexistsskip{0mu}` and `\MTforallskip{0mu}` should be sufficient, except if it is important that the `\exists` and `\forall` control sequences always expand to only one token each.

3 Package options and commands

3.1 Summary of main options

italic, frenchmath: italic letters in math, upright uppercase if **frenchmath**.

subdued: acts in a subdued way. The LaTeX normal and bold math versions are left (quasi) unchanged. With version `1.15e` of the package this statement applies also to the math alphabets `\mathbf`, `\mathit`, `\mathsf`, and `\mathtt` (and not only to `\mathnormal` and `\mathrm` as in previous versions.)

LGRgreek, eulergreek, symbolgreek: the Greek letters will be taken, respectively from the text font itself (in `LGR` encoding), or from the Euler font, or from the Postscript Symbol font.

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, **mathastext** with option **symbolmax** may give in the end a PDF file quite smaller than the one one would get without the package.

defaultmathsizes: **mathastext** sets up bigger sizes for subscripts (it also copies code from the **moresize** package to redefine `\Huge` and define `\HUGE`). Use this option to prevent it from doing so.

defaultalphabets: by default, **mathastext** redeclares the math alphabets `\mathrm`, `\mathit`, `\mathtt` etc... (but not `\mathcal` of course) to refer to the current document text fonts (at the time of loading the package and in each

[32] no command is provided to revert it, but its scope is limited to the current environment or group.

mathastext math version). Use this option to prevent it from doing so (each alphabet also has its own disabling option).

3.2 Miscellaneous

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 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 fontspec has been loaded).

amsmath: the behavior of the \DeclareMathOperator command of amsmath is slightly modified by mathastext. This command initially allows crazy things like \DeclareMathOperator\crazy{m.ch-in'tr/u:c} and then the ., -, ', / and : will be typeset in the roman font. But the font number was hardcoded in the macro and furthermore the code of amsmath would cause an error with Unicode engine as soon as some Unicode code is assigned to the minus character.[33] This specific issue will perhaps be fixed by some hypothetic future release of amsmath, or by other packages providing patches, but I decided for a preemptive strike. As a result the declaration above will not cause an error when \crazy is used with a Unicode engine, but there are now some spacings around the punctuation characters. To avoid this use (also with LaTeX):

\DeclareMathOperator\crazy{m{.}ch{-}in{'}tr{/}u{:}c}

Note though that the quote ' will be typeset as a derivative sign ′.

hbar: the definition of \hbar inherited from default LaTeX will 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 the bar accross the h will come from the original default math font for letters (usually cmmi), and furthermore its placement on the h can be odd-looking. So we redefine \hbar to use only the text font (and this will be aware of the italic option). Our construction does not always give an optimal result (and its scope is limited to the OT1, LY1 and T1 encodings), so an option nohbar deactivates it. There is no \hslash provided by the package, though. The version 1.12 of the package when dealing with a Unicode font tries to get the \hbar directly as a glyph from the font.

dotless i and j: by default the package redefines \i and \j to give the dotless i and j (if it exists at all), *also in math mode,* in the text font. Will overwrite the default commands \imath and \jmath. In version 1.12 of the package this should work in all encodings, including Unicode (it is then assumed that

[33] To the experts: the sin, cos, ... operator names are *not* defined by amsmath with the help of the \DeclareMathOperator macro, hence are not the cause of an error in XeTeX/LuaLaTeX. What mathastext does is to let to relax the \newmcodes@ macro, so it is possible to save it before loading mathastext and re-establish later, if really really this is what you want.

fontspec has been loaded, and of course that the glyphs are indeed in the font).

asterisk: previous versions of mathastext 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[34] is decided by the mandatory argument to the command \MTlowerast{⟨*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 subscripts of subscripts. But * is now 'mathematically active'[35] and R^* or $R^\ast$ *must* be input as R^{*} and $R^{\ast}$. Furthermore, they will obey the math alphabet commands.

XeTeX and LuaLaTeX: for the en-dash and the dotless i and j, the package expects to detect either the EU1 encoding for XeTeX or the EU2 encoding for LuaLaTeX (this will be true if fontspec was loaded), or one of OT1, LY1 or T1, else it will renounce and not attempt to access the en-dash or the dotless i and j glyphs. With LaTeX and PdfLaTeX, there is no such limitation and all 8bit-encodings (containing these glyphs) should be ok.

fontspec: one more note to users of XeTeX/LuaLaTeX with fontspec: it has to be loaded with the option no-math, and before mathastext.

vec accent: The default \vec accent is not appropriate for upright letters, so mathastext provides a \fouriervec which takes its glyph in a Fourier font, and an Ersatz \pmvec which is reasonably good looking on upright letters and works with the \rightarrow glyph. Contrarily to version 1.0, the default \vec is not overwritten with \fouriervec. And contrarily to version 1.1, one now needs to pass the option fouriervec to have the math accent \fouriervec defined by the package.

math alphabets:

- We define a new math alphabet command \mathnormalbold which gives direct access to the bold version of the \mathnormal alphabet (rather than using either the \bm command from the bm package or the

[34] with the option symbolmisc, 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.

[35] in a hopefully safe way, for example $\label{eq*1}$ is ok.

\boldsymbol command from the 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.

- The other math alphabet changing commands defined by the package are \MathEulerBold, \MathEuler and \MathPSymbol.
- \mathnormal, \mathrm, \mathbf, \mathit, \mathsf and \mathtt are modified to make reference to the document text fonts (this can be disabled by suitable package options).
- version 1.2 of 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.

math accents: an option mathaccents is provided to pick up the accents in math mode from the text font, but the package knows only T1, LY1 or OT1-compatible encodings.

Regarding the encoding-dependent glyphs: the en-dash, the dotless i and j, the math accents, the hbar, are encoding dependent and the relevant decisions are made once by mathastext at the time it is loaded and are applied to all declared math versions. So you can use math versions with different encodings but, regarding these characters only those with the same encoding as the normal math version will display them correctly.

> It is thus recommended that all declared mathastext math versions use the same font encoding.

3.3 Commands

3.3.1 Preamble-only commands

These commands mainly facilitate the definition of math versions, in a mathastext extended sense. It is thus not necessary to use them to activate the package basic functionalities, loading mathastext is enough.

- \Mathastext (or \mathastext): reinitializes mathastext according to the current defaults of encoding, family, series and shape.[36]
  - It can also be preceded optionally by one or more of[37] \MTencoding{⟨*enc*⟩}, \MTfamily{⟨*fam*⟩}, \MTseries{⟨*ser*⟩}, \MTshape{⟨*sh*⟩}, and, new with version 1.1, \MTlettershape{⟨*sh*⟩}. For example valid values

[36] updates also the font and shapes for the Greek letters (LGRgreek option), and the skips to be inserted after the symbols ∀ and ∃, see *infra*.

[37] these commands exist also with long names: \Mathastextencoding, etc. . . The same applies to the other commands mentioned in this section.

are, respectively, ⟨*T1*⟩, ⟨*phv*⟩, ⟨*m*⟩, ⟨*n*⟩, and ⟨*it*⟩: 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 `\Mathastext`, and can only be undone by using them again.

– **math versions:** starting with version `1.12` `\Mathastext` accepts an optional argument `[`⟨*version_name*⟩`]`, which will serve as a name to designate the corresponding math version (without optional argument `\Mathastext` redefines the default normal and bold versions.) This argument, being optional, must be enclosed within square brackets.[38]

- `\MTWillUse[`⟨*ltsh*⟩`]{`⟨*enc*⟩`}{`⟨*fam*⟩`}{`⟨*ser*⟩`}{`⟨*sh*⟩`}`: tells `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 ⟨*ltsh*⟩ specifies a shape for the letters, for example `\itdefault`, or directly ⟨*it*⟩ or ⟨*sc*⟩.

- `\MTDeclareVersion[`⟨*ltsh*⟩`]{`⟨*name*⟩`}{`⟨*enc*⟩`}{`⟨*fam*⟩`}{`⟨*ser*⟩`}{`⟨*sh*⟩`}`: declares that the document will have access to the font with the specified characteristics, under the math version name ⟨*name*⟩. For example:

  ```
  \MTDeclareVersion[sc]{palatino}{T1}{ppl}{b}{sl}
  ```

  declares 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).[39] When the optional argument is absent, and `mathastext` was loaded with the `italic` option, then the default letter shape will be `it`,[40] else letters will have the same shape as used for digits and operator-names.

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

- `\MTEulerScale{`⟨*factor*⟩`}`: scales the Euler font by ⟨*factor*⟩.

- `\MTSymbolScale{`⟨*factor*⟩`}`: scales the Symbol font by ⟨*factor*⟩.

- `\MTitgreek`, `\MTupgreek`, `\MTitGreek`, `\MTupGreek`: optional commands, active only in the case of the `LGRgreek` option, to decide the shape of the Greek letters in the versions which will be declared next.

[38] The allowed version names are as for the LaTeX `\DeclareMathVersion` macro. Do not use "normal" or "bold"; this is already taken care of by the initial loading of the package or a later command `\Mathastext` without any optional argument.

[39] I do not especially recommend to use this in real life!

[40] 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 `\itdefault` of the time of loading the package.

- \MTgreekfont{⟨*fontfamily*⟩}: optional command with a mandatory argument which specifies the font for Greek letters in all mathastext math versions declared afterwards via \Mathastext or \MTDeclareVersion. Only effective with LGRgreek option.

3.3.2 Commands usable everywhere

- \MTexistsskip{⟨*math glue*⟩}: specifies the amount of skip or more generally glue to put after each ∃ math symbol. Indeed, upright letters (or digits for that matter) often appear to be positioned a bit too close to the quantifier: ∃B. The package default is to add a 1mu skip (this default is set to zero in the case of italic): ∃B. One can change the default with the following syntax: \MTexistsskip{2mu plus 1mu minus 1mu}, which if used in the preamble and followed with a \Mathastext command (or \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 subdued, the skips are set to zero for the normal and bold math versions. In the case of the option italic, the default skip is set to zero.

- \MTforallskip{⟨*math glue*⟩}: the default is to add a .6667mu math skip after each ∀ (except with the option italic for which the default is set to zero). Compare ∀F (has the skip) with ∀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 skips are set to be zero by default 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 also at any time in the document.

- \MTprimeskip{⟨*math glue*⟩}: the default is to add a 0.5mu skip before the derivative glyph, except for the italic option, and except in the normal and bold math versions with option subdued. The same observations apply.[41]

- \MTlowerast{⟨*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\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, \MTnonlettersobeymathxx, \MTnonlettersdonotobeymathxx, \MTexplicitbracesobeymathxx, \MTexplicitbracesdonotobeymathxx, \MTeasynonlettersobeymathxx, \MTeasynonlettersdonotobeymathxx: these commands with-

[41] if \begin{document} is not followed with \MTversion{normal}, the latest values set in the preamble by these commands are used at the beginning of the document, even under option subdued.

out argument must be used *outside* of math mode. mathastext issues automatically \MTmathoperatorsobeymathxx, \MTeasynonlettersobeymathxx and and \MTnonlettersdonotobeymathxx at the time of \usepackage{mathastext}. See the discussion in section 2.5. The 'explicit braces' activation works only when \MTnonlettersobeymathxx has already been used.

- \MTicinmath: this command tells mathastext to add italic corrections after all letters in math mode. Its effect is cancelled inside math alphabets (except \mathnormal). The command is issued by default in the preamble when the shape of the letters used in math mode is neither italic nor slanted. It is reissued each time a math version is entered, under the same conditions (and except of course for the normal and bold subdued math versions). Its effect is local to the group or environment in which it has been issued. From inside math mode, it will only revert an earlier \MTnoicinmath from the same math group, and the math mode itself must have been entered with the italic corrections on.

- \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). Its effect remains local to the group or environment where it was issued.

- \MTICinmath, \MTnoICinmath: these commands set up the italic corrections only for the uppercase letters. Using \MTICinmath de-activates the added italic corrections for uppercase letters in the arguments of *all* the math alphabet commands.

- \MTicalsoinmathxx: this command de-activates the de-activation of the italic corrections inside the arguments to the math alphabet commands apart from \mathnormal. It can be used in as well as outside of math mode.[42] To cancel its effect either enclose it in a group or environment or re-issue \MTicinmath after it.

3.3.3 Body-only commands

- \MTversion[⟨*nametext*⟩]{⟨*namemath*⟩}:[43] in the absence of the optional argument changes simultaneously the text and the math fonts to be the fonts corresponding to the version ⟨*namemath*⟩. If there is an optional argument then the text fonts will use ⟨*nametext*⟩ and the math mode will use ⟨*namemath*⟩. To change only the math fonts, one may use the LaTeX command \mathversion. However this is not recommended as mathastext does quite a lot of additional configuration when invoked with \MTversion.

[42] it is provided for the mathastext defined math alphabet \mathnormalbold which contrarily to \mathnormal is not treated apart from the other math alphabets.

[43] \MTversion is also available as \MTVersion, \Mathastextversion and \mathastextversion.

Each invocation of \MTversion checks the chosen shape of letters in math mode and if it is not *italic* or *slanted* it activates the automatic insertion of italic corrections in math mode, for better positioning of subscripts. See the discussion in the section 2.6. With the frenchmath option the package checks separately whether to insert the italic corrections after lowercase and/or uppercase letters.

With the subdued option, \MTversion{normal} and \MTversion{bold} do \MTmathoperatorsdonotobeymathxx (and \MTnonlettersdonotobeymathxx).

All further commands are for math mode only.

- \hbar: this is constructed (in a way compatible with the italic option) from the h letter and the ¯ accent from the mathastext font. Note that \mathrm{\hbar} and \mathbf{\hbar} should work and that \hbar does scale in subscripts and exponents. Only for T1 and OT1 (or LY1) encodings.

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

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

- \mathnormal, \mathrm, \mathbf, \mathit, \mathsf, \mathtt: modifications of the original \mathnormal, \mathrm, \mathbf, \mathit, \mathsf, \mathtt to use the mathastextified font. The underlying internal LaTeX structures related to the original commands are not overwritten, so the original commands can be saved under other names before \usepackage{mathastext}, to be used in case of necessity (this is what option subdued does.)

- \mathnormalbold: a bold version of \mathnormal. Differs from \mathbf when the italic option has been used, or when use has been made of \MTlettershape to specify a shape for letters distinct from the one for digits and operator names, or similarly when the math version has been declared via \MTDeclareVersion with its optional parameter for shape of letters.

- \inodot, \jnodot: the corresponding glyphs in the chosen font for math mode. By default, will overwrite \imath and \jmath. With version 1.12 by default \i and \j work also in math mode and give then \inodot, resp. \jnodot. This should work for all 8bit-encodings having these glyphs, and also in Unicode.

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

- \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: \Digamma, \Alpha, \Beta, \Epsilon, \Zeta, \Eta, \Iota, \Kappa, \Mu, \Nu, \Omicron, \Rho, \Tau, \Chi (no \Digamma for Symbol). Also an \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\varepsilon=\epsilon, etc. . . , in case only the 'variant' form of the letter was used in the documents.

- Miscelleneous mathematical symbols from the postscript Symbol font are made available (or replaced) by option symbolmisc.[44] They are \prod ∏ \sum ∑ \implies ⇒ \impliedby ⇐ \iff ⟺ \shortiff ⇔ \to → \longto ⟶ \mapsto ↦ \longmapsto ⟼ \aleph ℵ \inftypsy ∞ \emptyset ∅ \surd √ \nabla ∇ \angle ∠ \forall ∀ \exists ∃ \neg ¬ \clubsuit ♣ \diamondsuit ♦ \heartsuit ♥ \spadesuit ♠ \smallint ∫ \wedge ∧ \vee ∨ \cap ∩ \cup ∪ \bullet • \div ÷ \otimes ⊗ \oplus ⊕ \pm ± \ast ∗ \times × \proptopsy ∝ \mid | \leq ≤ \geq ≥ \approx ≈ \supset ⊃ \subset ⊂ \supseteq ⊇ \subseteq ⊆ \in ∈ \sim ~ \cong ≅ \perp ⊥ \equiv ≡ \notin ∉ \langle ⟨ \rangle ⟩. And a \DotTriangle ∴ is made available by option symbolre (which overwrites \Re and \Im: ℜ, ℑ). The \inftypsy and \proptopsy 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 ∏∑ in inline math, and in display math the Computer Modern ones (or whatever is set up by other packages) :

$$\prod\sum$$

  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 \defaultprod and \defaultsum will refer to the \prod and \sum before redefinition by the package: this is to allow constructs such as $\displaystyle\defaultprod$ or \[\textstyle\defaultprod\], because they would not work with the \prod and \sum as re-defined by the package.

3.4 Complete list of options

- basic: only mathastextify letters and digits.

[44] option asterisk is also required to treat the ∗. Recall from section 2.5 that the asterisk in math mode (also when using the control sequence \ast) appears then to TeX to be a composite object.

- subdued: do not change the default fonts or the math alphabets in math mode for the normal and bold math versions, turn on the mathastext-ification only after an explicit \MTversion (or \mathastextversion) command activating an additional math version as declared in the preamble. With option subdued each \MTversion{normal} or \MTversion{bold} does \MTmathoperatorsdonotobeymathxx (and \MTnonlettersdonotobeymathxx).
- italic: the letters default to italic shape in math mode.
- frenchmath: italic lowercase Latin letters, but uppercase Latin letters in the same font as for digits and operator names. In general this means that they will be upright. In case of the LGRgreek option, frenchmath influences also the shape of the Greek letters.
- endash, emdash: use the text font en-dash (–) or even the em-dash (—, but this seems crazy) for the minus sign rather than -. endash option is default for the package.
- asterisk: use the text font (or the Symbol font) asterisk in math mode.
- noendash: the minus sign will be the - from the text font, not the en-dash –.
- nohbar: prevents mathastext from defining its own \hbar.
- nolessnomore: besides ! ? , . : ; + − = ( ) [ ] / # $ % & mathastext treats also < > | { } and \. Use this option to let it not do it. This is the default in case of OT1-encoding.
- further excluding options: noexclam ! ? nopunctuation , . : ; noplus, nominus, noplusnominus + − noequal = noparenthesis ( ) [ ] / nospecials # $ % & and nodigits.
- alldelims: true by default, means that the characters excluded by nolessnomore are treated. Use this option in case of a mono-width OT1-encoded font.
- nosmalldelims: this prevents 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 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.
- symbolgreek, symboldigits: to let Greek letters (digits) use the Symbol font.
- symbolre: replaces \Re and \Im by the Symbol glyphs $\Re, \Im$ and defines a \DotTriangle command ($\therefore$).

- 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 is for a font which is also available in LGR-encoding. It is possible to change the font per math version, via the use of the `\MTgreekfont` command in the preamble.
- LGRgreeks: each declared math version will be supposed to be with a font which is also available in LGR-encoding.
- selfGreek: this is for a font which is also available in `OT1`-encoding and contains the glyphs for the default eleven capital Greek letters.
- 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, upGreek, itGreek: options to tell to use `\itdefault` or `\updefault` for the lowercase and uppercase (or only the uppercase) Greek letters. Only operant in the case of the LGRgreek(s) and selfGreek(s) options.
- mathaccents: use the text font also for the math accents. As in vanilla LaTeX, they are taken from the font for the digits and `\log`-like names. Obey the alphabet changing commands. Will work only for `T1`, `LY1`, or `OT1`-compatible encodings.
- defaultbf, defaultit, defaultsf, defaulttt: do not set up, respectively, the `\mathbf`, `\mathit`, `\mathsf`, and `\mathtt` commands to use the mathastext-ified font. This also prevents mathastext to create internally `\Mathxx` alphabets (it never overwrites the original `\mathxx` things but let `\mathxx` point to `\Mathxx` instead), so one can use these options if one encounters a 'too many math alphabets' LaTeX error.
- defaulnormal, defaultrm: do not identify the default `\mathnormal` (resp. `\mathrm`) with the newly created `\Mathnormal` (resp. `\Mathrm`) commands which use the mathastextified fonts in each math version.

- `defaultalphabets`: all the `defaultxx` options together, and additionally tells `mathastext` not to create the `\mathnormalbold` alphabet either.
- `defaultimath`: do not overwrite `\imath` and `\jmath`, do not extend `\i` and `\j` to math mode use.
- `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/showcase.html`

4 Change log

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 only in math mode how an active character acts, without breaking things. The package now does that /only/ if the activation appears to originate in the Babel system, as it is then possible to modify appropriately the Babel macros \user@active<char> and \normal@char<char>. The relevant issues are discussed in section 2.10 of the user manual, in the test file testmathastextalphabets.tex, and in the source code comments to the 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.
* The technique of mathematical activation is maintained only for the characters which are not catcode active (at the entrance in math mode, as mathastext does all its activation job at everymath and everydisplay).
* 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.

* various improvements in the documentation.

* general clean up and better commenting of the source code.

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

* improvements to the pdf documentation

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 alpha-

bets \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.
* internal changes for better readability of the code.

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

```
\ProvidesPackage{mathastext}
 [2013/01/21 v1.2f Use the text font in math mode (jfB)]
\NeedsTeXFormat{LaTeX2e}
```

Testing for XeTeX and LuaLaTeX. I should load some package for this code which I copied somewhere. 2013/01/01: at least the code now does not overwrite previous `\ifXeTeX` or `\ifLuaTeX`...

```
\newif\ifmst@XeTeX
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname XeTeXinterchartoks\endcsname\relax
  \mst@XeTeXfalse
\else
  \mst@XeTeXtrue
\fi
\newif\ifmst@LuaTeX
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname directlua\endcsname\relax
  \mst@LuaTeXfalse
\else
  \mst@LuaTeXtrue
\fi
```

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

```
\def\mst@oti{OT1}\def\mst@ti{T1}\def\mst@lyi{LY1}
\def\mst@eui{EU1}\def\mst@euii{EU2}
\newif\ifmst@goahead
\newif\ifmst@abort
\newif\ifmst@optionalparam
```

`\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.

```
  \def\mst@enc{\encodingdefault}
  \def\mst@fam{\familydefault}
  \def\mst@ser{\seriesdefault}
  \def\mst@opsh{\shapedefault}  %% will be default shape for operator names
  \def\mst@bold{\bfdefault}
  \def\mst@ltsh{\shapedefault}  %% will be default shape for letters
```

`\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} %% v 1.15c
```

`Package options`

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

```
\newif\ifmst@italic
\newif\ifmst@frenchmath
    \DeclareOption{italic}{\mst@italictrue
        \def\mst@ltsh{\itdefault}}
    \DeclareOption{frenchmath}{\mst@frenchmathtrue\mst@italictrue
      \def\mst@ltsh{\itdefault}}
```

```
%%
\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
\edef\mst@tmp{\encodingdefault}\ifx\mst@oti\mst@tmp\else\mst@alldelimstrue\fi
    \DeclareOption{alldelims}{\mst@alldelimstrue}
    \DeclareOption{nolessnomore}{\mst@alldelimsfalse}
%% new with 1.2
\newif\ifmst@nosmalldelims
    \DeclareOption{nosmalldelims}{\mst@nosmalldelimstrue}
%%
\newif\ifmst@noplus
    \DeclareOption{noplus}{\mst@noplustrue}
\newif\ifmst@nominus
    \DeclareOption{nominus}{\mst@nominustrue}
\DeclareOption{noplusnominus}{\ExecuteOptions{noplus,nominus}}
%%
\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\string' is deprecated.^^J\space\space\space
         Check the documentation}}
    \DeclareOption{asterisk}{\mst@asterisktrue
      \AtBeginDocument{
        \everymath\expandafter{\the\everymath \mst@doasterisk
                               \let\mst@doasterisk\relax}
        \everydisplay\expandafter{\the\everydisplay \mst@doasterisk
                               \let\mst@doasterisk\relax}}}
\newif\ifmst@nospecials
    \DeclareOption{nospecials}{\mst@nospecialstrue}
\DeclareOption{basic}{\ExecuteOptions{noparenthesis,nopunctuation,%
noplusnominus,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@needsymbol
\newif\ifmst@symboldelimiters
    \DeclareOption{symboldelimiters}{\mst@needsymboltrue\mst@symboldelimiterstrue}
\newif\ifmst@symboldigits
    \DeclareOption{symboldigits}{\mst@needsymboltrue\mst@symboldigitstrue}
\newif\ifmst@symbolgreek
    \DeclareOption{symbolgreek}{\mst@needsymboltrue\mst@symbolgreektrue}
\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}
\newif\ifmst@eulergreek
    \DeclareOption{eulergreek}{\mst@needeulertrue\mst@eulergreektrue}
%%
\newif\ifmst@selfGreek
    \DeclareOption{selfGreek}{\mst@selfGreektrue}
\newif\ifmst@selfGreeks
    \DeclareOption{selfGreeks}{\mst@selfGreekstrue\mst@selfGreektrue}
\newif\ifmst@LGRgreek
    \DeclareOption{LGRgreek}{\mst@LGRgreektrue}
\newif\ifmst@LGRgreeks
    \DeclareOption{LGRgreeks}{\mst@LGRgreekstrue\mst@LGRgreektrue}
%%
\def\mst@greek@select{0}
\newif\ifmst@itgreek
\newif\ifmst@upgreek
    \DeclareOption{itgreek}{\mst@itgreektrue}
    \DeclareOption{upgreek}{\mst@upgreektrue}
    \DeclareOption{itGreek}{\def\mst@greek@select{1}}
    \DeclareOption{upGreek}{\def\mst@greek@select{2}}
%%
```

Starting with `1.15f` the meaning of the 'defaultxx' options has changed. They now prevent `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.

```
\newif\ifmst@defaultnormal
    \DeclareOption{defaultnormal}{\mst@defaultnormaltrue}
\newif\ifmst@defaultrm
    \DeclareOption{defaultrm}{\mst@defaultrmtrue}
```

```
\newif\ifmst@defaultbf
    \DeclareOption{defaultbf}{\mst@defaultbftrue}
\newif\ifmst@defaultit
    \DeclareOption{defaultit}{\mst@defaultittrue}
\newif\ifmst@defaultsf
    \DeclareOption{defaultsf}{\mst@defaultsftrue}
\newif\ifmst@defaultttt
    \DeclareOption{defaultttt}{\mst@defaulttttrue}
\newif\ifmst@nonormalbold
\DeclareOption{defaultalphabets}{\ExecuteOptions{defaultnormal,defaultrm,%
defaultbf,defaultit,defaultsf,defaultttt}\mst@nonormalboldtrue}
```

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

```
\newif\ifmst@defaultsizes
    \DeclareOption{defaultmathsizes}{\mst@defaultsizestrue}
\newif\ifmst@twelve
    \DeclareOption{12pt}{\mst@twelvetrue}
\newif\ifmst@fouriervec
    \DeclareOption{fouriervec}{\mst@fouriervectrue}
```

`1.15`: the subdued option.

```
\newif\ifmst@subdued
    \DeclareOption{subdued}{\mst@subduedtrue}
\DeclareOption*{\PackageWarningNoLine{mathastext}{Unknown option `\CurrentOption\string'}}

\ProcessOptions\relax
```

`\exists` `\mst@exists@skip` `\forall` `\mst@forall@skip`

`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 ∃ and ∀ 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.

```
\newmuskip\mst@exists@muskip  %% v 1.15e
\newmuskip\mst@forall@muskip
\def\mst@exists@skip{1mu}
\def\mst@forall@skip{.6667mu}
\ifmst@italic\ifmst@frenchmath\else
    \def\mst@exists@skip{0mu}\def\mst@forall@skip{0mu}\def\mst@prime@skip{0mu}
\fi\fi
\AtEndOfPackage{% must be at end of code
\let\mst@stdexists\exists
\let\mst@stdforall\forall
\renewcommand\exists{\mst@stdexists\mskip\mst@exists@muskip}
\renewcommand\forall{\mst@stdforall\mskip\mst@forall@muskip}
}
```

\prime
\mst@prime@skip
\active@math@prime
\MTnormalprime

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 txfonts works fine with upright letters). For this we replace the LaTeX kernel \active@math@prime with our own skip-enhanced version \mst@active@math@prime.

1.2b [2012/12/31]: doing

```
{\catcode`\'=\active \global\let'\mst@active@math@prime}
```

is awfully wrong when the right quote is made active at begin document by some other package (as happens with babel for some languages). So mathastext treats now the right quote with the same method as applied to the other characters it makes mathematically active. This uses the macro \mst@mathactivate which is defined later in the package.

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

```
\newmuskip\mst@prime@muskip   %% v 1.2
\def\mst@prime@skip{.5mu}
\ifmst@italic\ifmst@frenchmath\else\def\mst@prime@skip{0mu}\fi\fi
\def\mst@active@math@prime{\sp\bgroup\mskip\mst@prime@muskip\prim@s}
{\catcode`\'=12
\gdef\mst@modifyprime{\mst@mathactivate'{}\mst@active@math@prime}}
\AtBeginDocument{
\everymath\expandafter{\the\everymath \mst@modifyprime \MTnormalprime}
\everydisplay\expandafter{\the\everydisplay \mst@modifyprime \MTnormalprime}
}
\newcommand*\MTnormalprime{\let\mst@modifyprime\relax}
```

\MTexistsskip
\MTforallskip
\MTprimeskip

1.15e: These user macros set up the amount of mu skip or glue after \exists or \forall, on a math version basis (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), as each \Mathastext[⟨*version_name*⟩] stores the then 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.

```
\newcommand*\MTexistsskip[1]{\edef\mst@exists@skip{#1}%
    \mst@exists@muskip\mst@exists@skip\relax}
\newcommand*\MTforallskip[1]{\edef\mst@forall@skip{#1}%
    \mst@forall@muskip\mst@forall@skip\relax}
\newcommand*\MTprimeskip[1]{\edef\mst@prime@skip{#1}%
    \mst@prime@muskip\mst@prime@skip\relax}
\let\Mathastextexistsskip\MTexistsskip
\let\Mathastextforallskip\MTforallskip
\let\Mathastextprimeskip\MTprimeskip
\let\mathastextexistsskip\MTexistsskip
\let\mathastextforallskip\MTforallskip
\let\mathastextprimeskip\MTprimeskip
```

\resetMathstrut@ 2012/12/31: The amsmath macro \resetMathstrut@ is not compatible with a mathematically active opening parenthesis: it does

```
\mathchardef\@tempa\mathcode`\(\relax
```

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

Alas, with things such as $...\hbox{...$..$..}...$ mathastext will have already made the parenthesis (mathematically) active. And \boldsymbol from amsbsy disables the \@nomath switch and executes \mathversion{bold} directly in math mode. So we have a problem with \resetMathstrut@. lualatex-math replaces \resetMathstrut@ with its own version (which also looks at)). With lualatex no error is signaled when mathastext has done \mathcode`(="8000, but the \Mathstrutbox@ is then wrong.

The replacement macro avoids a potentially math active (. It assumes that there is still some appropriate glyph in slot 40 of operators and it sets the height and depth of \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 noparen was used, we leave everything untouched.

```
\ifmst@noparen\else
\AtBeginDocument{
\@ifundefined{resetMathstrut@}{}{%
    \PackageWarningNoLine{mathastext}{a custom version of the amsmath macro^^J%
      \space\space\space\string\resetMathstrut@\space will replace the original}
    \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}}
\fi
```

1.2 [2012/12/20] does some rather daring *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 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 \mathcode to "8000 through the macro \mst@mathactivate which in turn is put into the \everymath and \everydisplay token lists. No character is made active in the sense of the \catcode (the issues with catcode active characters at the entrance of the math mode are discussed later),

but the concerned characters will now expand in math mode to *two* tokens.

1.2c [2012/12/31]: hence, this current implementation puts constraints on the input: $x^?$ or $x\mathrel?y$ 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 mathastext 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 mathastext).

1.2e [2013/01/09]: alas $\ref{eq:1}$ still was a problem. Indeed in that case the mathastext prefix had no means to know it was inside a \ref so it made the character expand to its mathastext 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 on TeX StackExchange 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 mathastext 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, mathastext 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 `$\hbox{$?$}$`), 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 $?$}$` for an `?` which was unactive at the first `$` 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.

`\mst@do@az`
`\mst@do@AZ`
`\mst@addtodo@az`
`\mst@addtodo@AZ`
`\MTicinmath`
`\MTICinmath`
`\MTnoicinmath`
`\MTicalsoinmathxx`
`\mst@the`

`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!

Note: `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 used also to get the non-letters obey math alphabet while maintaining the TEX spacings, I could not use group braces. I should have added them later in the case of letters but then forgot. Fixed. (the group braces do not prevent ligatures when the letters are arguments to the math alphabet commands, the added macro `\mst@itcorr` then expands to nothing).

`\MTnoicinmath` can be used inside math mode (and has its scope limited by the group of course).

`\MTicalsoinmathxx` is destined to be used inside `\mathnormalbold` as I didn't want to add the complication of extracting the family number used inside `\mathnormalbold` (will perhaps come back if I have time to spend on `source2e`).

`\MTicinmath` can also be used inside math mode, but it can then only revert an earlier `\MTnoicinmath` inside the same math group. So the math mode had to be entered with italic corrections on.

`\MTnonlettersobeymathxx`
`\MTnonlettersdonotobeymathxx`

```
\newtoks\mst@do@nonletters
\newtoks\mst@do@easynonletters
\newtoks\mst@do@az
\newtoks\mst@do@AZ
\newcommand*\MTnonlettersdonotobeymathxx{\let\mst@the\@gobble}
\newcommand*\MTnonlettersobeymathxx{\let\mst@the\the}
\newcommand*\MTeasynonlettersdonotobeymathxx{\let\mst@theeasy\@gobble}
\newcommand*\MTeasynonlettersobeymathxx{\let\mst@theeasy\the}
\MTnonlettersdonotobeymathxx % 1.2c !!!
\MTeasynonlettersobeymathxx
\newcommand*\MTicinmath{%
    \let\mst@thef\the
    \let\mst@theF\the
    \def\mst@itcorr{\ifnum\fam=\m@ne\/\else\ifnum\fam=\symmtletterfont\/\fi\fi}%
    \let\mst@ITcorr\mst@itcorr}
```

```
\newcommand*\MTnoicinmath{% also usable in math mode
    \def\mst@itcorr{}\def\mst@ITcorr{}%
    \let\mst@thef\@gobble\let\mst@theF\@gobble}
\newcommand*\MTICinmath{%
    \let\mst@theF\the
    \def\mst@ITcorr{\ifnum\fam=\m@ne\/\fi}}
\newcommand*\MTnoICinmath{% also usable in math mode
    \def\mst@ITcorr{}%
    \let\mst@theF\@gobble}
\newcommand*\MTicalsoinmathxx{%
    \ifx\mst@itcorr\@empty\else\def\mst@itcorr{\/}\fi
    \ifx\mst@ITcorr\@empty\else\def\mst@ITcorr{\/}\fi}
\AtBeginDocument{
\everymath\expandafter{\the\everymath
    \mst@the\mst@do@nonletters \let\mst@the\@gobble
    \mst@theeasy\mst@do@easynonletters  \let\mst@theeasy\@gobble
    \mst@thef\mst@do@az \let\mst@thef\@gobble
    \mst@theF\mst@do@AZ \let\mst@theF\@gobble}
\everydisplay\expandafter{\the\everydisplay
    \mst@the\mst@do@nonletters \let\mst@the\@gobble
    \mst@theeasy\mst@do@easynonletters  \let\mst@theeasy\@gobble
    \mst@thef\mst@do@az \let\mst@thef\@gobble
    \mst@theF\mst@do@AZ \let\mst@theF\@gobble}
}
```

\mst@mathactivate
@addtodo@nonletters
todo@easynonletters
\mst@addtodo@az
\mst@addtodo@AZ

```
\def\mst@magic@v #1#2#3#4#5{#1#3#4}
\def\mst@magic@vi #1#2#3#4#5#6{#1#2#4#5}
\def\mst@fork{\ifmmode\mst@magic@v\fi\@thirdofthree}
\def\mst@safefork{\ifmmode\if@safe@actives\else\mst@magic@vi\fi\fi\@thirdofthree}
\def\mst@do@activecase#1#2#3{% #1 is a category 11 or 12 character token
    \@ifundefined{active@char#1}{}{%
         \ifcat #1a\mathcode`#1=#2\else
         \ifx\relax #2\relax\mathcode`#1=`#1 \else\mathcode`#1=#3\fi\fi
    \expandafter\expandafter\expandafter\let\expandafter\expandafter
                    \csname mst@orig@user@active#1\endcsname
                    \csname user@active#1\endcsname
    \expandafter\expandafter\expandafter\let\expandafter\expandafter
                    \csname mst@orig@normal@char#1\endcsname
                    \csname normal@char#1\endcsname
    \ifcat #1a%
      \expandafter\edef\csname user@active#1\endcsname
                    {\noexpand\mst@fork {{#2\noexpand#3}}{}\expandafter
                     \noexpand\csname mst@orig@user@active#1\endcsname
                    }%
      \expandafter\edef\csname normal@char#1\endcsname
                    {\noexpand\mst@safefork {{#2\noexpand#3}}{}\expandafter
                     \noexpand\csname mst@orig@normal@char#1\endcsname
                    }%
```

```
    \else
      \expandafter\edef\csname user@active#1\endcsname
                   {\noexpand\mst@fork {#2}\noexpand#3\expandafter
                    \noexpand\csname mst@orig@user@active#1\endcsname
                   }%
      \expandafter\edef\csname normal@char#1\endcsname
                   {\noexpand\mst@safefork {#2}\noexpand#3\expandafter
                    \noexpand\csname mst@orig@normal@char#1\endcsname
                   }%
    \fi}}
\begingroup % make sure ~ is active at mathastext loading time
  \catcode`~=\active
  \def\x{\endgroup
    \def\mst@mathactivate##1##2##3{% ##1 guaranteed of cat 11 or 12
      \begingroup
      \lccode`~=`##1
      \lccode`##1=`##1
      \lowercase{\endgroup
          \ifnum\catcode`##1=\active
            \mst@do@activecase ##1{##2}##3
            % careful as ##2 is empty in the asterisk and
            % prime case!
          \else
            \mathcode`##1="8000
            % I use " many times later, \string" everywhere would be a pain
            \ifcat##1a\def~{{##2##3}}\else\def~{##2##3}\fi
          \fi}}}
\x
\def\mst@addtodo@nonletters#1#2#3{%
  % #1 will be of cat 11 or 12.
  % #2 is  empty for asterisk and right quote
  \mst@do@nonletters\expandafter{\the\mst@do@nonletters
                                 \mst@mathactivate#1{#2}#3}%
}
\def\mst@addtodo@easynonletters#1#2{% #1 is a one char control sequence
  \mst@do@easynonletters\expandafter{\the\mst@do@easynonletters\mathcode`#1=#2}%
}
\def\mst@addtodo@az#1#2#3{%
  \mst@do@az\expandafter{\the\mst@do@az\mst@mathactivate#1#2#3}
}
\def\mst@addtodo@AZ#1#2#3{%
  \mst@do@AZ\expandafter{\the\mst@do@AZ\mst@mathactivate#1#2#3}%
}
```

`\newmcodes@` `1.15d`: this 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).

```
\@ifpackageloaded{amsmath}
```

```
{\let\newmcodes@\relax} % brutal, but bye bye and don't come back.
{}
```

subdued 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 LaTeX 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[1]. 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 ((1): but the *euler* package for example assigns the digits to the *letters* symbol font...)

`1.15d`: Oct 13, 2012. The `\mathcode` thing has to be used with care under Unicode engines. Unfortunately the `\luatexUmathcode` macro is helpless [update: I have since been told by TH that the next LuaLaTeX release will change this] 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.

`1.15e`: Oct 22, 2012. I add the necessary things to also subdue the `\mathbf`, `\mathit`, `\mathsf` and `\mathtt` macros (previous version only took care of the symbol alphabets `\mathnormal` and `\mathrm`.) [update: `1.15f` does that in a completely different and much simpler way] Notice that the package defines a `\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 mtletterfont and 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 LuaLaTeX problem with `\luatexUmathcode`). 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 `\mathversion{normal}` and not only `\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.

```
\ifmst@subdued
  \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
        \def\mst@normalmv@letter{#2}}
  \expandafter\mst@reserved\mv@normal\@nil
  \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
        \def\mst@boldmv@letter{#2}}
  \expandafter\mst@reserved\mv@bold\@nil
  \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
        \def\mst@normalmv@operator{#2}}
  \expandafter\mst@reserved\mv@normal\@nil
  \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
        \def\mst@boldmv@operator{#2}}
  \expandafter\mst@reserved\mv@bold\@nil
%%
  \edef\mst@tmp{\encodingdefault}
```

```
\def\mst@reserved#1/#2/#3/#4/{\gdef\mst@debut{#1}\gdef\mst@reste{#2/#3/#4}}
\begingroup\escapechar\m@ne
    \xdef\mst@funnyoti{\expandafter\string\csname OT1\endcsname}
    \expandafter\expandafter\expandafter
        \mst@reserved\expandafter\string\mst@normalmv@operator/
\endgroup
\ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
\edef\mst@normalmv@operator{\expandafter\noexpand\csname
                              \mst@tmp/\mst@reste\endcsname}
\begingroup\escapechar\m@ne
    \expandafter\expandafter\expandafter
        \mst@reserved\expandafter\string\mst@boldmv@operator/
\endgroup
\ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
\edef\mst@boldmv@operator{\expandafter\noexpand\csname
                            \mst@tmp/\mst@reste\endcsname}
\AtEndOfPackage{
\typeout{** ...entering subdued mode...}
\expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@letter\symmtletterfont
\expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@letter\symmtletterfont
\expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@operator\symmtoperatorfon
\expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@operator\symmtoperatorfont
\typeout{** ...done.}
}
\fi % fin de ce \ifmst@subdued
```

In the short-lived `1.15e` I was doing the following for alphabets:

```
\def\mst@reservedc#1#2#3#4{\def\mst@normalmv@mathbf{#4#3}}
\def\mst@reserveda#1{%
\def\mst@reservedb##1\install@mathalphabet#1##2##3\@nil{\mst@reservedc##2}%
\expandafter\mst@reservedb\mv@normal\@nil}
\expandafter\mst@reserveda\csname mathbf\space\endcsname
```

and later in the code:

```
\expandafter\expandafter\expandafter\SetMathAlphabet@
\expandafter\expandafter\expandafter\mv@normal
\expandafter\mst@normalmv@mathbf\csname Mathbf\space \endcsname\Mathbf
```

It does work! but `\let\mst@original@bf\mathbf` is so much simpler. And also safer, because `\mathbf` could have been redefined using `\DeclareSymbolFontAlphabet`... (I could have provided the necessary check to the already bloated code...)

`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.

```
\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 `\operator@font` will always refer to the 'operators' math font might be in for a surprise... well, we'll see.

`\MTmathoperators-obeymathxx` `1.2`: rather than just replacing `\symoperators` by `\symmtoperatorfont` I add a modification which makes the declared operator names sensitive to the math alphabets... ouh le vilain!

`\MTmathoperators-donot-obeymathxx`

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

`mtletterfont` In version `1.1`, we add the possibility to mimick the standard behavior, that is to have italic letters and upright digits. Thanks to Tariq Perwez and Kevin Klement who asked for such a feature.

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

`\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... and (`1.15f`) the previous status of the math alphabets is recorded for the sake of the `subdued` option.

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`.

```
\let\mst@alph@omicron\mathnormal
\@ifpackageloaded{fourier}{\ifsloped\else\let\mst@alph@omicron\mathrm\fi}{}
\DeclareSymbolFontAlphabet{\Mathnormal}{mtletterfont}
\DeclareSymbolFontAlphabet{\Mathrm}{mtoperatorfont}
\ifmst@nonormalbold\else
   \DeclareMathAlphabet{\mathnormalbold}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@ltsh}
\fi\ifmst@defaultbf\else
   \DeclareMathAlphabet{\Mathbf}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@opsh}
\fi\ifmst@defaultit\else
   \DeclareMathAlphabet{\Mathit}{\mst@enc}{\mst@fam}{\mst@ser}{\itdefault}
\fi\ifmst@defaultsf\else
   \DeclareMathAlphabet{\Mathsf}{\mst@enc}{\sfdefault}{\mst@ser}{\mst@opsh}
\fi\ifmst@defaulttt\else
   \DeclareMathAlphabet{\Mathtt}{\mst@enc}{\ttdefault}{\mst@ser}{\mst@opsh}
\fi
\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}
\def\mst@setalphabets{%
    \ifmst@defaultnormal\else\renewcommand{\mathnormal}{\Mathnormal}\fi
    \ifmst@defaultrm\else\renewcommand{\mathrm}{\Mathrm}\fi
    \ifmst@defaultbf\else\renewcommand{\mathbf}{\Mathbf}\fi
    \ifmst@defaultit\else\renewcommand{\mathit}{\Mathit}\fi
    \ifmst@defaultsf\else\renewcommand{\mathsf}{\Mathsf}\fi
    \ifmst@defaultTT\else\renewcommand{\mathtt}{\Mathtt}\fi}
\ifmst@subdued\else\mst@setalphabets\fi
```

LGRgreek
selfGreek
mtlgrfontupper
mtlgrfontlower
mtselfGreekfont

`1.14b`: We can not 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 `\MTitgreek` etc... will be used in-between calls to `\Mathastext` and re-adjust the shapes. And the command `\MTgreekfont` changes the Greek font family.

```
\def\mst@update@greeksh{
  \def\mst@greek@lsh{\mst@ltsh} %% default behavior
  \def\mst@greek@ush{\mst@opsh}
    \ifmst@itgreek\def\mst@greek@lsh{\itdefault}
                      \def\mst@greek@ush{\itdefault}\fi
    \ifmst@upgreek\def\mst@greek@lsh{\updefault}
                      \def\mst@greek@ush{\updefault}\fi
    \ifmst@frenchmath
      \ifmst@itgreek\else
      \ifmst@upgreek\else
        \def\mst@greek@lsh{\mst@opsh}
        \def\mst@greek@ush{\mst@opsh}
      \fi\fi
    \fi
    \ifcase\mst@greek@select
        \or\def\mst@greek@ush{\itdefault}
        \or\def\mst@greek@ush{\updefault}
    \fi}
\mst@update@greeksh
```

```
\ifmst@LGRgreek
    \DeclareFontEncoding{LGR}{}{}
    \DeclareSymbolFont{mtlgrfontlower}{LGR}
            {\mst@fam}{\mst@ser}{\mst@greek@lsh}
    \DeclareSymbolFont{mtlgrfontupper}{LGR}
            {\mst@fam}{\mst@ser}{\mst@greek@ush}
\else
\ifmst@selfGreek
        \DeclareSymbolFont{mtselfGreekfont}{OT1}
            {\mst@fam}{\mst@ser}{\mst@greek@ush}
\fi\fi
```

mteulervm
\MathEuler
\MathEulerBold

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

```
\ifmst@needeuler\typeout{** will use Euler font; command \string\MTEulerScale}
\DeclareSymbolFont{mteulervm}{U}{zeur}{m}{n}
\DeclareSymbolFontAlphabet{\MathEuler}{mteulervm}
\DeclareMathAlphabet{\MathEulerBold}{U}{zeur}{\mst@bold}{n}
\fi
\newcommand*\MTEulerScale[1]{\edef\zeu@Scale{#1}}
\let\MathastextEulerScale\MTEulerScale
```

LaTeX 2_ε has a strange initial configuration where the capital Greek letters are of type `mathalpha`, but the lower Greek letters of type `mathord`, so that `\mathbf` does not act on them, although lowercase Greek letters and Latin letters are from the same font. This is because `\mathbf` is set up to be like a bold version of `\mathrm`, and `\mathrm` uses the 'operators' font, by default `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 `cmmi` and `cmr`.

The package eulervm sets the lowercase Greek letters to be of type `mathalpha`, the default `\mathbf` and `\mathrm` will act wierdly on them, but a `\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 `mathastext`, Latin letters and Greek letters (lower and upper case) must be (generally) assumed to come from two different fonts, as a result the standard `\mathbf` (and `\mathrm`) will give weird results when used for Greek letters. We could coerce `\mathbf` to do something reasonable (cf `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 `eulervm` package and defineed `\MathEuler` and `\MathEulerBold` alphabet commands (the `eulervm` package does this only for the bold font).

mtpsymbol
\MathPSymbol

In case we need the Symbol font, we declare it here. The macro `\psy@scale` will be used to scale the font (see at the very end of this file).

```
\ifmst@needsymbol\typeout{** will use Symbol font; command \string\MTSymbolScale}
 \def\psy@scale{1}
 \DeclareSymbolFont{mtpsymbol}{U}{psy}{m}{n}
 \DeclareSymbolFontAlphabet{\MathPSymbol}{mtpsymbol}
\AtBeginDocument{
  \DeclareFontFamily{U}{psy}{}
  \DeclareFontShape{U}{psy}{m}{n}{<->s*[\psy@scale] psyr}{}
```

```
}
\fi
\newcommand*\MTSymbolScale[1]{\edef\psy@scale{#1}}
\let\MathastextSymbolScale\MTSymbolScale
```

I did not choose for name `\MathSymbol` as this may be defined somewhere for another thing. There is no bold for the postscript Symbol font distributed with the LaTeX 2ε `psnffs` core package.

`\pmvec` Definition of a poor man version of the `\vec` accent

```
\DeclareRobustCommand*\pmvec[1]{\mathord{\stackrel{\raisebox{-.5ex}%
{\tiny\boldmath$\mathord{\rightarrow}$}}{{}#1}}}
```

`\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.

```
\ifmst@fouriervec
  \DeclareFontEncoding{FML}{}{}
  \DeclareFontSubstitution{FML}{futm}{m}{it}
  \DeclareSymbolFont{mathastextfourier}{FML}{futm}{m}{it}
  \SetSymbolFont{mathastextfourier}{bold}{FML}{futm}{b}{it}
  \DeclareMathAccent{\fouriervec}{\mathord}{mathastextfourier}{"7E}
\fi
```

`\MTencoding` `\MTfamily` `\MTseries` `\MTshape` `\MTboldvariant` `\MTlettershape` 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...

```
\newcommand*\MTencoding[1]{\def\mst@enc{#1}}
\newcommand*\MTfamily[1]{\def\mst@fam{#1}}
\newcommand*\MTseries[1]{\def\mst@ser{#1}}
\newcommand*\MTshape[1]{\def\mst@opsh{#1}
  \ifmst@italic\else\def\mst@ltsh{#1}\fi}
\newcommand*\MTboldvariant[1]{\def\mst@bold{#1}}
\newcommand*\MTlettershape[1]{\def\mst@ltsh{#1}}
\@onlypreamble\MTencoding
\@onlypreamble\MTfamily
\@onlypreamble\MTseries
\@onlypreamble\MTshape
\@onlypreamble\MTboldvariant
\@onlypreamble\MTlettershape
\let\Mathastextencoding\MTencoding
\let\Mathastextfamily\MTfamily
\let\Mathastextseries\MTseries
\let\Mathastextshape\MTshape
\let\Mathastextboldvariant\MTboldvariant
\let\Mathastextlettershape\MTlettershape
```

\MTitgreek
\MTupgreek
\MTitGreek
\MTupGreek
\MTgreekfont

1.15c: These new macros can be used in-between calls to \Mathastext. They reset the shapes for Greek letters (applies to LGRgreek(s) and selfGreek(s) options). The \MTgreekfont presupposes either LGRgreek or selfGreek. \MTgreekfont{\familydefault} acts like turning on LGRgreeks or selfGreeks.

```
    \newcommand*\MTitgreek{%
\mst@itgreektrue\mst@upgreekfalse\def\mst@greek@select{0}}
    \newcommand*\MTupgreek{%
\mst@upgreektrue\mst@itgreekfalse\def\mst@greek@select{0}}
    \newcommand*\MTitGreek{\def\mst@greek@select{1}}
    \newcommand*\MTupGreek{\def\mst@greek@select{2}}
\@onlypreamble\MTitgreek
\@onlypreamble\MTupgreek
\@onlypreamble\MTitGreek
\@onlypreamble\MTitGreek
\let\Mathastextitgreek\MTitgreek
\let\Mathastextupgreek\MTupgreek
\let\MathastextitGreek\MTitGreek
\let\MathastextupGreek\MTupGreek
    \newcommand*\MTgreekfont[1]{\def\mst@greekfont{#1}}
\@onlypreamble\MTgreekfont
\let\Mathastextgreekfont\MTgreekfont
```

\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 \Mathastext.

```
\newcommand*\MTWillUse[5][\empty]{
  \MTencoding{#2}
  \MTfamily{#3}
  \MTseries{#4}
  \MTshape{#5}
    \ifmst@italic\MTlettershape{\itdefault}\fi % was missing in v 1.14 and prior
  \ifx\empty#1
  \else
%% we check if optional argument was [],  this does not detect [ ]
  \def\mst@tmp{#1}
   \ifx\empty\mst@tmp\else\MTlettershape{#1}\fi
  \fi
  \Mathastext}
\@onlypreamble\MTWillUse
\let\MathastextWillUse\MTWillUse
\let\Mathastextwilluse\MTWillUse
```

\Mathastext

The command \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 mtletterfont is for letters, and the mtoperatorfont for digits and log-like operator names. The default is that both are upright.

Starting with version 1.12, an optional argument makes \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.

```
\newcommand*\Mathastext[1][\empty]{
\ifx\empty#1 % no optional argument
  \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{\mst@exists@skip}
  \edef\mst@forall@skip@normal{\mst@forall@skip}
  \edef\mst@prime@skip@normal{\mst@prime@skip}
  \edef\mst@encoding@bold{\mst@enc}
  \edef\mst@family@bold{\mst@fam}
  \edef\mst@series@bold{\mst@bold}
  \edef\mst@shape@bold{\mst@opsh}
  \edef\mst@ltshape@bold{\mst@ltsh}
  \edef\mst@boldvariant@bold{\mst@bold}
  \edef\mst@itdefault@bold{\itdefault}
  \edef\mst@rmdefault@bold{\rmdefault}
  \edef\mst@sfdefault@bold{\sfdefault}
  \edef\mst@ttdefault@bold{\ttdefault}
  \edef\mst@exists@skip@bold{\mst@exists@skip}
  \edef\mst@forall@skip@bold{\mst@forall@skip}
  \edef\mst@prime@skip@bold{\mst@prime@skip}
  \MTicinmath
  \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@ltshape@normal\MTnoicinmath\fi}%
  \ifmst@frenchmath
      \MTICinmath
      \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@shape@normal\MTnoICinmath\fi}%
  \fi
  \ifmst@subdued
    \MTeasynonlettersdonotobeymathxx
    \MTnonlettersdonotobeymathxx
    \MTmathoperatorsdonotobeymathxx
    \MTnoicinmath
    \ifmst@asterisk\let\mst@doasterisk\relax\fi
    \def\mst@exists@skip@normal{0mu}
    \def\mst@forall@skip@normal{0mu}
    \def\mst@prime@skip@normal{0mu}
    \def\mst@exists@skip@bold{0mu}
    \def\mst@forall@skip@bold{0mu}
    \def\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
       \def\mst@exists@skip@normal{0mu}
       \def\mst@forall@skip@normal{0mu}
       \def\mst@prime@skip@normal{0mu}
       \def\mst@exists@skip@bold{0mu}
       \def\mst@forall@skip@bold{0mu}
       \def\mst@prime@skip@bold{0mu}
      \fi
    \else
        \mst@exists@muskip\mst@exists@skip\relax
        \mst@forall@muskip\mst@forall@skip\relax
        \mst@prime@muskip\mst@prime@skip\relax
    \fi
  \fi
%% v1.15f
\ifmst@nonormalbold\else
   \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}
\fi
%% v1.15f adds \ifmst@default.. checks
\ifmst@subdued\else
   \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}
\ifmst@defaultbf\else
   \SetMathAlphabet{\Mathbf}{normal}{\mst@encoding@normal}
```

```
                                    {\mst@family@normal}
                                    {\mst@series@bold}
                                    {\mst@shape@normal}
    \SetMathAlphabet{\Mathbf}{bold}{\mst@encoding@bold}
                                    {\mst@family@bold}
                                    {\mst@series@bold}
                                    {\mst@shape@bold}
\fi\ifmst@defaultit\else
    \SetMathAlphabet{\Mathit}{normal}{\mst@encoding@normal}
                                    {\mst@family@normal}
                                    {\mst@series@normal}
                                    {\mst@itdefault@normal}
    \SetMathAlphabet{\Mathit}{bold}{\mst@encoding@bold}
                                    {\mst@family@bold}
                                    {\mst@series@bold}
                                    {\mst@itdefault@bold}
\fi\ifmst@defaultsf\else
    \SetMathAlphabet{\Mathsf}{normal}{\mst@encoding@normal}
                                    {\mst@sfdefault@normal}
                                    {\mst@series@normal}
                                    {\mst@shape@normal}
    \SetMathAlphabet{\Mathsf}{bold}{\mst@encoding@bold}
                                    {\mst@sfdefault@bold}
                                    {\mst@series@bold}
                                    {\mst@shape@bold}
\fi\ifmst@defaulttt\else
    \SetMathAlphabet{\Mathtt}{normal}{\mst@encoding@normal}
                                    {\mst@ttdefault@normal}
                                    {\mst@series@normal}
                                    {\mst@shape@normal}
    \SetMathAlphabet{\Mathtt}{bold}{\mst@encoding@bold}
                                    {\mst@ttdefault@bold}
                                    {\mst@series@bold}
                                    {\mst@shape@bold}
\fi\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@needeuler
   \SetSymbolFont{mteulervm}{bold}{U}{zeur}{\mst@boldvariant@normal}{n}
   \SetMathAlphabet{\MathEulerBold}{normal}
               {U}{zeur}{\mst@boldvariant@normal}{n}
   \SetMathAlphabet{\MathEulerBold}{bold}
               {U}{zeur}{\mst@boldvariant@normal}{n}
\fi

\ifmst@needsymbol\SetSymbolFont{mtpsymbol}{bold}
                    {U}{psy}{\mst@boldvariant@normal}{n}
```

```
\fi
```

LGRgreek* LGRgreek, LGRgreeks, selfGreek, and selfGreeks options.
selfGreek*

```
\ifmst@subdued\else
\ifmst@LGRgreek
   \SetSymbolFont{mtlgrfontlower}{normal}{LGR}
            {\mst@greekfont}{\mst@series@normal}{\mst@greek@lsh}
   \SetSymbolFont{mtlgrfontlower}{bold}{LGR}
            {\mst@greekfont}{\mst@boldvariant@normal}{\mst@greek@lsh}
   \SetSymbolFont{mtlgrfontupper}{normal}{LGR}
            {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
   \SetSymbolFont{mtlgrfontupper}{bold}{LGR}
            {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
 \else
\ifmst@selfGreek
   \SetSymbolFont{mtselfGreekfont}{normal}{OT1}
            {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
   \SetSymbolFont{mtselfGreekfont}{bold}{OT1}
            {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
\fi\fi\fi

\ifmst@subdued
\typeout{** subdued mode will be activated for the normal and bold math versions}
\else
\typeout{** Latin letters in the normal (resp. bold) math versions are now^^J%
    ** set up to use the fonts
\mst@encoding@normal/\mst@family@normal/\mst@series@normal%
(\mst@boldvariant@normal)/\mst@ltshape@normal}
\ifmst@LGRgreek\typeout{** Greek letters (\mst@greek@lsh/\mst@greek@ush)
will use LGR/\mst@greekfont}
\fi
\ifmst@nodigits\else\typeout{** Other characters (digits, ...)
and \protect\log-like names will be^^J%
** typeset with the \mst@shape@normal\space shape.}
\fi
\fi
\else % optional argument, then we rather do DeclareVersion
 \edef\mst@tmp{#1}
 \ifx\mst@tmp\empty
    \Mathastext  % no infinite loop!
  \else
    \MTDeclareVersion[\mst@ltsh]{#1}
         {\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}
 \fi
\fi
}
\@onlypreamble\Mathastext
\let\mathastext\Mathastext
```

\MTDeclareVersion 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!

```
\newcommand*\MTDeclareVersion[6][\empty]{
  \edef\mst@version{#2}
  \DeclareMathVersion{\mst@version}
  \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}
  \mst@optionalparamfalse
    \ifx\empty#1
    \else % we check nevertheless for [] situation, will not work with [ ]
       \def\mst@tmp{#1}
       \ifx\mst@tmp\empty\else\mst@optionalparamtrue\fi
    \fi
  \ifmst@optionalparam
      \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}
    \typeout{** Latin letters in math version '#2\string' will use the font
    #3/#4/#5/#1^^J%
    ** Other characters (digits, ...) and \protect\log-like
    names will be in #6 shape.}
      \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#1}
  \else
    \ifmst@italic
      \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{\mst@ltsh}
    \typeout{** Latin letters in math version '#2\string' will use the font
    #3/#4/#5/\mst@ltsh^^J%
    ** Other characters (digits, ...) and \protect\log-like names
                    will be in #6 shape.}
      \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{\mst@ltsh}
    \else
```

```
      \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#6}
    \typeout{** Latin letters in math version '#2\string' will use the fonts
    #3/#4/#5(\mst@bold)/#6}
      \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#6}
    \fi
  \fi
  \ifmst@nonormalbold\else
  \SetMathAlphabet{\mathnormalbold}{#2}{#3}{#4}{\mst@bold}
                                  {\csname
                                    mst@ltshape@\mst@version\endcsname}
  \fi
  \SetSymbolFont{mtoperatorfont}{#2}{#3}{#4}{#5}{#6}
  \ifmst@defaultbf\else\SetMathAlphabet{\Mathbf}{#2}{#3}{#4}{\mst@bold}{#6}\fi
  \ifmst@defaultit\else\SetMathAlphabet{\Mathit}{#2}{#3}{#4}{#5}{\itdefault}\fi
  \ifmst@defaultsf\else\SetMathAlphabet{\Mathsf}{#2}{#3}{\sfdefault}{#5}{#6}\fi
  \ifmst@defaultt\else\SetMathAlphabet{\Mathtt}{#2}{#3}{\ttdefault}{#5}{#6}\fi
  \ifmst@needeuler
      \SetMathAlphabet{\MathEulerBold}{#2}{U}{zeur}{\mst@bold}{n}
  \fi
```

LGRgreeks selfGreeks

In the case of option LGRgreeks (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.

```
  \def\mst@greek@lsh@loc{\csname mst@ltshape@\mst@version\endcsname}
  \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
    \ifmst@itgreek\def\mst@greek@lsh@loc{\itdefault}
                      \def\mst@greek@ush@loc{\itdefault}\fi
    \ifmst@upgreek\def\mst@greek@lsh@loc{\updefault}
                      \def\mst@greek@ush@loc{\updefault}\fi
    \ifmst@frenchmath
      \ifmst@itgreek\else
      \ifmst@upgreek\else
        \def\mst@greek@lsh@loc{\csname mst@shape@\mst@version\endcsname}
        \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
      \fi\fi
    \fi
    \ifcase\mst@greek@select
        \or\def\mst@greek@ush@loc{\itdefault}
        \or\def\mst@greek@ush@loc{\updefault}
    \fi
\ifmst@LGRgreeks
      \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{#4}{#5}{\mst@greek@lsh@loc}
      \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{#4}{#5}{\mst@greek@ush@loc}
   \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
     LGR font #4}
\else
\ifmst@selfGreeks
      \SetSymbolFont{mtselfGreekfont}{#2}{OT1}{#4}{#5}{\mst@greek@ush@loc}
```

```
    \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
      will use OT1 font #4}
  \else
  \ifmst@LGRgreek
       \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@lsh@loc}
       \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
    \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
      LGR font \mst@greekfont}
  \else
  \ifmst@selfGreek
       \SetSymbolFont{mtselfGreekfont}{#2}{OT1}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
    \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
      will use OT1 font \mst@greekfont}
  \fi\fi\fi\fi
}
\@onlypreamble\MTDeclareVersion

\let\MathastextDeclareVersion\MTDeclareVersion
```

`\MTVersion` This is a wrapper around LaTeX's `\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 `\exists` and `\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...

```
\newcommand*\MTVersion[2][\empty]{%
    \mathversion{#2}%
    \ifx\empty#1% no optional argument
        \def\mst@tmp{#2}%
    \else% we check nevertheless for [] situation, will not work with [ ]
       \def\mst@tmpa{#1}%
       \ifx\mst@tmpa\empty\def\mst@tmp{#2}%
       \else\def\mst@tmp{#1}%
       \fi%
    \fi%
    \usefont{\csname mst@encoding@\mst@tmp\endcsname}%
        {\csname mst@family@\mst@tmp\endcsname}%
        {\csname mst@series@\mst@tmp\endcsname}%
        {\csname mst@shape@\mst@tmp\endcsname}%
    \edef\mst@@encoding{\csname mst@encoding@\mst@tmp\endcsname}%
        \renewcommand{\encodingdefault}{\mst@@encoding}%
%% some modifications for v1.15b
    \edef\mst@@famdefault{\csname mst@family@\mst@tmp\endcsname}%
```

```
        \renewcommand{\familydefault}{\mst@@famdefault}%
    \edef\mst@@series{\csname mst@series@\mst@tmp\endcsname}%
        \renewcommand{\mddefault}{\mst@@series}%
    \edef\mst@@shape{\csname mst@shape@\mst@tmp\endcsname}%
        \renewcommand{\shapedefault}{\mst@@shape}%
    \edef\mst@@boldvariant{\csname mst@boldvariant@\mst@tmp\endcsname}%
        \renewcommand{\bfdefault}{\mst@@boldvariant}%
    \edef\mst@@itdefault{\csname mst@itdefault@\mst@tmp\endcsname}%
        \renewcommand{\itdefault}{\mst@@itdefault}%
    \edef\mst@@rmdefault{\csname mst@rmdefault@\mst@tmp\endcsname}%
        \renewcommand{\rmdefault}{\mst@@rmdefault}%
    \edef\mst@@sfdefault{\csname mst@sfdefault@\mst@tmp\endcsname}%
        \renewcommand{\sfdefault}{\mst@@sfdefault}%
    \edef\mst@@ttdefault{\csname mst@ttdefault@\mst@tmp\endcsname}%
        \renewcommand{\ttdefault}{\mst@@ttdefault}%
    \edef\mst@tmp{#2}%
%% v1.15e: muskips
    \expandafter
     \mst@exists@muskip\csname mst@exists@skip@\mst@tmp\endcsname\relax
    \expandafter
     \mst@forall@muskip\csname mst@forall@skip@\mst@tmp\endcsname\relax
%% v1.2: muskip for \cs{prime}
    \expandafter
     \mst@prime@muskip\csname mst@prime@skip@\mst@tmp\endcsname\relax
%% v1.2b: italic corrections except for italic/slanted (sic) letters, and of
%% course except in the subdued normal and bold math versions
    \edef\mst@tmpa{\csname mst@ltshape@\mst@tmp\endcsname}%
    \edef\mst@tmpb{\csname mst@shape@\mst@tmp\endcsname}%
    \MTicinmath
    \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@tmpa\MTnoicinmath\fi}%
    \ifmst@frenchmath
      \MTICinmath
      \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@tmpb\MTnoICinmath\fi}%
    \fi
%% v1.15c: extending subdued to LGRgreek
%% v1.15f: subdueing math alphabets in a simpler way than in 1.15e
%% v1.2b: subdueing the activation of characters in math mode
%% v1.2d: special treatment of the asterisk
    \MTmathoperatorsobeymathxx
    \MTeasynonlettersobeymathxx
    \ifmst@asterisk\let\mst@doasterisk\mst@@doasterisk\fi
    \ifmst@subdued
      \edef\mst@tmp{#2}\def\mst@tmpa{normal}%
      \ifx\mst@tmp\mst@tmpa
        \mst@restorealphabets
        \MTmathoperatorsdonotobeymathxx
        \MTnonlettersdonotobeymathxx
        \MTeasynonlettersdonotobeymathxx
        \MTnoicinmath
```

```
        \ifmst@asterisk\let\mst@doasterisk\relax\fi
        \ifmst@LGRgreek\mst@restoregreek\fi
      \else
      \def\mst@tmpa{bold}%
      \ifx\mst@tmp\mst@tmpa
          \mst@restorealphabets
          \MTmathoperatorsdonotobeymathxx
          \MTnonlettersdonotobeymathxx
          \MTeasynonlettersdonotobeymathxx
          \MTnoicinmath
          \ifmst@asterisk\let\mst@doasterisk\relax\fi
          \ifmst@LGRgreek\mst@restoregreek\fi
      \else
      \mst@setalphabets
      \ifmst@LGRgreek\mst@setuplgrgreek\fi
    \fi\fi\fi
}
\let\MathastextVersion\MTVersion
\let\Mathastextversion\MTVersion
\let\MTversion\MTVersion
\let\mathastextversion\MTVersion
```

\Mathastext Initialization call:

```
\Mathastext
```

Additional appropriate messages to the terminal and the log.

```
\ifmst@eulergreek
      \typeout{** Greek letters will use the Euler font. Use^^J%
** \protect\MathastextEulerScale{<factor>} to scale the font.}
\else
\ifmst@symbolgreek
       \typeout{** Greek letters will use the PostScript Symbol font. Use^^J%
** \protect\MathastextSymbolScale{<factor>} to scale the font.}
\fi\fi
```

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 uses \mathcode‘x="8000 to insert the italic corrections after letters.

```
\def\mst@DeclareMathSymbol#1#2#3#4{%
    \DeclareMathSymbol{#1}{#2}{#3}{#4}
    \expandafter\DeclareMathSymbol\csname mst@#1\endcsname{#2}{#3}{#4}
    \expandafter\mst@addtodo@az
        \expandafter#1\csname mst@#1\endcsname\mst@itcorr}
\mst@DeclareMathSymbol{a}{\mathalpha}{mtletterfont}{‘a}
\mst@DeclareMathSymbol{b}{\mathalpha}{mtletterfont}{‘b}
\mst@DeclareMathSymbol{c}{\mathalpha}{mtletterfont}{‘c}
\mst@DeclareMathSymbol{d}{\mathalpha}{mtletterfont}{‘d}
\mst@DeclareMathSymbol{e}{\mathalpha}{mtletterfont}{‘e}
\mst@DeclareMathSymbol{f}{\mathalpha}{mtletterfont}{‘f}
\mst@DeclareMathSymbol{g}{\mathalpha}{mtletterfont}{‘g}
```

```
\mst@DeclareMathSymbol{h}{\mathalpha}{mtletterfont}{`h}
\mst@DeclareMathSymbol{i}{\mathalpha}{mtletterfont}{`i}
\mst@DeclareMathSymbol{j}{\mathalpha}{mtletterfont}{`j}
\mst@DeclareMathSymbol{k}{\mathalpha}{mtletterfont}{`k}
\mst@DeclareMathSymbol{l}{\mathalpha}{mtletterfont}{`l}
\mst@DeclareMathSymbol{m}{\mathalpha}{mtletterfont}{`m}
\mst@DeclareMathSymbol{n}{\mathalpha}{mtletterfont}{`n}
\mst@DeclareMathSymbol{o}{\mathalpha}{mtletterfont}{`o}
\mst@DeclareMathSymbol{p}{\mathalpha}{mtletterfont}{`p}
\mst@DeclareMathSymbol{q}{\mathalpha}{mtletterfont}{`q}
\mst@DeclareMathSymbol{r}{\mathalpha}{mtletterfont}{`r}
\mst@DeclareMathSymbol{s}{\mathalpha}{mtletterfont}{`s}
\mst@DeclareMathSymbol{t}{\mathalpha}{mtletterfont}{`t}
\mst@DeclareMathSymbol{u}{\mathalpha}{mtletterfont}{`u}
\mst@DeclareMathSymbol{v}{\mathalpha}{mtletterfont}{`v}
\mst@DeclareMathSymbol{w}{\mathalpha}{mtletterfont}{`w}
\mst@DeclareMathSymbol{x}{\mathalpha}{mtletterfont}{`x}
\mst@DeclareMathSymbol{y}{\mathalpha}{mtletterfont}{`y}
\mst@DeclareMathSymbol{z}{\mathalpha}{mtletterfont}{`z}
\ifmst@frenchmath\def\mst@font@tbu{mtoperatorfont}
   \else\def\mst@font@tbu{mtletterfont}\fi
\def\mst@DeclareMathSymbol#1#2#3#4{%
    \DeclareMathSymbol{#1}{#2}{#3}{#4}
    \expandafter\DeclareMathSymbol\csname mst@#1\endcsname{#2}{#3}{#4}
    \expandafter\mst@addtodo@AZ
        \expandafter#1\csname mst@#1\endcsname\mst@ITcorr}
\mst@DeclareMathSymbol{A}{\mathalpha}{\mst@font@tbu}{`A}
\mst@DeclareMathSymbol{B}{\mathalpha}{\mst@font@tbu}{`B}
\mst@DeclareMathSymbol{C}{\mathalpha}{\mst@font@tbu}{`C}
\mst@DeclareMathSymbol{D}{\mathalpha}{\mst@font@tbu}{`D}
\mst@DeclareMathSymbol{E}{\mathalpha}{\mst@font@tbu}{`E}
\mst@DeclareMathSymbol{F}{\mathalpha}{\mst@font@tbu}{`F}
\mst@DeclareMathSymbol{G}{\mathalpha}{\mst@font@tbu}{`G}
\mst@DeclareMathSymbol{H}{\mathalpha}{\mst@font@tbu}{`H}
\mst@DeclareMathSymbol{I}{\mathalpha}{\mst@font@tbu}{`I}
\mst@DeclareMathSymbol{J}{\mathalpha}{\mst@font@tbu}{`J}
\mst@DeclareMathSymbol{K}{\mathalpha}{\mst@font@tbu}{`K}
\mst@DeclareMathSymbol{L}{\mathalpha}{\mst@font@tbu}{`L}
\mst@DeclareMathSymbol{M}{\mathalpha}{\mst@font@tbu}{`M}
\mst@DeclareMathSymbol{N}{\mathalpha}{\mst@font@tbu}{`N}
\mst@DeclareMathSymbol{O}{\mathalpha}{\mst@font@tbu}{`O}
\mst@DeclareMathSymbol{P}{\mathalpha}{\mst@font@tbu}{`P}
\mst@DeclareMathSymbol{Q}{\mathalpha}{\mst@font@tbu}{`Q}
\mst@DeclareMathSymbol{R}{\mathalpha}{\mst@font@tbu}{`R}
\mst@DeclareMathSymbol{S}{\mathalpha}{\mst@font@tbu}{`S}
\mst@DeclareMathSymbol{T}{\mathalpha}{\mst@font@tbu}{`T}
\mst@DeclareMathSymbol{U}{\mathalpha}{\mst@font@tbu}{`U}
\mst@DeclareMathSymbol{V}{\mathalpha}{\mst@font@tbu}{`V}
\mst@DeclareMathSymbol{W}{\mathalpha}{\mst@font@tbu}{`W}
```

```
\mst@DeclareMathSymbol{X}{\mathalpha}{\mst@font@tbu}{`X}
\mst@DeclareMathSymbol{Y}{\mathalpha}{\mst@font@tbu}{`Y}
\mst@DeclareMathSymbol{Z}{\mathalpha}{\mst@font@tbu}{`Z}
%%
\ifmst@nodigits\else
\def\mst@font@tbu{mtoperatorfont}
```

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

```
\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
```

When `symboldelimiters` is passed as an option, we use the Symbol font for the printable characters other than letters and digits.

```
\ifmst@symboldelimiters
\def\mst@font@tbu{mtpsymbol}
\mst@endashfalse
\mst@emdashfalse
\else
\def\mst@font@tbu{mtoperatorfont}
\fi
```

`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`.

```
\ifmst@noexclam\else\typeout{** \string! and \string?}
\DeclareMathSymbol{!}{\mathclose}{\mst@font@tbu}{"21}
\DeclareMathSymbol{\mst@varfam@exclam}{\mathalpha}{\mst@font@tbu}{"21}
\expandafter\mst@addtodo@nonletters\string!\mathclose\mst@varfam@exclam
\DeclareMathSymbol{?}{\mathclose}{\mst@font@tbu}{"3F}
\DeclareMathSymbol{\mst@varfam@question}{\mathalpha}{\mst@font@tbu}{"3F}
\expandafter\mst@addtodo@nonletters\string?\mathclose\mst@varfam@question
\fi
```

`\MTlowerast` `\mst@doasterisk` `\mst@@doasterisk`

`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.

`1.12e` Somehow there was a big omission in `1.12d`, the command `\MTlowerast` as described in the manual was missing!

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

```
\def\mst@@doasterisk{\let\ast\mst@ast\mst@mathactivate*{}\mst@ast}
\ifmst@asterisk\typeout{** asterisk: \string\ast\space and *}
  \ifmst@symbolmisc
    \def\mst@bin@ast{%
    \mathbin{\mathchoice{\raisebox{-.1\height}
                          {\the\textfont\symmtpsymbol\char42}}
                        {\raisebox{-.1\height}
                          {\the\textfont\symmtpsymbol\char42}}
                        {\raisebox{-.1\height}
                          {\the\scriptfont\symmtpsymbol\char42}}
                        {\raisebox{-.1\height}
                          {\the\scriptscriptfont\symmtpsymbol\char42}}}}
  \else
    \def\mst@bin@ast{%
    \mathbin{\mathchoice{\raisebox{-\mst@lowerast}
                          {\the\textfont\symmtoperatorfont\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\textfont\symmtoperatorfont\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\scriptfont\symmtoperatorfont\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\scriptscriptfont\symmtoperatorfont\char42}}}}
  \fi
  \def\mst@varfam@ast{\ifnum\fam=\m@ne\mst@bin@ast\else
    \mathbin{\mathchoice{\raisebox{-\mst@lowerast}
                          {\the\textfont\fam\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\textfont\fam\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\scriptfont\fam\char42}}
                        {\raisebox{-\mst@lowerast}
                          {\the\scriptscriptfont\fam\char42}}}\fi}
  \let\mst@doasterisk\mst@@doasterisk
  \DeclareRobustCommand*{\mst@ast}{\mst@bin@ast}
  \newcommand*\MTlowerast[1]{\def\mst@lowerast{#1}}
  \MTlowerast{.3\height}
  \mst@do@easynonletters\expandafter{\the\mst@do@easynonletters
     \expandafter\def\csname mst@ast \endcsname{\mst@varfam@ast}}
\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 `mathdots` package and accept that we can not avoid the default fonts in that case. So here I just treat `.` (in the hope to really lessen

by 1 the number of fonts embedded at the end in the PDF).

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

```
\ifmst@nopunct\else\typeout{** punctuation\string: \string, \string. \string: \string; and
\DeclareMathSymbol{,}{\mathpunct}{\mst@font@tbu}{"2C}
\DeclareMathSymbol{\mst@varfam@comma}{\mathalpha}{\mst@font@tbu}{"2C}
\expandafter\mst@addtodo@nonletters\string,\mathpunct\mst@varfam@comma
\DeclareMathSymbol{.}{\mathord}{\mst@font@tbu}{"2E}
\DeclareMathSymbol{\mst@varfam@dot}{\mathalpha}{\mst@font@tbu}{"2E}
\mst@addtodo@easynonletters\.\mst@varfam@dot
\DeclareMathSymbol{:}{\mathrel}{\mst@font@tbu}{"3A} % relation spacing
\DeclareMathSymbol{\mst@varfam@colon}{\mathalpha}{\mst@font@tbu}{"3A}
\expandafter\mst@addtodo@nonletters\string:\mathrel\mst@varfam@colon
\@ifpackageloaded{amsmath}
{} % \colon defined in amsmath.sty in terms of : with some enlarged explicit
   % spacing. No need to intervene.
{% no amsmath, use standard punctuation spacing
\let\colon\undefined
  % the reason is if some package has redefined \colon which then
  % can not be used in \cs{DeclareMathSymbol} anymore (we
  % shamelessly overwrite...)
    \DeclareMathSymbol{\colon}{\mathpunct}{\mst@font@tbu}{"3A}
    \mst@do@nonletters\expandafter{\the\mst@do@nonletters
        \def\colon{\mathpunct{\mst@varfam@colon}}}}
\DeclareMathSymbol{;}{\mathpunct}{\mst@font@tbu}{"3B}
\DeclareMathSymbol{\mst@varfam@pointvirgule}{\mathalpha}{\mst@font@tbu}{"3B}
\expandafter\mst@addtodo@nonletters\string;\mathpunct\mst@varfam@pointvirgule
\fi
```

`\relbar` Due to the way = and - are used by LaTeX in arrows, we will have to redefine `\Relbar` and `\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 XeTeX, not for LuaLatex. Also I do my modifications to `\relbar` in a manner testing for the presence of amsmath.

```
\ifmst@nominus\else
\ifmst@XeTeX
\XeTeXmathcharnumdef\mst@minus@sign=\XeTeXmathcodenum`\-\relax
\else
\ifmst@LuaTeX
%% \luatexUmathcharnumdef\mst@minus@sign=\luatexUmathcodenum`\-\relax ne
%% marche pas pour le moment car \luatexUmathcodenum retourne parfois un
%% legacy mathcode, parfois un Unicode mathcode, \'a mon avis c'est un bug
%% de LuaLatex. En fait Taco H me dit que la prochaine version
%% r\'eglera ce probl\`eme. \luatexUmathchardef\mst@minus@sign="2 "2 "0\relax
\mathchardef\mst@minus@sign=8704\relax % "2200
\else
\mathchardef\mst@minus@sign=\mathcode`\-\relax
\fi\fi
\@ifpackageloaded{amsmath}
```

```
{\def\relbar{\mathrel{\mathpalette\mathsm@sh\mst@minus@sign}}}
{\DeclareRobustCommand\relbar{\mathrel{\smash\mst@minus@sign}}}
\def\rightarrowfill{$\m@th\mathord{\relbar}\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\relbar\mkern-2mu$}\hfill
  \mkern-7mu\mathord\rightarrow$}
\def\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\relbar\mkern-2mu$}\hfill
  \mkern-7mu\mathord{\relbar}$}
\fi
```

endash 2011/01/29, 1.1: Producing this next piece of code was not a piece of cake for a novice like myself!
2011/02/05, 1.11: Compatibility with Unicode (via use of fontspec encodings EU1 and EU2)
2011/02/07, 1.12: Improved dealing of Unicode possibility.
2011/04/02, 1.14b: Corrected some very irresponsible bug in the Unicode part which caused a problem when 10 or more math families have been allocated.
2012/09/24, 1.15: Added AtBeginDocument to circumvent some amsmath problem with unicode engines.

```
\def\do@the@endashstuff{\edef\mst@tmp@enc{\encodingdefault}
  \ifx\mst@tmp@enc\mst@eui % xetex+fontspec
  \AtBeginDocument{%
    \XeTeXmathcode`\-="2 \symmtoperatorfont "2013\relax
    \XeTeXmathchardef\mst@varfam@minus="7 \symmtoperatorfont "2013\relax
  }
\else
  \ifx\mst@tmp@enc\mst@euii % luatex+fontspec
  \AtBeginDocument{%
    \luatexUmathcode`\-="2 \symmtoperatorfont "2013\relax
    \luatexUmathchardef\mst@varfam@minus="7 \symmtoperatorfont "2013\relax
  }
\else
  \ifx\mst@tmp@enc\mst@ti % T1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{21}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{21}
\else
  \ifx\mst@tmp@enc\mst@oti % OT1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{123}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{123}
\else
  \ifx\mst@tmp@enc\mst@lyi % LY1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{150}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{150}
\else % make sure we have neither xetex nor luatex
   \ifmst@XeTeX\mst@aborttrue\else\ifmst@LuaTeX\mst@aborttrue\else
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}
{\expandafter\the\expandafter\csname\mst@tmp@enc\string\textendash\endcsname}
   \fi\fi
\fi\fi\fi\fi\fi}
```

```
\def\do@the@emdashstuff{\edef\mst@tmp@enc{\encodingdefault}
\ifx\mst@tmp@enc\mst@eui % xetex+fontspec
    \AtBeginDocument{%
      \XeTeXmathcode`\-="2 \symmtoperatorfont "2014\relax
      \XeTeXmathchardef\mst@varfam@minus="7 \symmtoperatorfont "2014\relax
    }
\else
\ifx\mst@tmp@enc\mst@euii % luatex+fontspec
    \AtBeginDocument{%
      \luatexUmathcode`\-="2 \symmtoperatorfont "2014\relax
      \luatexUmathchardef\mst@varfam@minus="7 \symmtoperatorfont "2014\relax
    }
\else
  \ifx\mst@tmp@enc\mst@ti % T1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{22}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{22}
\else
\ifx\mst@tmp@enc\mst@oti % OT1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{124}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{124}
\else
\ifx\mst@tmp@enc\mst@lyi % LY1
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}{151}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}{151}
\else % make sure we have neither xetex nor luatex
   \ifmst@XeTeX\mst@aborttrue\else\ifmst@LuaTeX\mst@aborttrue\else
   \DeclareMathSymbol{-}{\mathbin}{mtoperatorfont}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{mtoperatorfont}
{\expandafter\the\expandafter\csname\mst@tmp@enc\string\textemdash\endcsname}
   \fi\fi
\fi\fi\fi\fi\fi}
\ifmst@nominus\else\typeout{** minus as endash}
\mst@abortfalse
\ifmst@endash\do@the@endashstuff\else
\ifmst@emdash\do@the@emdashstuff\else
   \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{"2D}
   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{\mst@font@tbu}{"2D}
\fi\fi
\ifmst@abort
    \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{"2D}
    \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{\mst@font@tbu}{"2D}
\fi
\expandafter\mst@addtodo@nonletters\string-\mathbin\mst@varfam@minus
\fi
```

\hbar
\mst@ltbar

2011/01/31, 1.1 I decide to settle the question of the \hbar. First, the LaTeX definition

```
%%\def\hbar{{\mathchar'26\mkern-9muh}} % (original definition from latex.ltx)
```

Well, the fact is that there is a \DeclareMathSymbol in amsfonts.sty, so I can not always rely on the original which had the advantage that at least h would be in the correct font. But of

course not the macron character (\=, \bar). And there is also the issue of the kern whose length is given in a way which depends on cmsy (18mu=1em and em taken from info in cmsy). The first problem is that I don't know how to get the slot position of the macron, given the encoding. So I found another way. I will need an 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 rlap. (as an aside, I am only now (April 2, 2011) aware that the package mathtools provides the mathrlap etc...)

```
%
\def\mst@mathrlap{\mathpalette\mst@mathrlapinternal}
\def\mst@mathrlapinternal#1#2{\rlap{$\mathsurround=0pt#1{#2}$}}
\def\do@the@hbarstuff{\edef\mst@tmp@enc{\encodingdefault}
\ifx\mst@tmp@enc\mst@eui % xetex and unicode font
  \XeTeXmathchardef\hbar="7 \symmtletterfont "0127 \relax
\else
\ifx\mst@tmp@enc\mst@euii % luatex and unicode font
   \luatexUmathchardef\hbar="7 \symmtletterfont "0127 \relax %% or 210F?
\else % I must leave open the T1, OT1 possibility also for XeTeX or LuaTeX
    \ifx\mst@ti\mst@tmp@enc
    \DeclareMathAccent{\mst@ltbar}{\mathalpha}{mtletterfont}{9}
       \else  %% assume OT1. Bad idea? Should not provoke an error anyhow
    \DeclareMathAccent{\mst@ltbar}{\mathalpha}{mtletterfont}{22}
    \fi
    \def\hbar{\mst@mathrlap{\mst@ltbar{\ }}h}
\fi\fi}
\ifmst@nohbar\else\typeout{** \string\hbar}\do@the@hbarstuff\fi
```

As h is from mtletterfont, the accent \mst@ltbar is the \bar accent from that same font. Of course, if the user defines math versions with other encodings than the default one when loading the package this will probably not work there (if I knew how to do for accents what I did for the endash I could do it for all encodings. Surely easy for a TEXpert.) Not to mention if he/she changes the letter shape... one never should give so much freedom to users ;-) Well this construction gives an acceptable result for some of the fonts I have tested, whether upright or in italics.

1.15d: Oct 13, 2012. The \mathcode thing with = is (belatedly, sorry!) made Unicode compatible.

+,=,\Relbar

```
\ifmst@noplus\else\typeout{** \string+ and \string=}
\DeclareMathSymbol{+}{\mathbin}{\mst@font@tbu}{"2B}
\DeclareMathSymbol{\mst@varfam@plus}{\mathalpha}{\mst@font@tbu}{"2B}
\expandafter\mst@addtodo@nonletters\string+\mathbin\mst@varfam@plus
\fi
\ifmst@noequal\else
\ifmst@XeTeX
\XeTeXmathcharnumdef\mst@equal@sign=\XeTeXmathcodenum`\=\relax
\else
\ifmst@LuaTeX
%% 12 octobre 2012 \luatexUmathcodenum`\=
%% ne marche pas pour le moment \`a cause du fait que \luatexUmathcodenum
```

```
%% donne un mathcode sur 8bit ou un Unicode mathcode sans qu'on puisse
%% savoir lequel
%% \luatexUmathchardef\mst@equal@sign="3 "0 "3D\relax
\mathchardef\mst@equal@sign=12349\relax
\else
\mathchardef\mst@equal@sign=\mathcode`\=\relax
\fi\fi
\@ifpackageloaded{amsmath}
{\def\Relbar{\mathrel\mst@equal@sign}}
{\DeclareRobustCommand\Relbar{\mathrel{\mst@equal@sign}}}
\DeclareMathSymbol{=}{\mathrel}{\mst@font@tbu}{"3D}
\DeclareMathSymbol{\mst@varfam@equal}{\mathalpha}{\mst@font@tbu}{"3D}
```

`\nfss@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!}
% $x \csname foox\endcsname$
%
```

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 `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

```
\typeout{** adding \string= \string; and \string+ to \string\nfss@catcodes}
\g@addto@macro\nfss@catcodes{%
  \@makeother\=%
  \@makeother\;%
  \@makeother\+%
}
\expandafter\mst@addtodo@nonletters\string=\mathrel\mst@varfam@equal
\fi
```

`noparenthesis (,),[,],/` `\lbrack` and `\rbrack` are defined in `latex.ltx` by `\def\lbrack{[}\def\rbrack{]}` so this fits well with what we do here. `\lparen` and `\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 `\lbrace` and `\rbrace`.

```
\ifmst@noparen\else\typeout{** parentheses \string( \string) \string[ \string] and slash \s
\ifmst@nosmalldelims
```

```
    \DeclareMathSymbol{(}{\mathopen}{\mst@font@tbu}{"28}
    \DeclareMathSymbol{)}{\mathclose}{\mst@font@tbu}{"29}
    \DeclareMathSymbol{[}{\mathopen} {\mst@font@tbu}{"5B}
    \DeclareMathSymbol{]}{\mathclose}{\mst@font@tbu}{"5D}
    \DeclareMathSymbol{/}{\mathord}{\mst@font@tbu}{"2F}
\else
    \DeclareMathDelimiter{(}{\mathopen}{\mst@font@tbu}{"28}{largesymbols}{"00}
    \DeclareMathDelimiter{)}{\mathclose}{\mst@font@tbu}{"29}{largesymbols}{"01}
    \DeclareMathDelimiter{[}{\mathopen} {\mst@font@tbu}{"5B}{largesymbols}{"02}
    \DeclareMathDelimiter{]}{\mathclose}{\mst@font@tbu}{"5D}{largesymbols}{"03}
    \DeclareMathDelimiter{/}{\mathord}{\mst@font@tbu}{"2F}{largesymbols}{"0E}
\fi
\DeclareMathSymbol{\mst@varfam@lparen}{\mathalpha}{\mst@font@tbu}{40}
\DeclareMathSymbol{\mst@varfam@rparen}{\mathalpha}{\mst@font@tbu}{41}
\DeclareMathSymbol{\mst@varfam@lbrack}{\mathalpha}{\mst@font@tbu}{"5B}
\DeclareMathSymbol{\mst@varfam@rbrack}{\mathalpha}{\mst@font@tbu}{"5D}
\DeclareMathSymbol{\mst@varfam@slash}{\mathalpha}{\mst@font@tbu}{"2F}
\expandafter\mst@addtodo@nonletters\string(\mathopen\mst@varfam@lparen
\expandafter\mst@addtodo@nonletters\string)\mathclose\mst@varfam@rparen
\expandafter\mst@addtodo@nonletters\string[\mathopen\mst@varfam@lbrack
\expandafter\mst@addtodo@nonletters\string]\mathclose\mst@varfam@rbrack
\mst@addtodo@easynonletters\/\mst@varfam@slash
\fi
```

alldelims <,>,\setminus \backslash

```
\ifmst@alldelims\typeout{** alldelims: \string< \string>
  \string\backslash\space\string\setminus\space|
  \string\vert\space\string\mid\space\string\{\space and \string\}}
\ifmst@nosmalldelims
```

Dec 18, 2012. We then want `\let\backslash\mst@varfam@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`.

```
    \DeclareMathDelimiter{\mst@varfam@backslash}
        {\mathalpha}{symbols}{"6E}{largesymbols}{"0F}
\else
    \DeclareMathDelimiter{<}{\mathopen}{\mst@font@tbu}{"3C}{largesymbols}{"0A}
    \DeclareMathDelimiter{>}{\mathclose}{\mst@font@tbu}{"3E}{largesymbols}{"0B}
```

There is no backslash in the Symbol font hence `mtoperatorfont` here.

```
    \expandafter\DeclareMathDelimiter\@backslashchar
        {\mathord}{mtoperatorfont}{"5C}{largesymbols}{"0F}
    \DeclareMathDelimiter{\backslash}
        {\mathord}{mtoperatorfont}{"5C}{largesymbols}{"0F}
    \DeclareMathDelimiter{\mst@varfam@backslash}
        {\mathalpha}{mtoperatorfont}{"5C}{largesymbols}{"0F}
\fi
\DeclareMathSymbol{<}{\mathrel}{\mst@font@tbu}{"3C}
```

```
\DeclareMathSymbol{>}{\mathrel}{\mst@font@tbu}{"3E}
\DeclareMathSymbol{\mst@varfam@less}{\mathalpha}{\mst@font@tbu}{"3C}
\DeclareMathSymbol{\mst@varfam@more}{\mathalpha}{\mst@font@tbu}{"3E}
\expandafter\mst@addtodo@nonletters\string<\mathrel\mst@varfam@less
\expandafter\mst@addtodo@nonletters\string>\mathrel\mst@varfam@more
\mst@do@easynonletters\expandafter{\the\mst@do@easynonletters
            \let\backslash\mst@varfam@backslash}
\DeclareMathSymbol{\setminus}{\mathbin}{mtoperatorfont}{"5C}
\DeclareMathSymbol{\mst@varfam@setminus}{\mathalpha}{mtoperatorfont}{"5C}
\mst@do@nonletters\expandafter{\the\mst@do@nonletters
            \def\setminus{\mathbin{\mst@varfam@setminus}}}
```

`\models` 1.15d: 13 oct 2012. Before modifying | we must preserve `\models`.

```
\ifmst@XeTeX
\XeTeXmathcharnumdef\mst@vert@bar=\XeTeXmathcodenum`\|\relax
\else
\ifmst@LuaTeX
%% \luatexUmathchardef\mst@vert@bar="0 "2 "006A\relax
\mathchardef\mst@vert@bar=618\relax
\else\mathchardef\mst@vert@bar=\mathcode`\|\relax
\fi\fi
\DeclareRobustCommand\models{\mathrel{\mst@vert@bar}\joinrel\Relbar}
```

`|,\mid,\vert` (2011) I did not do anything then to try to emulate `\Vert` with the vertical bar from the text font... and now (2012) **mathastext** is not as radical as it used to be anyhow, so it is too late.

```
\ifmst@nosmalldelims
    \DeclareMathSymbol{|}{\mathord}{\mst@font@tbu}{124}
\else
    \DeclareMathDelimiter{|}{\mathord}{\mst@font@tbu}{124}{largesymbols}{"0C}
\fi
\def\vert{|}
\DeclareMathSymbol{\mst@varfam@vbar}{\mathalpha}{\mst@font@tbu}{124}
\mst@addtodo@easynonletters\|\mst@varfam@vbar
\DeclareMathSymbol{\mid}{\mathrel}{\mst@font@tbu}{124}
\mst@do@nonletters\expandafter{\the\mst@do@nonletters
    \def\mid{\mathrel\mst@varfam@vbar}}
```

`\MTexplicitbraces-obeymathxx` `\MTexplicitbraces-donotobeymathxx` Braces. With version 1.2, `\{` and `\}` will not be acceptable as delimiters anymore if the redefinitions below in `\mst@dobraces` are enacted. But they will obey math alphabets. Improvements in 1.2a, to preserve robustness.

```
\ifmst@nosmalldelims
   \DeclareMathSymbol{\lbrace}{\mathopen}{\mst@font@tbu}{123}
   \DeclareMathSymbol{\rbrace}{\mathclose}{\mst@font@tbu}{125}
\else
   \DeclareMathDelimiter{\lbrace}
        {\mathopen}{\mst@font@tbu}{123}{largesymbols}{"08}
   \DeclareMathDelimiter{\rbrace}
        {\mathclose}{\mst@font@tbu}{125}{largesymbols}{"09}
\fi
```

```
\DeclareMathSymbol{\mst@varfam@lbrace}{\mathalpha}{\mst@font@tbu}{123}
\DeclareMathSymbol{\mst@varfam@rbrace}{\mathalpha}{\mst@font@tbu}{125}
\DeclareRobustCommand*{\mst@lbrace}
        {\ifmmode\mathopen\mst@varfam@lbrace\else\textbraceleft\fi}
\DeclareRobustCommand*{\mst@rbrace}
        {\ifmmode\mathclose\mst@varfam@rbrace\else\textbraceright\fi}
\mst@do@nonletters\expandafter{\the\mst@do@nonletters
    \mst@dobraces{\let\{\mst@lbrace\let\}\mst@rbrace}}
\fi % end of \ifmst@alldelims
\newcommand*{\MTexplicitbracesobeymathxx}{\let\mst@dobraces\@firstofone}
\newcommand*{\MTexplicitbracesdonotobeymathxx}{\let\mst@dobraces\@gobble}
\MTexplicitbracesdonotobeymathxx
```

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 `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 `latex.ltx` the same definitions as in `plain.tex`: `\chardef\%='\%`, `\chardef\&='\&`, and `\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 *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 `\MTeasynonlettersdonotobeymathxx`. Besides, I have absolutely no idea why I had different looking code depending on the engine XeTeX, LuaTeX or default. Removed.

```
\ifmst@nospecials\else\typeout{** \string\#\space\string\mathdollar\space\string\%\space\st
\count255=\symmtoperatorfont
\multiply\count255 by 256
\advance\count255 by 35
    \mathcode`\#\count255
\advance\count255 by 1
    \mathchardef\mathdollar\count255
\advance\count255 by 1
    \mathcode`\%\count255
\advance\count255 by 1
    \mathcode`\&\count255
\count255=\symmtoperatorfont
\multiply\count255 by 256
\advance\count255 by 28707 % = "7023
    \mathchardef\mst@varfam@mathhash\count255
\advance\count255 by 1
    \mathchardef\mst@varfam@mathdollar\count255
\advance\count255 by 1
```

```
    \mathchardef\mst@varfam@mathpercent\count255
\advance\count255 by 1
    \mathchardef\mst@varfam@mathampersand\count255
\mst@do@easynonletters\expandafter{\the\mst@do@easynonletters
    \mathcode`\#=\mst@varfam@mathhash
    \let\mathdollar\mst@varfam@mathdollar
    \mathcode`\%=\mst@varfam@mathpercent
    \mathcode`\&=\mst@varfam@mathampersand}
\fi
```

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: `$\displaystyle\prod_1^2$` 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.

Note oct 2012: maybe I should re-examine what I did?

```
\ifmst@symbolmisc\typeout{** symbolmisc: miscellaneous math symbols from Symbol font}
\let\mst@prod\prod
\let\defaultprod\prod
\DeclareMathSymbol{\prodpsy}{\mathop}{mtpsymbol}{213}
\renewcommand*{\prod}{\ifinner\prodpsy\else\mst@prod\fi}
\let\mst@sum\sum
\let\defaultsum\sum
\DeclareMathSymbol{\sumpsy}{\mathop}{mtpsymbol}{229}
\renewcommand*{\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}
\DeclareMathSymbol{\pm}{\mathbin}{mtpsymbol}{177}
\DeclareMathSymbol{\times}{\mathbin}{mtpsymbol}{180}
\DeclareMathSymbol{\propto}{\mathrel}{mtpsymbol}{181}
\DeclareMathSymbol{\mid}{\mathrel}{mtpsymbol}{124}
\DeclareMathSymbol{\leq}{\mathrel}{mtpsymbol}{163}
\DeclareMathSymbol{\geq}{\mathrel}{mtpsymbol}{179}
\DeclareMathSymbol{\approx}{\mathrel}{mtpsymbol}{187}
\DeclareMathSymbol{\supset}{\mathrel}{mtpsymbol}{201}
\DeclareMathSymbol{\subset}{\mathrel}{mtpsymbol}{204}
\DeclareMathSymbol{\supseteq}{\mathrel}{mtpsymbol}{202}
\DeclareMathSymbol{\subseteq}{\mathrel}{mtpsymbol}{205}
\DeclareMathSymbol{\in}{\mathrel}{mtpsymbol}{206}
\DeclareMathSymbol{\sim}{\mathrel}{mtpsymbol}{126}
\let\cong\undefined
\DeclareMathSymbol{\cong}{\mathrel}{mtpsymbol}{64}
\DeclareMathSymbol{\perp}{\mathrel}{mtpsymbol}{94}
\DeclareMathSymbol{\equiv}{\mathrel}{mtpsymbol}{186}
\let\notin\undefined
\DeclareMathSymbol{\notin}{\mathrel}{mtpsymbol}{207}
\DeclareMathDelimiter{\rangle}
   {\mathclose}{mtpsymbol}{241}{largesymbols}{"0B}
\DeclareMathDelimiter{\langle}
   {\mathopen}{mtpsymbol}{225}{largesymbols}{"0A}
\fi
```

`symbolre` I like the `\Re` and `\Im` from Symbol, so I overwrite the CM ones.

```
\ifmst@symbolre\typeout{** symbolre: \string\Re\space and \string\Im\space from Symbol font}
\DeclareMathSymbol{\Re}{\mathord}{mtpsymbol}{"C2}
\DeclareMathSymbol{\Im}{\mathord}{mtpsymbol}{"C1}
\DeclareMathSymbol{\DotTriangle}{\mathord}{mtpsymbol}{92}
\fi
```

Greek letters

LGRgreek > selfGreek > eulergreek > symbolgreek

1.11 I correct some bugs on how eulergreek and symbolgreek interacted.

1.12b I introduced another very stupid bug (regarding uppercase Greek) in 1.1. This is corrected here. All this business of checking for \digamma was truly stupid as some package might well have defined the font for Greek without having defined \digamma, I don't know what crossed my dumb mind when I committed this insane code and started redefining uppercase Greek, and it is even worse than that as I was redefining not only relatively benign things such as \Alpha but also \Phi etc...

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.

* version 1.12b did not define the \omicron in the case when no Greek-related option was passed to the package. This was a bug.

```
\let\mst@mathord\mathalpha
\mst@goaheadtrue
\ifmst@selfGreek
    \def\mst@font@tbu{mtselfGreekfont}
 \else
\ifmst@eulergreek
    \def\mst@font@tbu{mteulervm}
 \else
\ifmst@symbolgreek
    \def\mst@font@tbu{mtpsymbol}
    \let\mst@mathord\mathord
 \else
\ifmst@LGRgreek
     \mst@goaheadfalse
 \else
```

The \omicron requires special treatment. By default we use the o from the (original) normal alphabet, if eulergreek 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.

```
       \mst@goaheadfalse
       \ifx\omicron\undefined
       \newcommand*{\omicron}{\mst@alph@omicron{o}}
       \fi
\fi\fi\fi\fi
\ifmst@goahead
\DeclareMathSymbol{\Alpha}{\mst@mathord}{\mst@font@tbu}{"41}
\DeclareMathSymbol{\Beta}{\mst@mathord}{\mst@font@tbu}{"42}
\DeclareMathSymbol{\Epsilon}{\mst@mathord}{\mst@font@tbu}{"45}
\DeclareMathSymbol{\Zeta}{\mst@mathord}{\mst@font@tbu}{"5A}
```

```
\DeclareMathSymbol{\Eta}{\mst@mathord}{\mst@font@tbu}{"48}
\DeclareMathSymbol{\Iota}{\mst@mathord}{\mst@font@tbu}{"49}
\DeclareMathSymbol{\Kappa}{\mst@mathord}{\mst@font@tbu}{"4B}
\DeclareMathSymbol{\Mu}{\mst@mathord}{\mst@font@tbu}{"4D}
\DeclareMathSymbol{\Nu}{\mst@mathord}{\mst@font@tbu}{"4E}
\DeclareMathSymbol{\Omicron}{\mst@mathord}{\mst@font@tbu}{"4F}
\DeclareMathSymbol{\Rho}{\mst@mathord}{\mst@font@tbu}{"50}
\DeclareMathSymbol{\Tau}{\mst@mathord}{\mst@font@tbu}{"54}
\DeclareMathSymbol{\Chi}{\mst@mathord}{\mst@font@tbu}{"58}
```

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`).

```
\ifx\mst@mathord\mathord
```

symbolgreek but neither eulergreek nor selfGreek

```
%% attention le P de Symbol est un \Pi pas un \Rho
\DeclareMathSymbol{\Rho}{\mathord}{mtpsymbol}{"52}
%% attention le X de Symbol est un \Xi pas un \Chi
\DeclareMathSymbol{\Chi}{\mathord}{mtpsymbol}{"43}
%% attention le F de Symbol est un \Phi. Il n'y a pas de \Digamma
\DeclareMathSymbol{\Gamma}{\mathord}{mtpsymbol}{"47}
\DeclareMathSymbol{\Delta}{\mathord}{mtpsymbol}{"44}
\DeclareMathSymbol{\Theta}{\mathord}{mtpsymbol}{"51}
\DeclareMathSymbol{\Lambda}{\mathord}{mtpsymbol}{"4C}
\DeclareMathSymbol{\Xi}{\mathord}{mtpsymbol}{"58}
\DeclareMathSymbol{\Pi}{\mathord}{mtpsymbol}{"50}
\DeclareMathSymbol{\Sigma}{\mathord}{mtpsymbol}{"53}
\DeclareMathSymbol{\Upsilon}{\mathord}{mtpsymbol}{"A1}
\DeclareMathSymbol{\Phi}{\mathord}{mtpsymbol}{"46}
\DeclareMathSymbol{\Psi}{\mathord}{mtpsymbol}{"59}
\DeclareMathSymbol{\Omega}{\mathord}{mtpsymbol}{"57}
\else
```

not symbolgreek but eulergreek or selfGreek or both

```
\DeclareMathSymbol{\Digamma}{\mathalpha}{\mst@font@tbu}{"46}
\DeclareMathSymbol\Gamma    {\mathalpha}{\mst@font@tbu}{"00}
\DeclareMathSymbol\Delta    {\mathalpha}{\mst@font@tbu}{"01}
\DeclareMathSymbol\Theta    {\mathalpha}{\mst@font@tbu}{"02}
\DeclareMathSymbol\Lambda   {\mathalpha}{\mst@font@tbu}{"03}
\DeclareMathSymbol\Xi       {\mathalpha}{\mst@font@tbu}{"04}
\DeclareMathSymbol\Pi       {\mathalpha}{\mst@font@tbu}{"05}
\DeclareMathSymbol\Sigma    {\mathalpha}{\mst@font@tbu}{"06}
\DeclareMathSymbol\Upsilon  {\mathalpha}{\mst@font@tbu}{"07}
\DeclareMathSymbol\Phi      {\mathalpha}{\mst@font@tbu}{"08}
\DeclareMathSymbol\Psi      {\mathalpha}{\mst@font@tbu}{"09}
\DeclareMathSymbol\Omega    {\mathalpha}{\mst@font@tbu}{"0A}
\fi
\fi
```

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 LaTeX are of type `mathord`. If we use the Euler font it is perhaps better to have them be of type `mathalpha`

```
\ifmst@goahead
\ifmst@eulergreek
\DeclareMathSymbol{\alpha}  {\mathalpha}{mteulervm}{"0B}
\DeclareMathSymbol{\beta}   {\mathalpha}{mteulervm}{"0C}
\DeclareMathSymbol{\gamma}  {\mathalpha}{mteulervm}{"0D}
\DeclareMathSymbol{\delta}  {\mathalpha}{mteulervm}{"0E}
\DeclareMathSymbol{\epsilon}{\mathalpha}{mteulervm}{"0F}
\DeclareMathSymbol{\zeta}   {\mathalpha}{mteulervm}{"10}
\DeclareMathSymbol{\eta}    {\mathalpha}{mteulervm}{"11}
\DeclareMathSymbol{\theta}  {\mathalpha}{mteulervm}{"12}
\DeclareMathSymbol{\iota}   {\mathalpha}{mteulervm}{"13}
\DeclareMathSymbol{\kappa}  {\mathalpha}{mteulervm}{"14}
\DeclareMathSymbol{\lambda} {\mathalpha}{mteulervm}{"15}
\DeclareMathSymbol{\mu}     {\mathalpha}{mteulervm}{"16}
\DeclareMathSymbol{\nu}     {\mathalpha}{mteulervm}{"17}
\DeclareMathSymbol{\xi}     {\mathalpha}{mteulervm}{"18}
\let\omicron\undefined
\DeclareMathSymbol{\omicron}{\mathalpha}{mteulervm}{"6F}
\DeclareMathSymbol{\pi}     {\mathalpha}{mteulervm}{"19}
\DeclareMathSymbol{\rho}    {\mathalpha}{mteulervm}{"1A}
\DeclareMathSymbol{\sigma}  {\mathalpha}{mteulervm}{"1B}
\DeclareMathSymbol{\tau}    {\mathalpha}{mteulervm}{"1C}
\DeclareMathSymbol{\upsilon}{\mathalpha}{mteulervm}{"1D}
\DeclareMathSymbol{\phi}    {\mathalpha}{mteulervm}{"1E}
\DeclareMathSymbol{\chi}    {\mathalpha}{mteulervm}{"1F}
\DeclareMathSymbol{\psi}    {\mathalpha}{mteulervm}{"20}
\DeclareMathSymbol{\omega}  {\mathalpha}{mteulervm}{"21}
\DeclareMathSymbol{\varepsilon}{\mathalpha}{mteulervm}{"22}
\DeclareMathSymbol{\vartheta}{\mathalpha}{mteulervm}{"23}
\DeclareMathSymbol{\varpi}  {\mathalpha}{mteulervm}{"24}
\let\varrho=\rho
\let\varsigma=\sigma
\DeclareMathSymbol{\varphi} {\mathalpha}{mteulervm}{"27}
\DeclareMathSymbol{\partial}{\mathalpha}{mteulervm}{"40}
\DeclareMathSymbol{\wp}{\mathalpha}{mteulervm}{"7D}
\DeclareMathSymbol{\ell}{\mathalpha}{mteulervm}{"60}
\else
\ifmst@symbolgreek
\DeclareMathSymbol{\alpha}{\mathord}{mtpsymbol}{"61}
\DeclareMathSymbol{\beta}{\mathord}{mtpsymbol}{"62}
\DeclareMathSymbol{\gamma}{\mathord}{mtpsymbol}{"67}
\DeclareMathSymbol{\delta}{\mathord}{mtpsymbol}{"64}
\DeclareMathSymbol{\epsilon}{\mathord}{mtpsymbol}{"65}
\DeclareMathSymbol{\zeta}{\mathord}{mtpsymbol}{"7A}
\DeclareMathSymbol{\eta}{\mathord}{mtpsymbol}{"68}
\DeclareMathSymbol{\theta}{\mathord}{mtpsymbol}{"71}
\DeclareMathSymbol{\iota}{\mathord}{mtpsymbol}{"69}
```

```
\DeclareMathSymbol{\kappa}{\mathord}{mtpsymbol}{"6B}
\DeclareMathSymbol{\lambda}{\mathord}{mtpsymbol}{"6C}
\DeclareMathSymbol{\mu}{\mathord}{mtpsymbol}{"6D}
\DeclareMathSymbol{\nu}{\mathord}{mtpsymbol}{"6E}
\DeclareMathSymbol{\xi}{\mathord}{mtpsymbol}{"78}
\let\omicron\undefined
\DeclareMathSymbol{\omicron}{\mathord}{mtpsymbol}{"6F}
\DeclareMathSymbol{\pi}{\mathord}{mtpsymbol}{"70}
\DeclareMathSymbol{\rho}{\mathord}{mtpsymbol}{"72}
\DeclareMathSymbol{\sigma}{\mathord}{mtpsymbol}{"73}
\DeclareMathSymbol{\tau}{\mathord}{mtpsymbol}{"74}
\DeclareMathSymbol{\upsilon}{\mathord}{mtpsymbol}{"75}
\DeclareMathSymbol{\phi}{\mathord}{mtpsymbol}{"66}
\DeclareMathSymbol{\chi}{\mathord}{mtpsymbol}{"63}
\DeclareMathSymbol{\psi}{\mathord}{mtpsymbol}{"79}
\DeclareMathSymbol{\omega}{\mathord}{mtpsymbol}{"77}
\let\varepsilon=\epsilon
\DeclareMathSymbol{\vartheta}{\mathord}{mtpsymbol}{"4A}
\DeclareMathSymbol{\varpi}{\mathord}{mtpsymbol}{"76}
\let\varrho=\rho
\DeclareMathSymbol{\varsigma}{\mathord}{mtpsymbol}{"56}
\DeclareMathSymbol{\varphi}{\mathord}{mtpsymbol}{"6A}
\DeclareMathSymbol{\partial}{\mathord}{mtpsymbol}{"B6}
\DeclareMathSymbol{\wp}{\mathord}{mtpsymbol}{"C3}
\fi\fi\fi
%%
\ifmst@LGRgreek % internal names for subdued implementation (1.15c-1.15e)
\let\mst@origAlpha\Alpha
\let\mst@origBeta\Beta
\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@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@origvarsigma\varsigma
\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@origDigamma\Digamma
\let\mst@origdigamma\digamma
%%
\def\mst@restoregreek{%
\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
%
\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\varsigma\mst@origvarsigma
\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
\let\Digamma\mst@origDigamma
\let\digamma\mst@origdigamma
}
%%
\def\mst@setuplgrgreek{%
\let\Alpha\mst@Alpha
\let\Beta\mst@Beta
\let\Gamma\mst@Gamma
\let\Delta\mst@Delta
\let\Epsilon\mst@Epsilon
\let\Zeta\mst@Zeta
\let\Eta\mst@Eta
\let\Theta\mst@Theta
\let\Iota\mst@Iota
\let\Kappa\mst@Kappa
\let\Lambda\mst@Lambda
```

```
\let\Mu\mst@Mu
\let\Nu\mst@Nu
\let\Xi\mst@Xi
\let\Omicron\mst@Omicron
\let\Pi\mst@Pi
\let\Rho\mst@Rho
\let\Sigma\mst@Sigma
\let\Tau\mst@Tau
\let\Upsilon\mst@Upsilon
\let\Phi\mst@Phi
\let\Chi\mst@Chi
\let\Psi\mst@Psi
\let\Omega\mst@Omega
%
\let\alpha\mst@alpha
\let\beta\mst@beta
\let\gamma\mst@gamma
\let\delta\mst@delta
\let\epsilon\mst@epsilon
\let\zeta\mst@zeta
\let\eta\mst@eta
\let\theta\mst@theta
\let\iota\mst@iota
\let\kappa\mst@kappa
\let\lambda\mst@lambda
\let\mu\mst@mu
\let\nu\mst@nu
\let\xi\mst@xi
\let\omicron\mst@omicron
\let\pi\mst@pi
\let\rho\mst@rho
\let\varsigma\mst@varsigma
\let\sigma\mst@sigma
\let\tau\mst@tau
\let\upsilon\mst@upsilon
\let\phi\mst@phi
\let\chi\mst@chi
\let\psi\mst@psi
\let\omega\mst@omega
\let\Digamma\mst@Digamma
\let\digamma\mst@digamma
}
%%
\DeclareMathSymbol{\mst@Alpha}{\mathalpha}{mtlgrfontupper}{65}
\DeclareMathSymbol{\mst@Beta}{\mathalpha}{mtlgrfontupper}{66}
\DeclareMathSymbol{\mst@Gamma}{\mathalpha}{mtlgrfontupper}{71}
\DeclareMathSymbol{\mst@Delta}{\mathalpha}{mtlgrfontupper}{68}
\DeclareMathSymbol{\mst@Epsilon}{\mathalpha}{mtlgrfontupper}{69}
\DeclareMathSymbol{\mst@Zeta}{\mathalpha}{mtlgrfontupper}{90}
```

```
\DeclareMathSymbol{\mst@Eta}{\mathalpha}{mtlgrfontupper}{72}
\DeclareMathSymbol{\mst@Theta}{\mathalpha}{mtlgrfontupper}{74}
\DeclareMathSymbol{\mst@Iota}{\mathalpha}{mtlgrfontupper}{73}
\DeclareMathSymbol{\mst@Kappa}{\mathalpha}{mtlgrfontupper}{75}
\DeclareMathSymbol{\mst@Lambda}{\mathalpha}{mtlgrfontupper}{76}
\DeclareMathSymbol{\mst@Mu}{\mathalpha}{mtlgrfontupper}{77}
\DeclareMathSymbol{\mst@Nu}{\mathalpha}{mtlgrfontupper}{78}
\DeclareMathSymbol{\mst@Xi}{\mathalpha}{mtlgrfontupper}{88}
\DeclareMathSymbol{\mst@Omicron}{\mathalpha}{mtlgrfontupper}{79}
\DeclareMathSymbol{\mst@Pi}{\mathalpha}{mtlgrfontupper}{80}
\DeclareMathSymbol{\mst@Rho}{\mathalpha}{mtlgrfontupper}{82}
\DeclareMathSymbol{\mst@Sigma}{\mathalpha}{mtlgrfontupper}{83}
\DeclareMathSymbol{\mst@Tau}{\mathalpha}{mtlgrfontupper}{84}
\DeclareMathSymbol{\mst@Upsilon}{\mathalpha}{mtlgrfontupper}{85}
\DeclareMathSymbol{\mst@Phi}{\mathalpha}{mtlgrfontupper}{70}
\DeclareMathSymbol{\mst@Chi}{\mathalpha}{mtlgrfontupper}{81}
\DeclareMathSymbol{\mst@Psi}{\mathalpha}{mtlgrfontupper}{89}
\DeclareMathSymbol{\mst@Omega}{\mathalpha}{mtlgrfontupper}{87}
%
\DeclareMathSymbol{\mst@alpha}{\mathalpha}{mtlgrfontlower}{97}
\DeclareMathSymbol{\mst@beta}{\mathalpha}{mtlgrfontlower}{98}
\DeclareMathSymbol{\mst@gamma}{\mathalpha}{mtlgrfontlower}{103}
\DeclareMathSymbol{\mst@delta}{\mathalpha}{mtlgrfontlower}{100}
\DeclareMathSymbol{\mst@epsilon}{\mathalpha}{mtlgrfontlower}{101}
\DeclareMathSymbol{\mst@zeta}{\mathalpha}{mtlgrfontlower}{122}
\DeclareMathSymbol{\mst@eta}{\mathalpha}{mtlgrfontlower}{104}
\DeclareMathSymbol{\mst@theta}{\mathalpha}{mtlgrfontlower}{106}
\DeclareMathSymbol{\mst@iota}{\mathalpha}{mtlgrfontlower}{105}
\DeclareMathSymbol{\mst@kappa}{\mathalpha}{mtlgrfontlower}{107}
\DeclareMathSymbol{\mst@lambda}{\mathalpha}{mtlgrfontlower}{108}
\DeclareMathSymbol{\mst@mu}{\mathalpha}{mtlgrfontlower}{109}
\DeclareMathSymbol{\mst@nu}{\mathalpha}{mtlgrfontlower}{110}
\DeclareMathSymbol{\mst@xi}{\mathalpha}{mtlgrfontlower}{120}
\DeclareMathSymbol{\mst@omicron}{\mathalpha}{mtlgrfontlower}{111}
\DeclareMathSymbol{\mst@pi}{\mathalpha}{mtlgrfontlower}{112}
\DeclareMathSymbol{\mst@rho}{\mathalpha}{mtlgrfontlower}{114}
\DeclareMathSymbol{\mst@varsigma}{\mathalpha}{mtlgrfontlower}{99}
\DeclareMathSymbol{\mst@sigma}{\mathalpha}{mtlgrfontlower}{115}
\DeclareMathSymbol{\mst@tau}{\mathalpha}{mtlgrfontlower}{116}
\DeclareMathSymbol{\mst@upsilon}{\mathalpha}{mtlgrfontlower}{117}
\DeclareMathSymbol{\mst@phi}{\mathalpha}{mtlgrfontlower}{102}
\DeclareMathSymbol{\mst@chi}{\mathalpha}{mtlgrfontlower}{113}
\DeclareMathSymbol{\mst@psi}{\mathalpha}{mtlgrfontlower}{121}
\DeclareMathSymbol{\mst@omega}{\mathalpha}{mtlgrfontlower}{119}
\DeclareMathSymbol{\mst@digamma}{\mathalpha}{mtlgrfontlower}{147}
\DeclareMathSymbol{\mst@Digamma}{\mathalpha}{mtlgrfontlower}{195}
% cf http://milde.users.sourceforge.net/LGR/lgrxenc.def.html
% et greek.ldf du package babel
\ifmst@subdued\else\mst@setuplgrgreek\fi
```

```
\fi
```

`\inodot` `\jnodot` 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.

```
\edef\mst@tmp@enc{\encodingdefault}
\mst@goaheadtrue
\ifx\mst@tmp@enc\mst@eui % xetex and unicode font
  \XeTeXmathchardef\inodot="7 \symmtletterfont "0131 \relax
  \XeTeXmathchardef\jnodot="7 \symmtletterfont "0237 \relax
\else
\ifx\mst@tmp@enc\mst@euii % luatex and unicode font
   \luatexUmathchardef\inodot="7 \symmtletterfont "0131 \relax
   \luatexUmathchardef\jnodot="7 \symmtletterfont "0237 \relax
\else
\ifx\mst@tmp@enc\mst@ti % T1
   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{25}
   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{26}
\else
\ifx\mst@tmp@enc\mst@oti % OT1
   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
\else
\ifx\mst@tmp@enc\mst@lyi % LY1
   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
\else
   \ifmst@XeTeX\mst@goaheadfalse\else\ifmst@LuaTeX\mst@goaheadfalse\else
   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}
{\expandafter\the\expandafter\csname\mst@tmp@enc\string\i\endcsname}
   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}
{\expandafter\the\expandafter\csname\mst@tmp@enc\string\j\endcsname}
   \fi\fi
\fi\fi\fi\fi\fi
\ifmst@defaultimath\else\typeout{** \string\i\space and \string\j\space}
  \ifmst@goahead
    \renewcommand*{\imath}{\inodot}
    \renewcommand*{\jmath}{\jnodot}
    \let\mst@oldi\i \let\mst@oldj\j
    \DeclareRobustCommand*{\i}{\ifmmode\inodot\else\mst@oldi\fi}
    \DeclareRobustCommand*{\j}{\ifmmode\jnodot\else\mst@oldj\fi}
  \fi
\fi
```

`math accents` 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 LaTeX, the mathaccents 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

```
\edef\mst@tmp@enc{\encodingdefault} %% rather one too many than sorry
\ifmst@mathaccents\typeout{** math accents}
\ifx\mst@ti\mst@tmp@enc
\DeclareMathAccent{\acute}{\mathalpha}{mtoperatorfont}{1}
\DeclareMathAccent{\grave}{\mathalpha}{mtoperatorfont}{0}
\DeclareMathAccent{\ddot}{\mathalpha}{mtoperatorfont}{4}
\DeclareMathAccent{\tilde}{\mathalpha}{mtoperatorfont}{3}
\DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{9}
\DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{8}
\DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{7}
\DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{2}
\DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{10}
\DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{6}
\else
\DeclareMathAccent{\acute}{\mathalpha}{mtoperatorfont}{19}
\DeclareMathAccent{\grave}{\mathalpha}{mtoperatorfont}{18}
\DeclareMathAccent{\ddot}{\mathalpha}{mtoperatorfont}{127}
\DeclareMathAccent{\tilde}{\mathalpha}{mtoperatorfont}{126}
\DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{22}
\DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{21}
\DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{20}
\DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{94}
\DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{95}
\DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{23}
  \ifx\mst@lyi\mst@tmp@enc   % LY1 encoding
    \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{5}
  \else
     \ifx\mst@oti\mst@tmp@enc\else
      \typeout{** mathastext: math accents have been assumed to be^^J%
        ** as in OT1 encoding.}
      \fi
   \fi
\fi\fi
```

`Math sizes` I took the code for `\Huge` and `\HUGE` from the `moresize` package of Christian Cornelssen

```
\ifmst@defaultsizes\else
\providecommand\@xxxpt{29.86}
\providecommand\@xxxvipt{35.83}
\ifmst@twelve
  \def\Huge{\@setfontsize\Huge\@xxxpt{36}}
  \def\HUGE{\@setfontsize\HUGE\@xxxvipt{43}}
\typeout{** \protect\Huge\space and \protect\HUGE\space have been (re)-defined.}
\else
  \def\HUGE{\@setfontsize\HUGE\@xxxpt{36}}
\typeout{** \protect\HUGE\space has been (re)-defined.}
\fi
```

I choose rather big subscripts.

```
\def\defaultscriptratio{.8333}
\def\defaultscriptscriptratio{.7}
\DeclareMathSizes{9}{9}{7}{5}
\DeclareMathSizes{\@xpt}{\@xpt}{8}{6}
\DeclareMathSizes{\@xipt}{\@xipt}{9}{7}
\DeclareMathSizes{\@xiipt}{\@xiipt}{10}{8}
\DeclareMathSizes{\@xivpt}{\@xivpt}{\@xiipt}{10}
\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}
\typeout{** mathastext has declared larger sizes for subscripts.^^J%
** To keep LaTeX defaults, use option ‘defaultmathsizes\string’.}
\fi

\endinput
```