

The **mathastext** package

JEAN-FRANÇOIS BURNOL

jfbu (at) free (dot) fr

Package version: v1.2b (2012/12/31)

Documentation generated from the source file

with timestamp "31-12-2012 at 16:37:39 CET"

Abstract

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

What mathastext does	1, p. 2
Examples	1.1, p. 2
Overview	1.2, p. 3
Basic use—sans in math—always load mathastext last—option LGRgreek —avoid OT1 encoding.	
Main options	1.3, p. 5
The italic option—The defaultmathsizes option—The subdued option.	
Math versions	1.4, p. 7
Extended scope of the math alphabets commands	1.5, p. 9
Italic corrections	1.6, p. 10
Additional whitespaces	1.7, p. 13
Greek letters	1.8, p. 14
Shape of Greek letters.	
Unicode engines	1.9, p. 16
Compatibility issues	1.10, p. 19
Package options and commands	2, p. 19
Summary of main options	2.1, p. 19
Miscellaneous	2.2, p. 20
Commands	2.3, p. 22
Preamble-only commands—Commands usable everywhere—Body-only commands.	
Complete list of options	2.4, p. 27
Implementation	3, p. 30

1 What `mathastext` does

1.1 Examples

`mathastext`'s basic aim is to have the same font for text and mathematics. With hundreds of free text fonts packaged for L^AT_EX 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 $(VI_{v,n})$:

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

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

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

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

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 $(VI_{v,n})$:

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

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

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

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

This was typeset using the Times font (available in any T_EX 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 $(VI_{\nu,n})$:

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

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

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

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

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 $(VI_{\nu,n})$:

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

then the quantity $q = a \frac{aX+Y}{X+aY}$ satisfies as function of $b = a^2$ the P_{VI} differential equation 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 **cm \tt** , **cm \ss** , and **tx \rm** , which are part of standard T_EX distributions.¹

1.2 Overview

1.2.1 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.

¹The first two are available (with no need to load explicitly any package in the document) via the combination **cbfonts** (**cbgreek-complete**) & **babel**, and the LGR encoded **tx \rm** font (again no package loading is necessary) is part of the files of the **txfontsb** package.

As soon as one tries out other fonts for text than the Computer Modern ones one realizes how extremely “thin” are the default T_EX 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:

<p style="text-align: center;"> abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789 ! ? * , . : ; + - = () [] / # \$ % & < > { } and \ </p>

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 `ascii` characters ‘ and - are not in the list:

- 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 -.

Nothing is changed to the “large” math symbols, except for \prod and \sum in in-line math which, like here: $\prod \sum$, will be taken from the Symbol Font if option **symbolmisc** was used.

1.2.2 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}
```

²let’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.

³v1.2 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' .

1.2.3 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).

1.2.4 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”.⁴ Of course it is always possible to leave the responsibility to set up Greek letters in math mode to packages loaded previously to `mathastext`.

1.2.5 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 `\` glyphs. If `mathastext` detects OT1 as the default encoding it will leave these characters to their defaults from the math fonts.⁵

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.

1.3 Main options

1.3.1 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`.

⁴a 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.

⁵the `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.

Nevertheless, soon after I posted the initial version of the package to CTAN, I was overwhelmed by numerous⁶ questions⁷ 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.⁸ There is also an option `frenchmath` to make the uppercase letters nevertheless upright, because this is the way of traditional French mathematical typography.⁹

1.3.2 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`.

1.3.3 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 L^AT_EX 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 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

⁶this means "more than one."

⁷I thank in particular Tariq PERWEZ and Kevin KLEMENT for their kind remarks (chronological order).

⁸more precisely stated, the value of `\itdefault` is used.

⁹more precisely stated, the value of `\shapedefault` is used.

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 **euler-greek**, **symbolgreek** or **symbol**. But **LGRgreek** is receptive to it.

1.4 Math versions

L^AT_EX 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**¹⁰: **lmvtt**, **newcent**, **times**, and **helvet**. The names can be taken arbitrarily (they only need to be suitable arguments to the L^AT_EX **\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 L^AT_EX one can limit the scope to the inside of

¹⁰math versions are discussed in the document `fntguide.pdf` from your T_EX distribution.

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 `\Mathastextversion`, `\MTversion`, `\MTVersion`), with its 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 used in math mode for letters, digits, operator names, punctuations and other ascii symbols,
- sets the font for 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,
- resets the user-defined extra spaces after the symbols \exists , \forall and before the derivative $'$ to the values as decided in the preamble on a *per version* basis,
- (*new with 1.2b*) checks if the used font is upright and then activates the automatic insertion of italic corrections after each letter in math mode,
- 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.¹¹ But one can use instead `\renewcommand{\rmdefault}{epigrafica}`,¹² 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}`.

1.5 Extended scope of the math alphabets commands

Since the initial version of `mathastext`, 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 `]` could not be declared of ‘variable family’ as this would, for reasons of the inner workings of `TEX`, not be compatible with the automatically decided spaces put by `TEX` around them, depending on the context.

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,
2. all non alphabetical characters treated by `mathastext` (among them `!`, `?`, `<`, `>`, `=`, `+`, `-`, `(`, `)`, `[`, `]`, `\`, `|`, etc...) will also obey the math alphabet commands (when they are not used as delimiters). The spaces added by `TEX` before and after are not modified.

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

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

with the traditional default behavior:

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

The mechanism is of course disabled by default for the normal and bold math versions when in *subdued* mode. It is activated by default for all other math versions, or for all math versions when option `subdued` was not passed to the package. The

¹¹may typically give a ‘too many math alphabets’ error message.

¹²sometimes 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 arcana of finding `.fd` files in one’s `TEX` distribution, one should look at the log file of a test document to see if for example `T1` is available for that font; for `epigrafica` it is not, only `OT1` and `LGR` are possible.

commands:

```
\MTmathoperatorsdonotobeymathxx  
\MTnonlettersdonotobeymathxx
```

cancel the new behavior and

```
\MTmathoperatorsobeymathxx  
\MTnonlettersobeymathxx
```

re-activate it.^{13 14}

The brace control sequences `\{` and `\}` have their own (supplementary) switch, which is not activated by default. The reason is that after `\MTexplicitbrace-sobeymathxx`, 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\}`, `\{a, a > b\}`, `\{a, a > b\}`, or even `\{a, a > b\}`.¹⁵

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 L^AT_EX a `\langle`: `\langle`. With standard L^AT_EX `\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 $\langle x \rangle$ (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 $\langle x \rangle$ 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{\langle a,b \rangle}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. `\mathbf{\left<a,b\right>}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. `\mathbf{\mathop{\langle}a,b\mathclose{\rangle}}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. The L^AT_EX standard behavior for `\mathbf{\mathop{\langle}a,b\mathclose{\rangle}}` is $\langle \mathbf{a}, \mathbf{b} \rangle$.

1.6 Italic corrections

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

¹³these commands are to be used outside of math mode. Their scope is limited to the current L^AT_EX environment or group. They use the `\everymath` and `\everydisplay` mechanism so if the document needs to modify these token lists it has to do it in a responsible manner, extending not annihilating their previous contents.

¹⁴when 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 \sim , when ‘non-letters’ are declared to obey math alphabets.

¹⁵this last example uses the `\mathnormalbold` additional alphabet defined by `mathastext`.

For the built-in placement routines of T_EX 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 boxes¹⁶ for the letter *f* in the math italic font to the one from the text italic font: f vs. \textit{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 $\mathbf{f(x)}$ is wider with the standard math italic $\mathit{f(x)}$ (21.31474pt) than it is with the text italic font used in math:¹⁷ $\textit{f(x)}$ (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 T_EX 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 \mathbf{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*.¹⁸ With the italic corrections included our bounding boxes are indeed more alike: f vs. \textit{f} .

Without the italic corrections¹⁹ it is f vs. \textit{f} . I said that \mathbf{f} included the italic correction automatically, but if we tell T_EX 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:

<i>abcdefghijklmnopqrstuvwxy</i>	text italic in text
<i>abcdefghijklmnopqrstuvwxy</i>	text italic in math
<i>abcdefghijklmnopqrstuvwxy</i>	math italic in math
<i>abcdefghijklmnopqrstuvwxy</i>	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 T_EXbook (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 T_EX believe it is. Doing so, we obtain in math mode with the text italic:

<i>abcdefghijklmnopqrstuvwxy</i>	text italic in math
<i>abcdefghijklmnopqrstuvwxy</i>	math italic in math

We saw that the italic correction was taken into account automatically (independently of the value of the interword space font parameter) in expressions such as $\mathbf{f(x)}$. Another clever thing done by T_EX 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

¹⁶let's be honest, we are lying here about what exactly the first of these is bounding; this is explained later!

¹⁷we used simply $\mathit{f(x)}$.

¹⁸these values are for the Latin Modern fonts of course.

¹⁹here we give correctly the bounding box for the math italic *f*... without its italic correction!

very different from f^j ... where the latter was coded with $\mathit{\hbox{\itshape f}}^j$. The inputs $\mathit{\hbox{\itshape f}}^j$ and f^j give almost identical results: f^j vs. 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 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 $\mathrm{T\!E\!X}$ book 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 $\mathit{\hbox{\itshape f}}_i^i$ 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!²⁰ 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 $\mathrm{T\!E\!X}$ 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 $\mathrm{T\!E\!X}$ 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 $\mathrm{T\!E\!X}$ has been told to use it. Let's try Latin Modern Upright for letters in math: f_i^i now gives²¹ f_i^i . We see the italic correction in action for the positioning of the superscript! Compare with $\mathit{\hbox{\itshape f}}_i^i$: f_i^i . Overlapping with $\mathit{\hbox{\itshape f}}_i^i$ gives f_i^i and shows that the upright f has an italic correction which was used to shift the superscript to the right (and it is now in a slightly lower position). Let's now do f_i^i : this gives f_i^i and the subscript is shifted to the right, and is now on the same vertical axis as the superscript. There are also some slight vertical displacements, $\mathit{\hbox{\itshape f}}_i^i$ gives f_i^i .

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

We end up in this strange situation that it is attractive to systematically incor-

²⁰there are also some tiny vertical displacements of the sub- and superscripts.

²¹we just use $\mathit{\hbox{\itshape f}}_i^i$.

porate 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 ff , ff , \mathbf{ff} , ff all give ligatures ff , ff , \mathbf{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.²²

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 T_EX font set-up in math.

The feature-implementing commands `\MTicinmath`, `\MTnoicinmath`, `\MTical-soinmathxx` are described in section 2.3.2.

Note: from brief testing on 2012/12/28, X_YT_EX seems not to create fake italic corrections for OpenType fonts. Hence the T_EX 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 X_YT_EX engine. On the other hand LuaL^AT_EX 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 T_EX 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 pdfL^AT_EX. All three outputs are different on my T_EX installation. X_YT_EX does not have the italic corrections. LuaL^AT_EX does, but only for the italic font. pdfL^AT_EX has them for both the italic and the upright font. The fonts used are the Latin Modern fonts, in OpenType format for the unicode engines.

1.7 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`.

²²the situation is rather ironical!

They are especially useful when using an upright font in math mode.²³

- 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 that 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' .²⁴

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. Or, the commands may be used directly in the document.

Also, when the `subdued` option has been used, the normal and bold math versions have by default zero length skips (no preamble setting will change that). Nevertheless it is possible to use also then the commands in the body of the document.

1.8 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`:

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 T_EX 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

²³The `mu` is a unit length (‘math unit’) used in math mode. Its value is relative to the current math style (normal size, subscript size, subsubscript size, ...).

²⁴the derivative glyph from the `txfonts` math symbols adapts itself better to an upright letter, no skip seems to be needed then.

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`.

1.8.1 Shape of Greek letters

Classic T_EX 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 `\MTgreek-font{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...

1.9 Unicode engines

`mathastext` has been made minimally unicode-aware and can be used with $\X_{\text{f}}\text{TEX}$ or $\text{Lua}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, 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 $\X_{\text{f}}\text{TEX}$), 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 \LuaTeX 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 \LuaTeX engine than it is with \XeTeX . This is a temporary situation as the needed feature of \LuaTeX 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 1.6 the fact that the italic corrections were not available for OpenType fonts under the \XeTeX engine and only partially available for the \LuaTeX 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, but I include here two examples which compiled successfully with \XeTeX and \LuaTeX , the first one on a Linux machine, the second one on a Mac OS X machine.²⁵

```
\documentclass{article}
\usepackage[hscale=0.8]{geometry}
\usepackage{multicol}
\usepackage[no-math]{fontspec}
\usepackage{lmodern}
```

²⁵running `latex` on the source of this document `mathastext.dtx` (in a temporary repertory) extracts the test files.

1.10 Compatibility issues

Compatibility issues 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`, especially when using Unicode engines (and also `fontspec` with its option `no-math` should be before `mathastext`).

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 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 to math operator names. And the already mathematically active right quote is modified to have some extra space added before the derivative glyph `'`. Version 1.2b does this in a careful manner with the hope to not create conflict with packages (typically, language definition files from the `babel` system) making characters ‘globally active’ in the document.²⁶ All the action done by `mathastext` is done anew at each math environment and has no effect elsewhere.

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

Doing (after the `\begin{document}`) `\everymath={}` and `\everydisplay={}` 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`, `\MTnoicinmath`, and `\MTnormalprime` (which cancels the redefinition done by `mathastext` of the action of the right quote `'` in math mode).

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

2 Package options and commands

2.1 Summary of main options

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

subdued: acts in a subdued way. The L^AT_EX normal and bold math versions are left

²⁶it is better if these characters are not ‘active’ at the time of loading `mathastext`, and fortunately `babel` does the activation of characters only at the `\begin{document}`.

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

2.2 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.²⁷ This specific issue will perhaps be fixed by some hypothetical 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

²⁷To 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 $\text{\XeTeX/Lua}\text{\TeX}$. 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.

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

X_YTeX and Lua \LaTeX : 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 Lua \LaTeX (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 Pdf \LaTeX , there is no such limitation and all 8bit-encodings (containing these glyphs) should be ok.

fontspec: one more note to users of X_YTeX/Lua \LaTeX 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 `\boldsymbol` command from the `amssy` package). As it does not exist

in the default L^AT_EX 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 T_EX 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.

2.3 Commands

2.3.1 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.²⁸
 - It can also be preceded optionally by one or more of²⁹ `\MTencoding{<enc>}`, `\MTfamily{<fam>}`, `\MTseries{<ser>}`, `\MTshape{<sh>}`, and, new with version 1.1, `\MTlettershape{<sh>}`. For example valid values are, respectively, `<T1>`, `<phv>`, `<m>`, `<n>`, and `<it>`: this is the Helvetica font

²⁸updates also the font and shapes for the Greek letters (`LGRgreek` option), and the skips to be inserted after the symbols \forall and \exists , see *infra*.

²⁹these commands exist also with long names: `\Mathastextencoding`, etc. . . The same applies to the other commands mentioned in this section.

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.³⁰
- `\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).³¹ When the optional argument is absent, and `mathastext` was loaded with the `italic` option, then the default letter shape will be `it`,³² 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.

³⁰The 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.

³¹I do not especially recommend to use this in real life!

³²more 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.

2.3.2 Commands usable everywhere

- `\MTexistsskip{<math glue>}`: specifies the amount of skip or more generally glue to put after each \exists math symbol. Indeed, upright letters (or digits for that matter) often appear to be positioned a bit too close to the quantifier: $\exists B$. The package default is to add a `1mu` skip (this default is set to zero in the case of `italic`): $\exists B$. 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 \forall (except with the option `italic` for which the default is set to zero). Compare $\forall F$ (has the skip) with $\forall F$ (has no skip). Use this command in the preamble to set up the skip or glue to be used in the *next to be declared* math versions. In the case of the option `subdued`, the 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.
- `\MTmathoperatorsobeymathxx`, `\MTmathoperatorsdonotobeymathxx`, `\MTnonlettersobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTexplicitbracesobeymathxx`, `\MTexplicitbracesdonotobeymathxx`: these commands without argument must be used *outside* of math mode. `mathastext` issues automatically `\MTmathoperatorsobeymathxx` and `\MTnonlettersobeymathxx` at the time of `\usepackage{mathastext}`, except in case of the `subdued` option. The ‘explicit braces’ activation works only when `\MTnonlettersobeymathxx` has 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 issued in as well as outside of math mode.³³ To cancel its effect either enclose it in a group or environment or re-issue `\MTicinmath` after it.

2.3.3 Body-only commands

- `\MTversion` [*<nametext>*] {*<namemath>*}:³⁴ 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 L^AT_EX command `\mathversion`. However this is not recommended as `mathastext` does quite a lot of additional configuration when invoked with `\MTversion`.

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 1.6. 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.

³³it is provided for the `mathastext` defined math alphabet `\mathnormalbold` which contrarily to `\mathnormal` is not treated apart from the other math alphabets.

³⁴`\MTversion` is also available as `\MTVersion`, `\Mathastextversion` and `\mathastextversion`.

- `\hbar`: this is constructed (in a way compatible with the `italic` option) from the `h` letter and the `ˉ` 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 L^AT_EX 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 L^AT_EX 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.

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

Regarding the `\prod` and `\sum` commands: they will use the Symbol glyphs $\prod \Sigma$ in inline math, and in display math the Computer Modern ones (or whatever is set up by other packages) :

$$\prod \Sigma$$

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.

2.4 Complete list of options

- `basic`: only mathastextify letters and digits.
- `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.
- `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 not do it. The option `nolessnomore` is activated by default in case of OT1-encoding.
- further excluding options: `noexclam ! ?`, `noasterisk *`, `nopunctuation , . : ;`, `no-plus`, `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 ($\dot{\triangle}$).
- `symbolmisc`: takes quite a few glyphs, including logical arrows, product and sum signs from Symbol. They are listed *supra*. A `\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.

- **LGRgreek**s: 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.
- **selfGreek**s: 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 L^AT_EX, 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’ L^AT_EX 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 L^AT_EX 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>

3 Implementation

```

1 \ProvidesPackage{mathastext}
2 [2012/12/31 v1.2b Use the text font in math mode (jfB)]
3 \NeedsTeXFormat{LaTeX2e}

```

Testing for XeTeX and LuaLaTeX. I should load some package for this code picked up somewhere.

```

4 \newif\ifXeTeX
5 \begingroup\expandafter\expandafter\expandafter\endgroup
6 \expandafter\ifx\csname XeTeXinterchartoks\endcsname\relax
7   \XeTeXfalse
8 \else
9   \XeTeXtrue
10 \fi
11 \newif\ifLuaTeX
12 \begingroup\expandafter\expandafter\expandafter\endgroup
13 \expandafter\ifx\csname directlua\endcsname\relax
14   \LuaTeXfalse
15 \else
16   \LuaTeXtrue
17 \fi

```

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

```

18 \def\mst@oti{OT1}\def\mst@ti{T1}\def\mst@lyi{LY1}
19 \def\mst@eui{EU1}\def\mst@euii{EU2}
20 \newif\ifmst@goahead
21 \newif\ifmst@abort
22 \newif\ifmst@optionalparam

```

\mst@enc Macros to store the font settings, each math version will store its own records.

```

\mst@fam 23 \def\mst@enc{\encodingdefault}
\mst@ser 24 \def\mst@fam{\familydefault}
\mst@opsh 25 \def\mst@ser{\seriesdefault}
\mst@bold 26 \def\mst@opsh{\shapedefault} %% will be default shape for operator names
\mst@ltsh 27 \def\mst@bold{\bfdefault}
28 \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**.

```

29 \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.

```

30 \newif\ifmst@italic
31 \newif\ifmst@frenchmath
32 \DeclareOption{italic}{\mst@italictrue}
33 \def\mst@ltsh{\itdefault}}
34 \DeclareOption{frenchmath}{\mst@frenchmathtrue\mst@italictrue}
35 \def\mst@ltsh{\itdefault}}
36 %%

```

```

37 \newif\ifmst@endash\mst@endashtrue
38   \DeclareOption{endash}{\mst@endashtrue}
39   \DeclareOption{noendash}{\mst@endashfalse}
40 \newif\ifmst@emdash
41   \DeclareOption{emdash}{\mst@emdashtrue\mst@endashfalse}
42 %%
43 \newif\ifmst@alldelims
44 \edef\mst@tmp{\encodingdefault}\ifx\mst@oti\mst@tmp\else\mst@alldelimstrue\fi
45   \DeclareOption{alldelims}{\mst@alldelimstrue}
46   \DeclareOption{nolessnomore}{\mst@alldelimsfalse}
47 %% new with 1.2
48 \newif\ifmst@nosmalldelims
49   \DeclareOption{nosmalldelims}{\mst@nosmalldelimstrue}
50 %%
51 \newif\ifmst@noplus
52   \DeclareOption{noplus}{\mst@noplustrue}
53 \newif\ifmst@nominus
54   \DeclareOption{nominus}{\mst@nominustrue}
55 \DeclareOption{noplusnominus}{\ExecuteOptions{noplus,nominus}}
56 %%
57 \newif\ifmst@noparen
58   \DeclareOption{noparenthesis}{\mst@noparenttrue}
59 \newif\ifmst@nopunct
60   \DeclareOption{nopunctuation}{\mst@nopuncttrue}
61 \newif\ifmst@noequal
62   \DeclareOption{noequal}{\mst@noequaltrue}
63 \newif\ifmst@noexclam
64   \DeclareOption{noexclam}{\mst@noexclamtrue}
65 \newif\ifmst@noast
66   \DeclareOption{noasterisk}{\mst@noasttrue}
67 \newif\ifmst@nospecials
68   \DeclareOption{nospecials}{\mst@nospecialstrue}
69 \DeclareOption{basic}{\ExecuteOptions{noparenthesis,nopunctuation,%
70 noplusnominus,noequal,noexclam,noasterisk,nospecials,nolessnomore}}
71 %%
72 \newif\ifmst@nohbar
73   \DeclareOption{nohbar}{\mst@nohbartrue}
74 \newif\ifmst@nodigits
75   \DeclareOption{nodigits}{\mst@nodigitstrue}
76 \newif\ifmst@defaultimath
77   \DeclareOption{defaultimath}{\mst@defaultimathtrue}
78 \newif\ifmst@mathaccents
79   \DeclareOption{mathaccents}{\mst@mathaccentstrue}
80 %%
81 \newif\ifmst@needsymbol
82 \newif\ifmst@symboldelimiters
83   \DeclareOption{symboldelimiters}{\mst@needsymboltrue\mst@symboldelimiterstrue}
84 \newif\ifmst@symboldigits
85   \DeclareOption{symboldigits}{\mst@needsymboltrue\mst@symboldigitstrue}

```

```

86 \newif\ifmst@symbolgreek
87   \DeclareOption{symbolgreek}{\mst@needsymboltrue\mst@symbolgreektrue}
88 \newif\ifmst@symbolre
89   \DeclareOption{symbolre}{\mst@needsymboltrue\mst@symbolretrue}
90 \newif\ifmst@symbolmisc
91   \DeclareOption{symbolmisc}{\mst@needsymboltrue\mst@symbolmisctrue}
92 \DeclareOption{symbol}{\ExecuteOptions{symbolgreek,symbolmisc,symbolre}}
93 \DeclareOption{symbolmax}{\ExecuteOptions{symbol,symboldelimiters}}
94 %%
95 \newif\ifmst@needeuler
96 \newif\ifmst@eulerdigits
97   \DeclareOption{eulerdigits}{\mst@needeulertrue\mst@eulerdigitstrue}
98 \newif\ifmst@eulergreek
99   \DeclareOption{eulergreek}{\mst@needeulertrue\mst@eulergreektrue}
100 %%
101 \newif\ifmst@selfGreek
102   \DeclareOption{selfGreek}{\mst@selfGreektrue}
103 \newif\ifmst@selfGreeks
104   \DeclareOption{selfGreeks}{\mst@selfGreekstrue\mst@selfGreektrue}
105 \newif\ifmst@LGRgreek
106   \DeclareOption{LGRgreek}{\mst@LGRgreektrue}
107 \newif\ifmst@LGRgreeks
108   \DeclareOption{LGRgreeks}{\mst@LGRgreekstrue\mst@LGRgreektrue}
109 %%
110 \def\mst@greek@select{0}
111 \newif\ifmst@itgreek
112 \newif\ifmst@upgreek
113   \DeclareOption{itgreek}{\mst@itgreektrue}
114   \DeclareOption{upgreek}{\mst@upgreektrue}
115   \DeclareOption{itGreek}{\def\mst@greek@select{1}}
116   \DeclareOption{upGreek}{\def\mst@greek@select{2}}
117 %%

```

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.

```

118 \newif\ifmst@defaultnormal
119   \DeclareOption{defaultnormal}{\mst@defaultnormaltrue}
120 \newif\ifmst@defaulttrm
121   \DeclareOption{defaulttrm}{\mst@defaulttrmtrue}
122 \newif\ifmst@defaultbf
123   \DeclareOption{defaultbf}{\mst@defaultbftrue}
124 \newif\ifmst@defaulttit
125   \DeclareOption{defaulttit}{\mst@defaulttittrue}
126 \newif\ifmst@defaultsf
127   \DeclareOption{defaultsf}{\mst@defaultsftrue}
128 \newif\ifmst@defaultttt
129   \DeclareOption{defaultttt}{\mst@defaultttttrue}
130 \newif\ifmst@nonnormalbold

```



```

131 \DeclareOption{defaultalphabets}{\ExecuteOptions{defaultnormal,defaultrm,%
132 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.
133 \newif\ifmst@defaultsizes
134     \DeclareOption{defaultmathsizes}{\mst@defaultsizetrue}
135 \newif\ifmst@twelve
136     \DeclareOption{12pt}{\mst@twelvetrue}
137 \newif\ifmst@fouriervec
138     \DeclareOption{fouriervec}{\mst@fouriervectrue}
    1.15: the subdued option.
139 \newif\ifmst@subdued
140     \DeclareOption{subdued}{\mst@subduedtrue}
141 \DeclareOption*{\PackageWarning{mathastext}{Unknown option ‘\CurrentOption’}}
142 \ProcessOptions\relax

```

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

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

```

143 \newmuskip\mst@exists@muskip %% v 1.15e
144 \newmuskip\mst@forall@muskip
145 \def\mst@exists@skip{1mu}
146 \def\mst@forall@skip{.6667mu}
147 \ifmst@italic\ifmst@frenchmath\else
148     \def\mst@exists@skip{0mu}\def\mst@forall@skip{0mu}\def\mst@prime@skip{0mu}
149 \fi\fi
150 \AtEndOfPackage{% must be at end of code
151 \let\mst@stdexists\exists
152 \let\mst@stdforall\forall
153 \renewcommand\exists{\mst@stdexists\mskip\mst@exists@muskip}
154 \renewcommand\forall{\mst@stdforall\mskip\mst@forall@muskip}
155 }

```

`\prime` 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, 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.

The macro `\mst@modifyprime` is defined here and not At Begin Document as this could be a time when the right quote has been made active. The safe fall-back definition (for use in an `\hbox` or a `\label` inside math for example), which is made anew each time by `\mst@mathactivate` at each math mode entrance, invokes, in case `'` is *not* active then, the `'` with its catcode as seen by T_EX when it first reads `\mst@mathactivate'`..., so here when it stores the definition of `\mst@modifyprime`. Hence there would be a problem if this happened at the time of an active `'` and then the `'` is *not* active anymore when entering a math mode list. To be extra safe, I should perhaps issue a local `\@makeother'` here, but is it really worth the trouble? (and if I did it for `'` I would have for consistency to apply the same safeguard to all the other characters treated later by `mathastext`).

The `\string'` below seems to contradict my belief in a non-active `'` here but it is just to be extra sure loading the package is safe. A user command `\MTnormalprime` is provided in case there is a problem later.

```
156 \newmuskip\mst@prime@muskip    %% v 1.2
157 \def\mst@prime@skip{.5mu}
158 \ifmst@italic\ifmst@frenchmath\else\def\mst@prime@skip{0mu}\fi\fi
159 \def\mst@active@math@prime{\sp\bgroup\mskip\mst@prime@muskip\prim@s}
160 \def\mst@modifyprime{\mst@mathactivate'\mst@active@math@prime{}}
161 \expandafter\let\csname mst@ifdone\string'\endcsname\iffalse
162 \AtBeginDocument{
163 \everymath\expandafter{\the\everymath \mst@modifyprime}
164 \everydisplay\expandafter{\the\everydisplay \mst@modifyprime}
165 }
166 \newcommand\MTnormalprime{\let\mst@modifyprime\relax}
```

`\MTexistsskip` 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.

`\MTforallskip`
`\MTprimeskip`

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.

```
167 \newcommand*\MTexistsskip[1]{\edef\mst@exists@skip{#1}%
168   \expandafter\mst@exists@muskip\mst@exists@skip\relax}
169 \newcommand*\MTforallskip[1]{\edef\mst@forall@skip{#1}%
170   \expandafter\mst@forall@muskip\mst@forall@skip\relax}
171 \newcommand*\MTprimeskip[1]{\edef\mst@prime@skip{#1}%
172   \expandafter\mst@prime@muskip\mst@prime@skip\relax}
173 \let\Mathastextexistsskip\MTexistsskip
174 \let\Mathastextforallskip\MTforallskip
175 \let\Mathastextprimeskip\MTprimeskip
176 \let\mathastextexistsskip\MTexistsskip
177 \let\mathastextforallskip\MTforallskip
178 \let\mathastextprimeskip\MTprimeskip
```

`\MTnonletters-` 1.2 [2012/12/20] does some rather daring *math* activation of `!,?,+,-,=,<,>,(,),[,],|` in
`obeymathxx`
`\MTnonletters-`
`donotobeymathxx`

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 mechanism is 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 to.

`\mst@mathactivate` 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`.

`\mst@fork` 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 new code provides (let's hope) correct behavior: the `\mst@fork` macro will expand the active character to its `mathastext` re-definition *only* in math mode and *only* if `\protect` is `\@typeset@protect`. This covers situations such as `$(\hbox{?})$` with an active `?` or `$(\label{eq:1})$` with an active `:` (an iteration of math modes and restricted horizontal modes is no problem).

In the case the character is not active at the entrance of the math mode, the fall-back macro `\mst@original<char>` expands to the character with its catcode at the time of the loading of the package, so, *it is assumed*, 11 for the letters a-z, A-Z, and 12 for the other concerned characters. Perhaps this is not the catcode of the character at the time the current math mode is now entered. As the fall-back macro will be activated presumably when writing to a display or file, or building a `\csname...\endcsname`, being of catcode 11 or 12 should not be an issue. If the character has been assigned a catcode neither 11 nor 12, then \TeX does not look at its `\mathcode`, so what `mathastext` does in `\mst@mathactivate` is of no consequence. And in the case of a character which has catcode 13 when entering math mode, the precautions taken have been explained in the previous paragraph.

`\mst@do@az` 1.2b [2012/12/28] now that we understand the great advantages of "8000 we do it also for all
`\mst@do@AZ` letters a-z and A-Z to insert automatically the italic corrections. See the discussion in the user
`\mst@addtodo@az` manual. Ironically I wrote the code initially for the *italic* option only to realize later it was
`\mst@addtodo@AZ` more suitable to using an *upright* text font in math mode!

`\MTicinmath` One may question the fact of doing this at every math/display, as it seems costly. Why not do
`\MTICinmath` it once and for all? well, processor chips are fast these days. True, for the case of the characters
`\MTnoicinmath` such as `?`, `!`, `>`, `<`, ... which could reasonably be expected to be made active elsewhere in the
`\MTicalsoinmathxx` document, it is certainly a great simplification for me to do my business at every math/display,
`\mst@the` as I store the then current definitions of the possibly activated character. For the letters, which
are not really in danger of being made active in the document, I could possibly have moved the
`\mathcode="8000` thing to the document preamble, but to cancel it (in the document body)
would mean to have to reset all mathcodes, rather than let simply as is done here a `\the` to be
a `\@gobble`.

`\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 `\mathnormabold` (will probably return to this point 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.

```

179 \newtoks\mst@do@nonletters
180 \newtoks\mst@do@az
181 \newtoks\mst@do@AZ
182 \newcommand\MTnonlettersdonotobeymathxx{\let\mst@the\@gobble}
183 \newcommand\MTnonlettersobeymathxx{\let\mst@the\the}
184 \MTnonlettersobeymathxx
185 \newcommand\MTicinmath{%
186     \let\mst@thei\the
187     \let\mst@theI\the
188     \def\mst@itcorr{\ifnum\fam=\m@ne\else\ifnum\fam=\symmtletterfont\fi\fi}%
189     \let\mst@ITcorr\mst@itcorr}
190 \newcommand\MTnoicinmath{% also usable in math mode
191     \def\mst@itcorr{}\def\mst@ITcorr{}\%
192     \let\mst@thei\@gobble\let\mst@theI\@gobble}
193 \newcommand\MTICinmath{%
194     \let\mst@theI\the
195     \def\mst@ITcorr{\ifnum\fam=\m@ne\fi}}
196 \newcommand\MTnoICinmath{% also usable in math mode
197     \def\mst@ITcorr{}\%
198     \let\mst@theI\@gobble}
199 \newcommand\MTicalsoinmathxx{%
200     \ifx\mst@itcorr\empty\else\def\mst@itcorr{\fi}
201     \ifx\mst@ITcorr\empty\else\def\mst@ITcorr{\fi}
202 \AtBeginDocument{
203 \everymath\expandafter{\the\everymath \mst@the\mst@do@nonletters
204 \mst@thei\mst@do@az \mst@theI\mst@do@AZ}
205 \everydisplay\expandafter{\the\everydisplay \mst@the\mst@do@nonletters
206 \mst@thei\mst@do@az \mst@theI\mst@do@AZ}
207 }
208 \def\mst@fork{%
209     \ifx\protect\@typeset@protect\ifmmode
210         \expandafter\expandafter\expandafter\@thirdofthree
211         \fi\fi
212     \@firstoftwo}
213 \begingroup % make sure ~ is active
214 \catcode'\~=\active
215 \def\x{\endgroup
216 \def\mst@mathactivate##1##2##3{%
217     \begingroup
218     \lccode'\~='##1
219     \lccode'##1='##1
220     \lowercase{\endgroup
221     \csname mst@ifdone\string##1\endcsname
222     \else
223         \expandafter\expandafter\expandafter
224         \let\expandafter\expandafter\csname mst@original\string##1\endcsname
225         \expandafter=\ifnum\catcode'##1=\active~\else##1\fi
226         \expandafter\def\csname mst@mathactive\string##1\endcsname{##2##3}%
227         \mathcode'##1="8000

```

```

228      \edef~{\noexpand\mst@fork
229      \expandafter\noexpand\csname mst@original\string##1\endcsname
230      \expandafter\noexpand\csname mst@mathactive\string##1\endcsname}%
231      \fi}%
232      \expandafter\let\csname mst@ifdone\string##1\endcsname=\iftrue}
233 }\x
234 \def\mst@addtodo@nonletters#1#2#3{
235   \expandafter\let\csname mst@ifdone\string#1\endcsname\iffalse
236   \mst@do@nonletters\expandafter{\the\mst@do@nonletters\mst@mathactivate#1#2#3}
237 }
238 \def\mst@addtodo@az#1#2#3{
239   \expandafter\let\csname mst@ifdone\string#1\endcsname\iffalse
240   \mst@do@az\expandafter{\the\mst@do@az\mst@mathactivate#1#2#3}
241 }
242 \def\mst@addtodo@AZ#1#2#3{
243   \expandafter\let\csname mst@ifdone\string#1\endcsname\iffalse
244   \mst@do@AZ\expandafter{\the\mst@do@AZ\mst@mathactivate#1#2#3}
245 }

```

`\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).

```

246 \@ifpackageloaded{amsmath}
247 {\let\newmcodes@relax} % brutal, but bye bye and don't come back.
248 {}

```

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’, respectively¹. Due to the reassignments 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 Lua^AT_EX 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.

```

249 \ifmst@subdued
250 \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
251 \def\mst@normalmv@letter{#2}}
252 \expandafter\mst@reserved\mv@normal\@nil
253 \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
254 \def\mst@boldmv@letter{#2}}
255 \expandafter\mst@reserved\mv@bold\@nil
256 \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
257 \def\mst@normalmv@operator{#2}}
258 \expandafter\mst@reserved\mv@normal\@nil
259 \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
260 \def\mst@boldmv@operator{#2}}
261 \expandafter\mst@reserved\mv@bold\@nil
262 %%
263 \edef\mst@tmp{\encodingdefault}
264 \def\mst@reserved#1/#2/#3/#4/{\gdef\mst@debut{#1}\gdef\mst@reste{#2/#3/#4}}
265 \begingroup\escapechar\m@ne
266 \xdef\mst@funnyoti{\expandafter\string\csname OT1\endcsname}
267 \expandafter\expandafter\expandafter
268 \mst@reserved\expandafter\string\mst@normalmv@operator/
269 \endgroup
270 \ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
271 \edef\mst@normalmv@operator{\expandafter\noexpand\csname
272 \mst@tmp/\mst@reste\endcsname}
273 \begingroup\escapechar\m@ne
274 \expandafter\expandafter\expandafter
275 \mst@reserved\expandafter\string\mst@boldmv@operator/
276 \endgroup
277 \ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
278 \edef\mst@boldmv@operator{\expandafter\noexpand\csname
279 \mst@tmp/\mst@reste\endcsname}
280 \AtEndOfPackage{
281 \typeout{** ...entering subdued mode...}
282 \expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@letter\symmtletterfont
283 \expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@letter\symmtletterfont
284 \expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@operator\symmtoperatorfont
285 \expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@operator\symmtoperatorfont
286 \typeout{** ...done.}
287 }
288 \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.

```
289 \DeclareSymbolFont{mtoperatorfont}
290 {\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}
```

\operator@font We modify this L^AT_EX 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- 1.2: rather than just replacing `\symoperators` by `\symmtoperatorfont` I add a modification
obeymathxx which makes the declared operator names sensitive to the math alphabets... ouh le vilain!

```
\MTmathoperators- 291 \newcommand{\MTmathoperatorsobeymathxx}
donot- 292 {\def\operator@font{\mathgroup\ifnum\fam=\m@ne\symmtoperatorfont\else\fam\fi}}
obeymathxx 293 \newcommand{\MTmathoperatorsdonotobeymathxx}
294 {\def\operator@font{\mathgroup\symmtoperatorfont}}
295 \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.

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

\Mathnormal We redefine the default normal, rm, bf, it, sf, and tt alphabets, but this will be done via
\Mathrm `\renewcommand{\mathrm}{\Mathrm}` etc... and (1.15f) the previous status of the math alpha-
\Mathbf bets is recorded for the sake of the subdued option.

\Mathit We follow the standard L^AT_EX behavior for `\mathbf`, which is to pick up the bold series of the
\Mathsf roman font (digits and operator names).

\Mathtt We will access (if no option is passed for Greek) the `\omicron` via `\mathnormal`. But un-
\mathnormalbold fortunately 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`.

```

298 \let\mst@alph@omicron\mathnormal
299 \@ifpackageloaded{fourier}{\ifsloped\else\let\mst@alph@omicron\mathrm\fi}{\fi}
300 \DeclareSymbolFontAlphabet{\Mathnormal}{\mtletterfont}
301 \DeclareSymbolFontAlphabet{\Mathrm}{\mtoperatorfont}
302 \ifmst@nonnormalbold\else
303   \DeclareMathAlphabet{\mathnormalbold}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@ltsh}
304 \fi\ifmst@defaultbf\else
305   \DeclareMathAlphabet{\Mathbf}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@opsh}
306 \fi\ifmst@defaultit\else
307   \DeclareMathAlphabet{\Mathit}{\mst@enc}{\mst@fam}{\mst@ser}{\itdefault}
308 \fi\ifmst@defaultsf\else
309   \DeclareMathAlphabet{\Mathsf}{\mst@enc}{\sfdefault}{\mst@ser}{\mst@opsh}
310 \fi\ifmst@defaulttt\else
311   \DeclareMathAlphabet{\Mathtt}{\mst@enc}{\ttdefault}{\mst@ser}{\mst@opsh}
312 \fi
313 \let\mst@original@normal\mathnormal
314 \let\mst@original@rm\mathrm
315 \let\mst@original@bf\mathbf
316 \let\mst@original@it\mathit
317 \let\mst@original@sf\mathsf
318 \let\mst@original@tt\mathtt
319 \def\mst@restorealphabets{% for subdued
320 \let\mathnormal\mst@original@normal
321 \let\mathrm\mst@original@rm
322 \let\mathbf\mst@original@bf
323 \let\mathit\mst@original@it
324 \let\mathsf\mst@original@sf
325 \let\mathtt\mst@original@tt}
326 \def\mst@setalphabets{%
327 \ifmst@defaultnormal\else\renewcommand{\mathnormal}{\Mathnormal}\fi
328 \ifmst@defaultrm\else\renewcommand{\mathrm}{\Mathrm}\fi
329 \ifmst@defaultbf\else\renewcommand{\mathbf}{\Mathbf}\fi
330 \ifmst@defaultit\else\renewcommand{\mathit}{\Mathit}\fi
331 \ifmst@defaultsf\else\renewcommand{\mathsf}{\Mathsf}\fi
332 \ifmst@defaulttt\else\renewcommand{\mathtt}{\Mathtt}\fi}
333 \ifmst@subdued\else\mst@setalphabets\fi

```

LGRgreek 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 `selfGreek` 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.

```

334 \def\mst@update@greeksh{
335   \def\mst@greek@lsh{\mst@ltsh} %% default behavior
336   \def\mst@greek@ush{\mst@opsh}
337   \ifmst@itgreek\def\mst@greek@lsh{\itdefault}
338                   \def\mst@greek@ush{\itdefault}\fi
339   \ifmst@upgreek\def\mst@greek@lsh{\updefault}
340                   \def\mst@greek@ush{\updefault}\fi
341   \ifmst@frenchmath
342     \ifmst@itgreek\else
343     \ifmst@upgreek\else
344       \def\mst@greek@lsh{\mst@opsh}
345       \def\mst@greek@ush{\mst@opsh}
346     \fi\fi
347   \fi
348   \ifcase\mst@greek@select
349     \or\def\mst@greek@ush{\itdefault}
350     \or\def\mst@greek@ush{\updefault}
351   \fi}
352 \mst@update@greeksh
353 \ifmst@LGRgreek
354   \DeclareFontEncoding{LGR}{-}{-}
355   \DeclareSymbolFont{mtlgrfontlower}{LGR}
356                     {\mst@fam}{\mst@ser}{\mst@greek@lsh}
357   \DeclareSymbolFont{mtlgrfontupper}{LGR}
358                     {\mst@fam}{\mst@ser}{\mst@greek@ush}
359 \else
360 \ifmst@selfGreek
361   \DeclareSymbolFont{mtselfGreekfont}{OT1}
362                     {\mst@fam}{\mst@ser}{\mst@greek@ush}
363 \fi\fi

```

`mteulervm` In case we need the Euler font, we declare it here. It will use `uzeur.fd` from the `eulervm` package
`\MathEuler` of Walter SCHMIDT
`\MathEulerBold`

```

364 \ifmst@needeuler
365 \DeclareSymbolFont{mteulervm}{U}{zeur}{m}{n}
366 \DeclareSymbolFontAlphabet{\MathEuler}{mteulervm}
367 \DeclareMathAlphabet{\MathEulerBold}{U}{zeur}{\mst@bold}{n}
368 \fi
369 \newcommand\MTEulerScale[1]{\edef\zeu@Scale{#1}}
370 \let\MathastextEulerScale\MTEulerScale

```

L^AT_EX 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`,


```

387 \newcommand*\MTencoding[1]{\def\mst@enc{#1}}
388 \newcommand*\MTfamily[1]{\def\mst@fam{#1}}
389 \newcommand*\MTseries[1]{\def\mst@ser{#1}}
390 \newcommand*\MTshape[1]{\def\mst@opsh{#1}}
391 \ifmst@italic\else\def\mst@ltsh{#1}\fi}
392 \newcommand*\MTboldvariant[1]{\def\mst@bold{#1}}
393 \newcommand*\MTlettershape[1]{\def\mst@ltsh{#1}}
394 \@onlypreamble\MTencoding
395 \@onlypreamble\MTfamily
396 \@onlypreamble\MTseries
397 \@onlypreamble\MTshape
398 \@onlypreamble\MTboldvariant
399 \@onlypreamble\MTlettershape
400 \let\Mathastextencoding\MTencoding
401 \let\Mathastextfamily\MTfamily
402 \let\Mathastextseries\MTseries
403 \let\Mathastextshape\MTshape
404 \let\Mathastextboldvariant\MTboldvariant
405 \let\Mathastextlettershape\MTlettershape

```

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

```

406 \newcommand\MTitgreek{%
407 \mst@itgreektrue\mst@upgreekfalse\def\mst@greek@select{0}}
408 \newcommand\MTupgreek{%
409 \mst@upgreektrue\mst@itgreekfalse\def\mst@greek@select{0}}
410 \newcommand\MTitGreek{\def\mst@greek@select{1}}
411 \newcommand\MTupGreek{\def\mst@greek@select{2}}
412 \@onlypreamble\MTitgreek
413 \@onlypreamble\MTupgreek
414 \@onlypreamble\MTitGreek
415 \@onlypreamble\MTitGreek
416 \let\Mathastextitgreek\MTitgreek
417 \let\Mathastextupgreek\MTupgreek
418 \let\MathastextitGreek\MTitGreek
419 \let\MathastextupGreek\MTupGreek
420 \newcommand*\MTgreekfont[1]{\def\mst@greekfont{#1}}
421 \@onlypreamble\MTgreekfont
422 \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`.

```

423 \newcommand*\MTWillUse[5][\@empty]{
424 \MTencoding{#2}
425 \MTfamily{#3}
426 \MTseries{#4}
427 \MTshape{#5}

```

```

428 \ifmst@italic\MTlettershape{\itdefault}\fi % was missing in v 1.14 and prior
429 \ifx\@empty#1
430 \else
431 %% we check if optional argument was [], this does not detect [ ]
432 \def\mst@tmp{#1}
433 \ifx\@empty\mst@tmp\else\MTlettershape{#1}\fi
434 \fi
435 \Mathastext}
436 \@onlypreamble\MTWillUse
437 \let\MathastextWillUse\MTWillUse
438 \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.

```

439 \newcommand*\Mathastext[1][\@empty]{
440 \ifx\@empty#1 % no optional argument
441 \mst@update@greeksh
442 \edef\mst@encoding@normal{\mst@enc}
443 \edef\mst@family@normal{\mst@fam}
444 \edef\mst@series@normal{\mst@ser}
445 \edef\mst@shape@normal{\mst@opsh}
446 \edef\mst@ltshape@normal{\mst@ltsh}
447 \edef\mst@itdefault@normal{\itdefault}
448 \edef\mst@rmdefault@normal{\rmdefault}
449 \edef\mst@sfdefault@normal{\sfdefault}
450 \edef\mst@ttdefault@normal{\ttdefault}
451 \edef\mst@boldvariant@normal{\mst@bold}
452 \edef\mst@exists@skip@normal{\mst@exists@skip}
453 \edef\mst@forall@skip@normal{\mst@forall@skip}
454 \edef\mst@prime@skip@normal{\mst@prime@skip}
455 \edef\mst@encoding@bold{\mst@enc}
456 \edef\mst@family@bold{\mst@fam}
457 \edef\mst@series@bold{\mst@bold}
458 \edef\mst@shape@bold{\mst@opsh}
459 \edef\mst@ltshape@bold{\mst@ltsh}
460 \edef\mst@boldvariant@bold{\mst@bold}
461 \edef\mst@itdefault@bold{\itdefault}
462 \edef\mst@rmdefault@bold{\rmdefault}
463 \edef\mst@sfdefault@bold{\sfdefault}
464 \edef\mst@ttdefault@bold{\ttdefault}
465 \edef\mst@exists@skip@bold{\mst@exists@skip}
466 \edef\mst@forall@skip@bold{\mst@forall@skip}
467 \edef\mst@prime@skip@bold{\mst@prime@skip}

```

```

468 \MTicinmath
469 \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@ltshape@normal\MTnoicinmath\fi}%
470 \ifmst@frenchmath
471 \MTICinmath
472 \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@shape@normal\MTnoICinmath\fi}%
473 \fi
474 \ifmst@subdued
475 \MTnonlettersdonotobeymathxx
476 \MTmathoperatorsdonotobeymathxx
477 \MTnoicinmath
478 \def\mst@exists@skip@normal{0mu}
479 \def\mst@forall@skip@normal{0mu}
480 \def\mst@prime@skip@normal{0mu}
481 \def\mst@exists@skip@bold{0mu}
482 \def\mst@forall@skip@bold{0mu}
483 \def\mst@prime@skip@bold{0mu}
484 \else % not subdued
485 \ifmst@italic
486 \ifmst@frenchmath
487 \expandafter\mst@exists@muskip\mst@exists@skip\relax
488 \expandafter\mst@forall@muskip\mst@forall@skip\relax
489 \expandafter\mst@prime@muskip\mst@prime@skip\relax
490 \else
491 \def\mst@exists@skip@normal{0mu}
492 \def\mst@forall@skip@normal{0mu}
493 \def\mst@prime@skip@normal{0mu}
494 \def\mst@exists@skip@bold{0mu}
495 \def\mst@forall@skip@bold{0mu}
496 \def\mst@prime@skip@bold{0mu}
497 \fi
498 \else
499 \expandafter\mst@exists@muskip\mst@exists@skip\relax
500 \expandafter\mst@forall@muskip\mst@forall@skip\relax
501 \expandafter\mst@prime@muskip\mst@prime@skip\relax
502 \fi
503 \fi
504 %% v1.15f
505 \ifmst@nonnormalbold\else
506 \SetMathAlphabet{\mathnormalbold}{normal}{\mst@encoding@normal}
507 \{\mst@family@normal}
508 \{\mst@boldvariant@normal}
509 \{\mst@ltshape@normal}
510 \SetMathAlphabet{\mathnormalbold}{bold}{\mst@encoding@bold}
511 \{\mst@family@bold}
512 \{\mst@boldvariant@bold}
513 \{\mst@ltshape@bold}
514 \fi
515 %% v1.15f adds \ifmst@default.. checks
516 \ifmst@subdued\else

```

```

517 \SetSymbolFont{mtletterfont}{normal}{\mst@encoding@normal}
518             {\mst@family@normal}
519             {\mst@series@normal}
520             {\mst@ltshape@normal}
521 \SetSymbolFont{mtletterfont}{bold} {\mst@encoding@bold}
522             {\mst@family@bold}
523             {\mst@series@bold}
524             {\mst@ltshape@bold}
525 \SetSymbolFont{mtooperatorfont}{normal}{\mst@encoding@normal}
526             {\mst@family@normal}
527             {\mst@series@normal}
528             {\mst@shape@normal}
529 \SetSymbolFont{mtooperatorfont}{bold} {\mst@encoding@bold}
530             {\mst@family@bold}
531             {\mst@series@bold}
532             {\mst@shape@bold}
533 \ifmst@defaultbf\else
534 \SetMathAlphabet{\Mathbf}{normal}{\mst@encoding@normal}
535             {\mst@family@normal}
536             {\mst@series@bold}
537             {\mst@shape@normal}
538 \SetMathAlphabet{\Mathbf}{bold}{\mst@encoding@bold}
539             {\mst@family@bold}
540             {\mst@series@bold}
541             {\mst@shape@bold}
542 \fi\ifmst@defaultit\else
543 \SetMathAlphabet{\Mathit}{normal}{\mst@encoding@normal}
544             {\mst@family@normal}
545             {\mst@series@normal}
546             {\mst@itdefault@normal}
547 \SetMathAlphabet{\Mathit}{bold}{\mst@encoding@bold}
548             {\mst@family@bold}
549             {\mst@series@bold}
550             {\mst@itdefault@bold}
551 \fi\ifmst@defaultsf\else
552 \SetMathAlphabet{\Mathsf}{normal}{\mst@encoding@normal}
553             {\mst@sfdefault@normal}
554             {\mst@series@normal}
555             {\mst@shape@normal}
556 \SetMathAlphabet{\Mathsf}{bold}{\mst@encoding@bold}
557             {\mst@sfdefault@bold}
558             {\mst@series@bold}
559             {\mst@shape@bold}
560 \fi\ifmst@defaultttt\else
561 \SetMathAlphabet{\Mathtt}{normal}{\mst@encoding@normal}
562             {\mst@ttdefault@normal}
563             {\mst@series@normal}
564             {\mst@shape@normal}
565 \SetMathAlphabet{\Mathtt}{bold}{\mst@encoding@bold}

```

```

566                                     {\mst@ttdefault@bold}
567                                     {\mst@series@bold}
568                                     {\mst@shape@bold}
569 \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.

570 \ifmst@needeuler
571     \SetSymbolFont{mteulervm}{bold}{U}{zeur}{\mst@boldvariant@normal}{n}
572     \SetMathAlphabet{\MathEulerBold}{normal}
573         {U}{zeur}{\mst@boldvariant@normal}{n}
574     \SetMathAlphabet{\MathEulerBold}{bold}
575         {U}{zeur}{\mst@boldvariant@normal}{n}
576 \fi

577 \ifmst@needsymbol\SetSymbolFont{mtpsymbold}{bold}
578     {U}{psy}{\mst@boldvariant@normal}{n}
579 \fi

LGRgreek* LGRgreek, LGRgreek, selfGreek, and selfGreeks options.
selfGreek* 580 \ifmst@subdued\else
581 \ifmst@LGRgreek
582     \SetSymbolFont{mtlgrfontlower}{normal}{LGR}
583         {\mst@greekfont}{\mst@series@normal}{\mst@greek@lsh}
584     \SetSymbolFont{mtlgrfontlower}{bold}{LGR}
585         {\mst@greekfont}{\mst@boldvariant@normal}{\mst@greek@lsh}
586     \SetSymbolFont{mtlgrfontupper}{normal}{LGR}
587         {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
588     \SetSymbolFont{mtlgrfontupper}{bold}{LGR}
589         {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
590 \else
591 \ifmst@selfGreek
592     \SetSymbolFont{mtselfGreefont}{normal}{OT1}
593         {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
594     \SetSymbolFont{mtselfGreefont}{bold}{OT1}
595         {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
596 \fi\fi\fi

597 \ifmst@subdued
598 \typeout{** subdued mode will be activated for the normal and bold math versions}
599 \else
600 \typeout{** Latin letters in the normal (resp. bold) math versions are now^J%
601     ** set up to use the fonts
602 \mst@encoding@normal/\mst@family@normal/\mst@series@normal%
603 (\mst@boldvariant@normal)/\mst@ltshape@normal}
604 \ifmst@LGRgreek\typeout{** Greek letters (\mst@greek@lsh/\mst@greek@ush)
605 will use LGR/\mst@greekfont}
606 \fi

```

```

607 \ifmst@nodigits\else\typeout{** Other characters (digits, ...)
608 and \protect\log-like names will be^^J%
609 ** typeset with the \mst@shape@normal\space shape.}
610 \fi
611 \fi
612 \else % optional argument, then we rather do DeclareVersion
613 \edef\mst@tmp{#1}
614 \ifx\mst@tmp\@empty
615 \Mathastext % no infinite loop!
616 \else
617 \MTDeclareVersion[\mst@ltsh]{#1}
618 {\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}
619 \fi
620 \fi
621 }
622 \@onlypreamble\Mathastext
623 \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!

```

624 \newcommand*\MTDeclareVersion[6][\@empty]{
625 \edef\mst@version{#2}
626 \DeclareMathVersion{\mst@version}
627 \expandafter\edef\csname mst@encoding@\mst@version\endcsname{#3}
628 \expandafter\edef\csname mst@family@\mst@version\endcsname{#4}
629 \expandafter\edef\csname mst@series@\mst@version\endcsname{#5}
630 \expandafter\edef\csname mst@shape@\mst@version\endcsname{#6}
631 \expandafter\edef\csname mst@boldvariant@\mst@version\endcsname{\mst@bold}
632 \expandafter\edef\csname mst@itdefault@\mst@version\endcsname{\itdefault}
633 \expandafter\edef\csname mst@rmdefault@\mst@version\endcsname{\rmdefault}
634 \expandafter\edef\csname mst@sfdefault@\mst@version\endcsname{\sfdefault}
635 \expandafter\edef\csname mst@ttdefault@\mst@version\endcsname{\ttdefault}
636 \expandafter\edef\csname mst@exists@skip@\mst@version\endcsname
637 {\mst@exists@skip}
638 \expandafter\edef\csname mst@forall@skip@\mst@version\endcsname
639 {\mst@forall@skip}
640 \expandafter\edef\csname mst@prime@skip@\mst@version\endcsname
641 {\mst@prime@skip}
642 \mst@optionalparamfalse
643 \ifx\@empty#1
644 \else % we check nevertheless for [] situation, will not work with [ ]

```



```

645     \def\mst@tmp{#1}
646     \ifx\mst@tmp\@empty\else\mst@optionalparamtrue\fi
647     \fi
648     \ifmst@optionalparam
649         \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}
650         \typeout{** Latin letters in math version '#2' will use the font
651         #3/#4/#5/#1~J%
652         ** Other characters (digits, ...) and \protect\log-like
653         names will be in #6 shape.}
654         \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#1}
655     \else
656         \ifmst@italic
657             \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{\mst@ltsh}
658             \typeout{** Latin letters in math version '#2' will use the font
659             #3/#4/#5/\mst@ltsh~J%
660             ** Other characters (digits, ...) and \protect\log-like names
661             will be in #6 shape.}
662             \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{\mst@ltsh}
663         \else
664             \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#6}
665             \typeout{** Latin letters in math version '#2' will use the fonts
666             #3/#4/#5(\mst@bold)/#6}
667             \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#6}
668         \fi
669     \fi
670     \ifmst@nonormalbold\else
671         \SetMathAlphabet{\mathnormalbold}{#2}{#3}{#4}{\mst@bold}
672         \csname
673             mst@ltshape@\mst@version\endcsname}
674     \fi
675     \SetSymbolFont{mtooperatorfont}{#2}{#3}{#4}{#5}{#6}
676     \ifmst@defaultbf\else\SetMathAlphabet{\Mathbf}{#2}{#3}{#4}{\mst@bold}{#6}\fi
677     \ifmst@defaultit\else\SetMathAlphabet{\Mathit}{#2}{#3}{#4}{#5}{\itdefault}\fi
678     \ifmst@defaultsf\else\SetMathAlphabet{\Mathsf}{#2}{#3}{\sfdefault}{#5}{#6}\fi
679     \ifmst@defaultttt\else\SetMathAlphabet{\Mathtt}{#2}{#3}{\ttdefault}{#5}{#6}\fi
680     \ifmst@needeuler
681         \SetMathAlphabet{\MathEulerBold}{#2}{U}{\zeur}{\mst@bold}{n}
682     \fi

```

LGRgreek In the case of option LGRgreek (selfGreeks), it is expected that the fonts used in each math
selfGreeks 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.

```

683     \def\mst@greek@lsh@loc{\csname mst@ltshape@\mst@version\endcsname}
684     \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
685     \ifmst@itgreek\def\mst@greek@lsh@loc{\itdefault}
686         \def\mst@greek@ush@loc{\itdefault}\fi
687     \ifmst@upgreek\def\mst@greek@lsh@loc{\updefault}
688         \def\mst@greek@ush@loc{\updefault}\fi

```

```

689 \ifmst@frenchmath
690 \ifmst@itgreek\else
691 \ifmst@upgreek\else
692 \def\mst@greek@lsh@loc{\csname mst@shape@\mst@version\endcsname}
693 \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
694 \fi\fi
695 \fi
696 \ifcase\mst@greek@select
697 \or\def\mst@greek@ush@loc{\itdefault}
698 \or\def\mst@greek@ush@loc{\updefault}
699 \fi
700 \ifmst@LGRgreeks
701 \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{#4}{#5}{\mst@greek@lsh@loc}
702 \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{#4}{#5}{\mst@greek@ush@loc}
703 \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
704 LGR font #4}
705 \else
706 \ifmst@selfGreeks
707 \SetSymbolFont{mtselfGreefont}{#2}{OT1}{#4}{#5}{\mst@greek@ush@loc}
708 \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
709 will use OT1 font #4}
710 \else
711 \ifmst@LGRgreek
712 \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@lsh@loc}
713 \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
714 \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
715 LGR font \mst@greekfont}
716 \else
717 \ifmst@selfGreek
718 \SetSymbolFont{mtselfGreefont}{#2}{OT1}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
719 \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
720 will use OT1 font \mst@greekfont}
721 \fi\fi\fi\fi
722 %\mst@font@info@on
723 }
724 \@onlypreamble\MTDeclareVersion

725 \let\MathastextDeclareVersion\MTDeclareVersion

```

\MTVersion This is a wrapper around L^AT_EX'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...

```

726 \newcommand*{\MTVersion}[2][\@empty]{%
727   \mathversion{#2}%
728   \ifx\@empty#1% no optional argument
729     \def\mst@tmp{#2}%
730   \else% we check nevertheless for [] situation, will not work with [ ]
731     \def\mst@tmpa{#1}%
732     \ifx\mst@tmpa\@empty\def\mst@tmp{#2}%
733     \else\def\mst@tmp{#1}%
734     \fi%
735   \fi%
736   \usefont{\csname mst@encoding@\mst@tmp\endcsname}%
737     {\csname mst@family@\mst@tmp\endcsname}%
738     {\csname mst@series@\mst@tmp\endcsname}%
739     {\csname mst@shape@\mst@tmp\endcsname}%
740   \edef\mst@@encoding{\csname mst@encoding@\mst@tmp\endcsname}%
741   \renewcommand{\encodingdefault}{\mst@@encoding}%
742 %% some modifications for v1.15b
743   \edef\mst@@famdefault{\csname mst@family@\mst@tmp\endcsname}%
744   \renewcommand{\familydefault}{\mst@@famdefault}%
745   \edef\mst@@series{\csname mst@series@\mst@tmp\endcsname}%
746   \renewcommand{\mddefault}{\mst@@series}%
747   \edef\mst@@shape{\csname mst@shape@\mst@tmp\endcsname}%
748   \renewcommand{\shapedefault}{\mst@@shape}%
749   \edef\mst@@boldvariant{\csname mst@boldvariant@\mst@tmp\endcsname}%
750   \renewcommand{\bfdefault}{\mst@@boldvariant}%
751   \edef\mst@@itdefault{\csname mst@itdefault@\mst@tmp\endcsname}%
752   \renewcommand{\itdefault}{\mst@@itdefault}%
753   \edef\mst@@rmdefault{\csname mst@rmdefault@\mst@tmp\endcsname}%
754   \renewcommand{\rmdefault}{\mst@@rmdefault}%
755   \edef\mst@@sfdefault{\csname mst@sfdefault@\mst@tmp\endcsname}%
756   \renewcommand{\sfdefault}{\mst@@sfdefault}%
757   \edef\mst@@ttdefault{\csname mst@ttdefault@\mst@tmp\endcsname}%
758   \renewcommand{\ttdefault}{\mst@@ttdefault}%
759   \edef\mst@tmp{#2}%
760 %% v1.15e: muskips
761   \expandafter
762     \mst@exists@muskip\csname mst@exists@skip@\mst@tmp\endcsname\relax
763   \expandafter
764     \mst@forall@muskip\csname mst@forall@skip@\mst@tmp\endcsname\relax
765 %% v1.2: muskip for \cs{prime}
766   \expandafter
767     \mst@prime@muskip\csname mst@prime@skip@\mst@tmp\endcsname\relax
768 %% v1.2b: italic corrections except for italic/slanted (sic) letters, and of
769 %% course except in the subdued normal and bold math versions
770   \edef\mst@tmpa{\csname mst@ltshape@\mst@tmp\endcsname}%
771   \edef\mst@tmpb{\csname mst@shape@\mst@tmp\endcsname}%
772   \MTicinmath

```

```

773 \ifmst@frenchmath
774 \ifmst@frenchmath
775 \MTICinmath
776 \ifmst@LGRgreek
777 \fi
778 %% v1.15c: extending subdued to LGRgreek
779 %% v1.15f: subdueing math alphabets in a simpler way than in 1.15e
780 %% v1.2b: subdueing the activation of characters in math mode
781 \ifmst@subdued
782 \edef\mst@tmp{#2}\def\mst@tmpa{normal}%
783 \ifx\mst@tmp\mst@tmpa
784 \mst@restorealphabets
785 \MTmathoperatorsdonotobeymathxx
786 \MTnonlettersdonotobeymathxx
787 \MTnoicinmath
788 \ifmst@LGRgreek\mst@restoregreek\fi
789 \else
790 \def\mst@tmpa{bold}%
791 \ifx\mst@tmp\mst@tmpa
792 \mst@restorealphabets
793 \MTmathoperatorsdonotobeymathxx
794 \MTnonlettersdonotobeymathxx
795 \MTnoicinmath
796 \ifmst@LGRgreek\mst@restoregreek\fi
797 \else
798 \mst@setalphabets
799 \ifmst@LGRgreek\mst@setuplrgreek\fi
800 \fi\fi\fi
801 }
802 \let\MathastextVersion\MTVersion
803 \let\Mathastextversion\MTVersion
804 \let\MTversion\MTVersion
805 \let\mathastextversion\MTVersion

```

`\Mathastext` Initialization call:

```
806 \Mathastext
```

Additional appropriate messages to the terminal and the log.

```

807 \ifmst@eulergreek
808 \typeout{** Greek letters will use the Euler font. Use^^J%
809 ** \protect\MathastextEulerScale{<factor>} to scale the font.}
810 \else
811 \ifmst@symbolgreek
812 \typeout{** Greek letters will use the PostScript Symbol font. Use^^J%
813 ** \protect\MathastextSymbolScale{<factor>} to scale the font.}
814 \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.

```

815 \def\mst@DeclareMathSymbol#1#2#3#4{%
816     \DeclareMathSymbol{#1}{#2}{#3}{#4}
817     \expandafter\DeclareMathSymbol\csname mst@#1\endcsname{#2}{#3}{#4}
818     \expandafter\mst@addtodo@az
819     \expandafter#1\csname mst@#1\endcsname\mst@itcorr}
820 \mst@DeclareMathSymbol{a}{\mathalpha}{mtletterfont}{'a}
821 \mst@DeclareMathSymbol{b}{\mathalpha}{mtletterfont}{'b}
822 \mst@DeclareMathSymbol{c}{\mathalpha}{mtletterfont}{'c}
823 \mst@DeclareMathSymbol{d}{\mathalpha}{mtletterfont}{'d}
824 \mst@DeclareMathSymbol{e}{\mathalpha}{mtletterfont}{'e}
825 \mst@DeclareMathSymbol{f}{\mathalpha}{mtletterfont}{'f}
826 \mst@DeclareMathSymbol{g}{\mathalpha}{mtletterfont}{'g}
827 \mst@DeclareMathSymbol{h}{\mathalpha}{mtletterfont}{'h}
828 \mst@DeclareMathSymbol{i}{\mathalpha}{mtletterfont}{'i}
829 \mst@DeclareMathSymbol{j}{\mathalpha}{mtletterfont}{'j}
830 \mst@DeclareMathSymbol{k}{\mathalpha}{mtletterfont}{'k}
831 \mst@DeclareMathSymbol{l}{\mathalpha}{mtletterfont}{'l}
832 \mst@DeclareMathSymbol{m}{\mathalpha}{mtletterfont}{'m}
833 \mst@DeclareMathSymbol{n}{\mathalpha}{mtletterfont}{'n}
834 \mst@DeclareMathSymbol{o}{\mathalpha}{mtletterfont}{'o}
835 \mst@DeclareMathSymbol{p}{\mathalpha}{mtletterfont}{'p}
836 \mst@DeclareMathSymbol{q}{\mathalpha}{mtletterfont}{'q}
837 \mst@DeclareMathSymbol{r}{\mathalpha}{mtletterfont}{'r}
838 \mst@DeclareMathSymbol{s}{\mathalpha}{mtletterfont}{'s}
839 \mst@DeclareMathSymbol{t}{\mathalpha}{mtletterfont}{'t}
840 \mst@DeclareMathSymbol{u}{\mathalpha}{mtletterfont}{'u}
841 \mst@DeclareMathSymbol{v}{\mathalpha}{mtletterfont}{'v}
842 \mst@DeclareMathSymbol{w}{\mathalpha}{mtletterfont}{'w}
843 \mst@DeclareMathSymbol{x}{\mathalpha}{mtletterfont}{'x}
844 \mst@DeclareMathSymbol{y}{\mathalpha}{mtletterfont}{'y}
845 \mst@DeclareMathSymbol{z}{\mathalpha}{mtletterfont}{'z}
846 \ifmst@frenchmath\def\mst@font@tbu{mtoperatorfont}
847     \else\def\mst@font@tbu{mtletterfont}\fi
848 \def\mst@DeclareMathSymbol#1#2#3#4{%
849     \DeclareMathSymbol{#1}{#2}{#3}{#4}
850     \expandafter\DeclareMathSymbol\csname mst@#1\endcsname{#2}{#3}{#4}
851     \expandafter\mst@addtodo@AZ
852     \expandafter#1\csname mst@#1\endcsname\mst@ITcorr}
853 \mst@DeclareMathSymbol{A}{\mathalpha}{\mst@font@tbu}{'A}
854 \mst@DeclareMathSymbol{B}{\mathalpha}{\mst@font@tbu}{'B}
855 \mst@DeclareMathSymbol{C}{\mathalpha}{\mst@font@tbu}{'C}
856 \mst@DeclareMathSymbol{D}{\mathalpha}{\mst@font@tbu}{'D}
857 \mst@DeclareMathSymbol{E}{\mathalpha}{\mst@font@tbu}{'E}
858 \mst@DeclareMathSymbol{F}{\mathalpha}{\mst@font@tbu}{'F}
859 \mst@DeclareMathSymbol{G}{\mathalpha}{\mst@font@tbu}{'G}
860 \mst@DeclareMathSymbol{H}{\mathalpha}{\mst@font@tbu}{'H}
861 \mst@DeclareMathSymbol{I}{\mathalpha}{\mst@font@tbu}{'I}
862 \mst@DeclareMathSymbol{J}{\mathalpha}{\mst@font@tbu}{'J}
863 \mst@DeclareMathSymbol{K}{\mathalpha}{\mst@font@tbu}{'K}

```

```

864 \mst@DeclareMathSymbol{L}{\mathalpha}{\mst@font@tbu}{‘L}
865 \mst@DeclareMathSymbol{M}{\mathalpha}{\mst@font@tbu}{‘M}
866 \mst@DeclareMathSymbol{N}{\mathalpha}{\mst@font@tbu}{‘N}
867 \mst@DeclareMathSymbol{O}{\mathalpha}{\mst@font@tbu}{‘O}
868 \mst@DeclareMathSymbol{P}{\mathalpha}{\mst@font@tbu}{‘P}
869 \mst@DeclareMathSymbol{Q}{\mathalpha}{\mst@font@tbu}{‘Q}
870 \mst@DeclareMathSymbol{R}{\mathalpha}{\mst@font@tbu}{‘R}
871 \mst@DeclareMathSymbol{S}{\mathalpha}{\mst@font@tbu}{‘S}
872 \mst@DeclareMathSymbol{T}{\mathalpha}{\mst@font@tbu}{‘T}
873 \mst@DeclareMathSymbol{U}{\mathalpha}{\mst@font@tbu}{‘U}
874 \mst@DeclareMathSymbol{V}{\mathalpha}{\mst@font@tbu}{‘V}
875 \mst@DeclareMathSymbol{W}{\mathalpha}{\mst@font@tbu}{‘W}
876 \mst@DeclareMathSymbol{X}{\mathalpha}{\mst@font@tbu}{‘X}
877 \mst@DeclareMathSymbol{Y}{\mathalpha}{\mst@font@tbu}{‘Y}
878 \mst@DeclareMathSymbol{Z}{\mathalpha}{\mst@font@tbu}{‘Z}
879 %%
880 \ifmst@nodigits\else
881 \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.

```

882 \ifmst@symboldigits \def\mst@font@tbu{mtpsymbol} \fi
883 \ifmst@eulerdigits \def\mst@font@tbu{mteulervm} \fi
884 \DeclareMathSymbol{0}{\mathalpha}{\mst@font@tbu}{‘0}
885 \DeclareMathSymbol{1}{\mathalpha}{\mst@font@tbu}{‘1}
886 \DeclareMathSymbol{2}{\mathalpha}{\mst@font@tbu}{‘2}
887 \DeclareMathSymbol{3}{\mathalpha}{\mst@font@tbu}{‘3}
888 \DeclareMathSymbol{4}{\mathalpha}{\mst@font@tbu}{‘4}
889 \DeclareMathSymbol{5}{\mathalpha}{\mst@font@tbu}{‘5}
890 \DeclareMathSymbol{6}{\mathalpha}{\mst@font@tbu}{‘6}
891 \DeclareMathSymbol{7}{\mathalpha}{\mst@font@tbu}{‘7}
892 \DeclareMathSymbol{8}{\mathalpha}{\mst@font@tbu}{‘8}
893 \DeclareMathSymbol{9}{\mathalpha}{\mst@font@tbu}{‘9}
894 \fi

```

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

```

895 \ifmst@symboldelimiters
896 \def\mst@font@tbu{mtpsymbol}
897 \mst@endashfalse
898 \mst@emdashfalse
899 \else
900 \def\mst@font@tbu{mtoperatorfont}
901 \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`.

```

902 \ifmst@noexclam\else
903 \DeclareMathSymbol{!}{\mathclose}{\mst@font@tbu}{"21}
904 \DeclareMathSymbol{\mst@alpha@exclam}{\mathalpha}{\mst@font@tbu}{"21}

```



```

941 \else
942 \ifLuaTeX
943 %% \luatexUmathcharnumdef\mst@minus@sign=\luatexUmathcodenum'\-\relax ne
944 %% marche pas pour le moment car \luatexUmathcodenum retourne parfois un
945 %% legacy mathcode, parfois un Unicode mathcode, \a mon avis c'est un bug
946 %% de LuaLatex. En fait Taco H me dit que la prochaine version
947 %% r\eglera ce probl\eme. \luatexUmathchardef\mst@minus@sign="2 "2 "0\relax
948 \mathchardef\mst@minus@sign=8704\relax % "2200
949 \else
950 \mathchardef\mst@minus@sign=\mathcode'\-\relax
951 \fi\fi
952 \@ifpackageloaded{amsmath}
953 {\def\relbar{\mathrel{\mathpalette\mathsm@sh\mst@minus@sign}}}
954 {\DeclareRobustCommand\relbar{\mathrel{\smash\mst@minus@sign}}}
955 \def\rightarrowfill{${\m@th\mathord{\relbar}}\mkern-7mu%
956 \cleaders\hbox{${\mkern-2mu\relbar\mkern-2mu}$}\hfill
957 \mkern-7mu\mathord{\rightarrow}$}
958 \def\leftarrowfill{${\m@th\mathord{\leftarrow}}\mkern-7mu%
959 \cleaders\hbox{${\mkern-2mu\relbar\mkern-2mu}$}\hfill
960 \mkern-7mu\mathord{\leftarrow}$}
961 \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.

```

962 \def\do@the@endashstuff{\edef\mst@tmp@enc{\encodingdefault}
963 \ifx\mst@tmp@enc\mst@eui % xetex+fontspec
964 \AtBeginDocument{\XeTeXmathcode'\-="2 \symmoperatorfont "2013\relax}
965 \AtBeginDocument{\XeTeXmathchardef\mst@alpha@minus="7 \symmoperatorfont "2013\re-
lax}
966 \else
967 \ifx\mst@tmp@enc\mst@euii % luatex+fontspec
968 \AtBeginDocument{\luatexUmathcode'\-="2 \symmoperatorfont "2013\relax}
969 \AtBeginDocument{\luatexUmathchardef\mst@alpha@minus="7 \symmoperatorfont "2013\re-
lax}
970 \else
971 \ifx\mst@tmp@enc\mst@ti % T1
972 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{21}
973 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{21}
974 \else
975 \ifx\mst@tmp@enc\mst@oti % OT1
976 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{123}
977 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{123}
978 \else

```



```

979 \ifx\mst@tmp@enc\mst@lyi % LY1
980 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{150}
981 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{150}
982 \else % make sure we have neither xetex nor luatex
983 \ifXeTeX\mst@aborttrue\else\ifLuaTeX\mst@aborttrue\else
984 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}
985 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}
986 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\textendash\endcsname}
987 \fi\fi
988 \fi\fi\fi\fi\fi}
989 \def\do@the@emdashstuff{\edef\mst@tmp@enc{\encodingdefault}
990 \ifx\mst@tmp@enc\mst@eui % xetex+fontspec
991 \AtBeginDocument{\XeTeXmathcode'\-="2 \symmoperatorfont "2014\relax}
992 \AtBeginDocument{\XeTeXmathchardef\mst@alpha@minus="7 \symmoperatorfont "2014\re-
lax}
993 \else
994 \ifx\mst@tmp@enc\mst@euii % luatex+fontspec
995 \AtBeginDocument{\luatexUmathcode'\-="2 \symmoperatorfont "2014\relax}
996 \AtBeginDocument{\luatexUmathchardef\mst@alpha@minus="7 \symmoperatorfont "2014\re-
lax}
997 \else
998 \ifx\mst@tmp@enc\mst@ti % T1
999 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{22}
1000 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{22}
1001 \else
1002 \ifx\mst@tmp@enc\mst@oti % OT1
1003 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{124}
1004 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{124}
1005 \else
1006 \ifx\mst@tmp@enc\mst@lyi % LY1
1007 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{151}
1008 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}{151}
1009 \else % make sure we have neither xetex nor luatex
1010 \ifXeTeX\mst@aborttrue\else\ifLuaTeX\mst@aborttrue\else
1011 \DeclareMathSymbol{-}{\mathbin}{moperatorfont}
1012 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{moperatorfont}
1013 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\textemdash\endcsname}
1014 \fi\fi
1015 \fi\fi\fi\fi\fi}
1016 \ifmst@nominus\else
1017 \mst@abortfalse
1018 \ifmst@endash\do@the@emdashstuff\else
1019 \ifmst@emdash\do@the@emdashstuff\else
1020 \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{2D}
1021 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{\mst@font@tbu}{2D}
1022 \fi\fi
1023 \ifmst@abort
1024 \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{2D}
1025 \DeclareMathSymbol{\mst@alpha@minus}{\mathalpha}{\mst@font@tbu}{2D}

```

```

1026 \fi
1027 \mst@addtodo@nonletters-\mathbin\mst@alpha@minus
1028 \fi

```

`\hbar` 2011/01/31, 1.1 I decide to settle the question of the `\hbar`. First, the L^AT_EX definition

```

\mst@ltbar 1029 %%\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...)

```

1030 %
1031 \def\mst@mathrlap{\mathpalette\mst@mathrlapinternal}
1032 \def\mst@mathrlapinternal#1#2{\rlap{$\mathsurround=0pt#1{#2}$}}
1033 \def\do@the@hbarstuff{\edef\mst@tmp@enc{\encodingdefault}
1034 \ifx\mst@tmp@enc\mst@eui % xetex and unicode font
1035 \XeTeXmathchardef\hbar="7 \symmtletterfont "0127 \relax
1036 \else
1037 \ifx\mst@tmp@enc\mst@euii % luatex and unicode font
1038 \luatexumathchardef\hbar="7 \symmtletterfont "0127 \relax %% or 210F?
1039 \else % I must leave open the T1, OT1 possibility also for XeTeX or LuaTeX
1040 \ifx\mst@ti\mst@tmp@enc
1041 \DeclareMathAccent{\mst@ltbar}{\mathalpha}{mtletterfont}{9}
1042 \else %% assume OT1. Bad idea? Should not provoke an error anyhow
1043 \DeclareMathAccent{\mst@ltbar}{\mathalpha}{mtletterfont}{22}
1044 \fi
1045 \def\hbar{\mst@mathrlap{\mst@ltbar{\ }h}}
1046 \fi\fi}
1047 \ifmst@nohbar\else\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 T_EXpert.) 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`

```

1048 \ifmst@noplus\else
1049 \DeclareMathSymbol{+}{\mathbin}{\mst@font@tbu}{"2B}
1050 \DeclareMathSymbol{\mst@alpha@plus}{\mathalpha}{\mst@font@tbu}{"2B}
1051 \mst@addtodo@nonletters+\mathbin\mst@alpha@plus

```

```

1052 \fi
1053 \ifmst@noequal\else
1054 \ifXeTeX
1055 \XeTeXmathcharnumdef\mst@equal@sign=\XeTeXmathcodenum'\=\relax
1056 \else
1057 \ifLuaTeX
1058 %% 12 octobre 2012 \luatexUmathcodenum'\=
1059 %% ne marche pas pour le moment \'a cause du fait que \luatexUmathcodenum
1060 %% donne un mathcode sur 8bit ou un Unicode mathcode sans qu'on puisse
1061 %% savoir lequel
1062 %% \luatexUmathchardef\mst@equal@sign="3 "0 "3D\relax
1063 \mathchardef\mst@equal@sign=12349\relax
1064 \else
1065 \mathchardef\mst@equal@sign=\mathcode'\=\relax
1066 \fi\fi
1067 \@ifpackageloaded{amsmath}
1068 {\def\Relbar{\mathrel\mst@equal@sign}}
1069 {\DeclareRobustCommand\Relbar{\mathrel{\mst@equal@sign}}}
1070 \DeclareMathSymbol{=}{\mathrel}{\mst@font@tbu}{"3D}
1071 \DeclareMathSymbol{\mst@alpha@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: In fact this = ends up in \csname...\endcsname. I have discovered that if it is not active but only has its mathcode set to "8000 it is not expanded. Try:

```

% \mathcode'x="8000
% \begingroup
% \catcode'x=\active
% \global\everymath{\defx{Hello}}
% \endgroup
% \def\foox{World!}
% $x \csname foom\endcsname$
%

```

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

```

1072 \g@addto@macro\nfss@catcodes{%
1073   \makeother\=%
1074   \makeother\;%
1075   \makeother\+%
1076 }
1077 \mst@addtodo@nonletters=\mathrel\mst@alpha@equal
1078 \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`.

```

1079 \ifmst@noparen\else
1080 \ifmst@nosmallldelims
1081     \DeclareMathSymbol{\{}{\mathopen}{\mst@font@tbu}{28}
1082     \DeclareMathSymbol{\}}{\mathclose}{\mst@font@tbu}{29}
1083     \DeclareMathSymbol{\lbrack}{\mathopen}{\mst@font@tbu}{5B}
1084     \DeclareMathSymbol{\rbrack}{\mathclose}{\mst@font@tbu}{5D}
1085     \DeclareMathSymbol{\}/}{\mathord}{\mst@font@tbu}{2F}
1086 \else
1087     \DeclareMathDelimiter{\{}{\mathopen}{\mst@font@tbu}{28}{largesymbols}{00}
1088     \DeclareMathDelimiter{\}}{\mathclose}{\mst@font@tbu}{29}{largesymbols}{01}
1089     \DeclareMathDelimiter{\lbrack}{\mathopen}{\mst@font@tbu}{5B}{largesymbols}{02}
1090     \DeclareMathDelimiter{\rbrack}{\mathclose}{\mst@font@tbu}{5D}{largesymbols}{03}
1091     \DeclareMathDelimiter{\}/}{\mathord}{\mst@font@tbu}{2F}{largesymbols}{0E}
1092 \fi
1093 \DeclareMathSymbol{\mst@alpha@lparen}{\mathalpha}{\mst@font@tbu}{40}
1094 \DeclareMathSymbol{\mst@alpha@rparen}{\mathalpha}{\mst@font@tbu}{41}
1095 \DeclareMathSymbol{\mst@alpha@lbrack}{\mathalpha}{\mst@font@tbu}{5B}
1096 \DeclareMathSymbol{\mst@alpha@rbrack}{\mathalpha}{\mst@font@tbu}{5D}
1097 \DeclareMathSymbol{\mst@alpha@slash}{\mathalpha}{\mst@font@tbu}{2F}
1098 \mst@addtodo@nonletters{\mathopen\mst@alpha@lparen
1099 \mst@addtodo@nonletters}\mathclose\mst@alpha@rparen
1100 \mst@addtodo@nonletters[\mathopen\mst@alpha@lbrack
1101 \mst@addtodo@nonletters]\mathclose\mst@alpha@rbrack
1102 \mst@addtodo@nonletters/\mathord\mst@alpha@slash
1103 \fi

```

alldelims

 $\langle, \rangle, \backslash$ 1104 \ifmst@alldelims

```
\setminusifmst@nosmalldelims
```

Dec 18, 2012. We then want `\let\backslashmst@alpha@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`.

```

1106 \DeclareMathDelimiter{\mst@alpha@backslash}
1107     {\mathalpha}{symbols}{"6E}{largesymbols}{"0F}
1108 \else
1109     \DeclareMathDelimiter{<}{\mathopen}{\mst@font@tbu}{"3C}{largesymbols}{"0A}
1110     \DeclareMathDelimiter{>}{\mathclose}{\mst@font@tbu}{"3E}{largesymbols}{"0B}

```

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

```

1111 \expandafter\DeclareMathDelimiter\@backslashchar
1112     {\mathord}{mtopoperatorfont}{\string"5C}{largesymbols}{\string"0F}
1113 \DeclareMathDelimiter{\backslash}
1114     {\mathord}{mtopoperatorfont}{\string"5C}{largesymbols}{\string"0F}

```

```

1115 \DeclareMathDelimiter{\mst@alpha@backslash}
1116 {\mathalpha}{mtoperatorfont}{5C}{largesymbols}{0F}
1117 \fi
1118 \DeclareMathSymbol{<}{\mathrel}{\mst@font@tbu}{3C}
1119 \DeclareMathSymbol{>}{\mathrel}{\mst@font@tbu}{3E}
1120 \DeclareMathSymbol{\mst@alpha@less}{\mathalpha}{\mst@font@tbu}{3C}
1121 \DeclareMathSymbol{\mst@alpha@more}{\mathalpha}{\mst@font@tbu}{3E}
1122 \mst@addtodo@nonletters<\mathrel\mst@alpha@less
1123 \mst@addtodo@nonletters>\mathrel\mst@alpha@more
1124 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1125 \let\backslash\mst@alpha@backslash}
1126 \DeclareMathSymbol{\setminus}{\mathbin}{mtoperatorfont}{5C}
1127 \DeclareMathSymbol{\mst@alpha@setminus}{\mathalpha}{mtoperatorfont}{5C}
1128 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1129 \def\setminus{\mathbin{\mst@alpha@setminus}}}

```

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

```

1130 \ifXeTeX
1131 \XeTeXmathcharnumdef\mst@vert@bar=\XeTeXmathcodenum'\|\relax
1132 \else
1133 \ifLuaTeX
1134 %% \luatexUmathchardef\mst@vert@bar="0 "2 "006A\relax
1135 \mathchardef\mst@vert@bar=618\relax
1136 \else\mathchardef\mst@vert@bar=\mathcode'\|\relax
1137 \fi\fi
1138 \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.

```

1139 \ifmst@nosmallldelims
1140 \DeclareMathSymbol{||}{\mathord}{\mst@font@tbu}{124}
1141 \else
1142 \DeclareMathDelimiter{||}{\mathord}{\mst@font@tbu}{124}{largesymbols}{0C}
1143 \fi
1144 \def\vert{||}
1145 \DeclareMathSymbol{\mst@alpha@vbar}{\mathalpha}{\mst@font@tbu}{124}
1146 \mst@addtodo@nonletters|\mathord\mst@alpha@vbar
1147 \DeclareMathSymbol{\mid}{\mathrel}{\mst@font@tbu}{124}
1148 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1149 \def\mid{\mathrel\mst@alpha@vbar}}

```

\MTeXplicitbraces- 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
obeymathxx
\MTeXplicitbraces- in 1.2a, to preserve robustness.
donotobeymathxx

```

1150 \ifmst@nosmallldelims
1151 \DeclareMathSymbol{\lbrace}{\mathopen}{\mst@font@tbu}{123}
1152 \DeclareMathSymbol{\rbrace}{\mathclose}{\mst@font@tbu}{125}
1153 \else
1154 \DeclareMathDelimiter{\lbrace}

```

```

1155      {\mathopen}{\mst@font@tbu}{123}{largesymbols}{"08}
1156 \DeclareMathDelimiter{\rbrace}
1157      {\mathclose}{\mst@font@tbu}{125}{largesymbols}{"09}
1158 \fi
1159 \DeclareMathSymbol{\mst@alpha@lbrace}{\mathalpha}{\mst@font@tbu}{123}
1160 \DeclareMathSymbol{\mst@alpha@rbrace}{\mathalpha}{\mst@font@tbu}{125}
1161 \DeclareRobustCommand{\mst@lbrace}
1162      {\ifmmode\mathopen\mst@alpha@lbrace\else\textbraceleft\fi}
1163 \DeclareRobustCommand{\mst@rbrace}
1164      {\ifmmode\mathclose\mst@alpha@rbrace\else\textbraceright\fi}
1165 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1166   \mst@dobraces{\let\{\mst@lbrace\let\}\mst@rbrace}}
1167 \fi % end of \ifmst@alldelims
1168 \newcommand{\MTexplicitbracesobeymathxx}{\let\mst@dobraces\@firstofone}
1169 \newcommand{\MTexplicitbracesdonotobeymathxx}{\let\mst@dobraces\@gobble}
1170 \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 a bit insane and had definitions which could surprise other packages. I (finally) correct this: I had been told that my earlier definitions caused a problem with microtype...

1.15g (2012/10/25): ... but it was *not* that insane, I just had forgotten the reason I did it that way! I just extended to the other specials what latex was doing with \\$ (\mathdollar!)...

```

1171 \ifmst@nospecials\else
1172 \ifXeTeX
1173   \XeTeXmathcode'\#=7 \symmoperatorfont "0023 \relax
1174   \XeTeXmathchardef\mathdollar=7 \symmoperatorfont "0024 \relax
1175   \XeTeXmathcode'\%=7 \symmoperatorfont "0025 \relax
1176   \XeTeXmathcode'\&=7 \symmoperatorfont "0026 \relax
1177 \else
1178 \ifLuaTeX
1179   \luatexUmathcode'\#=7 \symmoperatorfont "0023 \relax
1180   \luatexUmathchardef\mathdollar=7 \symmoperatorfont "0024 \relax
1181   \luatexUmathcode'\%=7 \symmoperatorfont "0025 \relax
1182   \luatexUmathcode'\&=7 \symmoperatorfont "0026 \relax
1183 \else
1184 \count255=\symmoperatorfont
1185 \multiply\count255 by 256
1186 \advance\count255 by 28707 % "7023
1187   \mathcode'\#\count255
1188 \advance\count255 by 1
1189   \mathchardef\mathdollar\count255
1190 \advance\count255 by 1
1191   \mathcode'\%\count255
1192 \advance\count255 by 1
1193   \mathcode'\&\count255
1194 \fi\fi\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 L^AT_EX, 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` 1.13b: I correct the brutal re-definitions of `\prod` and `\sum` from the earlier versions of the
`\sum` 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?

```

1195 \ifmst@symbolmisc
1196 \let\mst@prod\prod
1197 \let\defaultprod\prod
1198 \DeclareMathSymbol{\prodpsy}{\mathop}{mtpsymbol}{213}
1199 \renewcommand{\prod}{\ifinner\prodpsy\else\mst@prod\fi}
1200 \let\mst@sum\sum
1201 \let\defaultsum\sum
1202 \DeclareMathSymbol{\sumpsy}{\mathop}{mtpsymbol}{229}
1203 \renewcommand{\sum}{\ifinner\sumpsy\else\mst@sum\fi}

1204 \DeclareMathSymbol{\mst@implies}{\mathrel}{mtpsymbol}{222}
1205 \DeclareRobustCommand{\implies}{\;\;\mst@implies\;}
1206 \DeclareMathSymbol{\mst@impliedby}{\mathrel}{mtpsymbol}{220}
1207 \DeclareRobustCommand{\impliedby}{\;\;\mst@impliedby\;}
1208 \DeclareRobustCommand{\iff}{\;\;\mst@impliedby\mathrel{\mkern-3mu}\mst@implies\;}
1209 \DeclareMathSymbol{\mst@iff}{\mathrel}{mtpsymbol}{219}
1210 \DeclareRobustCommand{\shortiff}{\;\;\mst@iff\;}
1211 \DeclareMathSymbol{\mst@to}{\mathrel}{mtpsymbol}{174}
1212 \DeclareMathSymbol{\mst@trait}{\mathrel}{mtpsymbol}{190}
1213 \DeclareRobustCommand{\to}{\mst@to}
1214 \DeclareRobustCommand{\longto}{\mkern2mu\mst@trait\mathrel{\mkern-10mu}\mst@to}
1215 \DeclareRobustCommand{\mapsto}{\mapstochar\mathrel{\mkern0.2mu}\mst@to}
1216 \DeclareRobustCommand{\longmapsto}{%
1217 \mapstochar\mathrel{\mkern2mu}\mst@trait\mathrel{\mkern-10mu}\mst@to}
1218 \DeclareMathSymbol{\aleph}{\mathord}{mtpsymbol}{192}
1219 \DeclareMathSymbol{\infty}{\mathord}{mtpsymbol}{165}
1220 \DeclareMathSymbol{\emptyset}{\mathord}{mtpsymbol}{198}
1221 \let\varnothing\emptyset
1222 \DeclareMathSymbol{\nabla}{\mathord}{mtpsymbol}{209}
1223 \DeclareMathSymbol{\surd}{\mathop}{mtpsymbol}{214}
1224 \let\angle\undefined
1225 \DeclareMathSymbol{\angle}{\mathord}{mtpsymbol}{208}
1226 \DeclareMathSymbol{\forall}{\mathord}{mtpsymbol}{34}
1227 \DeclareMathSymbol{\exists}{\mathord}{mtpsymbol}{36}
1228 \DeclareMathSymbol{\neg}{\mathord}{mtpsymbol}{216}
1229 \DeclareMathSymbol{\clubsuit}{\mathord}{mtpsymbol}{167}

```

```

1230 \DeclareMathSymbol{\diamondsuit}{\mathord}{mtpsymbol}{168}
1231 \DeclareMathSymbol{\heartsuit}{\mathord}{mtpsymbol}{169}
1232 \DeclareMathSymbol{\spadesuit}{\mathord}{mtpsymbol}{170}
1233 \DeclareMathSymbol{\smallint}{\mathop}{mtpsymbol}{242}
1234 \DeclareMathSymbol{\wedge}{\mathbin}{mtpsymbol}{217}
1235 \DeclareMathSymbol{\vee}{\mathbin}{mtpsymbol}{218}
1236 \DeclareMathSymbol{\cap}{\mathbin}{mtpsymbol}{199}
1237 \DeclareMathSymbol{\cup}{\mathbin}{mtpsymbol}{200}
1238 \DeclareMathSymbol{\bullet}{\mathbin}{mtpsymbol}{183}
1239 \DeclareMathSymbol{\div}{\mathbin}{mtpsymbol}{184}
1240 \DeclareMathSymbol{\otimes}{\mathbin}{mtpsymbol}{196}
1241 \DeclareMathSymbol{\oplus}{\mathbin}{mtpsymbol}{197}
1242 \DeclareMathSymbol{\pm}{\mathbin}{mtpsymbol}{177}
1243 %%\DeclareMathSymbol{*}{\mathbin}{mtpsymbol}{42}
1244 \DeclareMathSymbol{\ast}{\mathbin}{mtpsymbol}{42}
1245 \DeclareMathSymbol{\times}{\mathbin}{mtpsymbol}{180}
1246 \DeclareMathSymbol{\proptopsy}{\mathrel}{mtpsymbol}{181}
1247 \DeclareMathSymbol{\mid}{\mathrel}{mtpsymbol}{124}
1248 \DeclareMathSymbol{\leq}{\mathrel}{mtpsymbol}{163}
1249 \DeclareMathSymbol{\geq}{\mathrel}{mtpsymbol}{179}
1250 \DeclareMathSymbol{\approx}{\mathrel}{mtpsymbol}{187}
1251 \DeclareMathSymbol{\supset}{\mathrel}{mtpsymbol}{201}
1252 \DeclareMathSymbol{\subset}{\mathrel}{mtpsymbol}{204}
1253 \DeclareMathSymbol{\supseteq}{\mathrel}{mtpsymbol}{202}
1254 \DeclareMathSymbol{\subseteq}{\mathrel}{mtpsymbol}{205}
1255 \DeclareMathSymbol{\in}{\mathrel}{mtpsymbol}{206}
1256 \DeclareMathSymbol{\sim}{\mathrel}{mtpsymbol}{126}
1257 \let\cong\undefined
1258 \DeclareMathSymbol{\cong}{\mathrel}{mtpsymbol}{64}
1259 \DeclareMathSymbol{\perp}{\mathrel}{mtpsymbol}{94}
1260 \DeclareMathSymbol{\equiv}{\mathrel}{mtpsymbol}{186}
1261 \let\notin\undefined
1262 \DeclareMathSymbol{\notin}{\mathrel}{mtpsymbol}{207}
1263 \DeclareMathDelimiter{\rangle}
1264   {\mathclose}{mtpsymbol}{241}{largesymbols}{"0B}
1265 \DeclareMathDelimiter{\langle}
1266   {\mathopen}{mtpsymbol}{225}{largesymbols}{"0A}
1267 \fi

```

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

```

1268 \ifmst@symbolre
1269 \DeclareMathSymbol{\Re}{\mathord}{mtpsymbol}{C2}
1270 \DeclareMathSymbol{\Im}{\mathord}{mtpsymbol}{C1}
1271 \DeclareMathSymbol{\DotTriangle}{\mathord}{mtpsymbol}{92}
1272 \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.

```
1273 \let\mst@mathord\mathalpha
1274 \mst@goaheadtrue
1275 \ifmst@selfGreek
1276     \def\mst@font@tbu{mtselfGreekfont}
1277 \else
1278 \ifmst@eulergreek
1279     \def\mst@font@tbu{mteulervm}
1280 \else
1281 \ifmst@symbolgreek
1282     \def\mst@font@tbu{mtpsymbol}
1283     \let\mst@mathord\mathord
1284 \else
1285 \ifmst@LGRgreek
1286     \mst@goaheadfalse
1287 \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.

```
1288     \mst@goaheadfalse
1289     \ifx\omicron\undefined
1290     \newcommand{\omicron}{\mst@alph@omicron{o}}
1291     \fi
1292 \fi\fi\fi\fi
1293 \ifmst@goahead
1294 \DeclareMathSymbol{\Alpha}{\mst@mathord}{\mst@font@tbu}{41}
1295 \DeclareMathSymbol{\Beta}{\mst@mathord}{\mst@font@tbu}{42}
1296 \DeclareMathSymbol{\Epsilon}{\mst@mathord}{\mst@font@tbu}{45}
1297 \DeclareMathSymbol{\Zeta}{\mst@mathord}{\mst@font@tbu}{5A}
1298 \DeclareMathSymbol{\Eta}{\mst@mathord}{\mst@font@tbu}{48}
1299 \DeclareMathSymbol{\Iota}{\mst@mathord}{\mst@font@tbu}{49}
1300 \DeclareMathSymbol{\Kappa}{\mst@mathord}{\mst@font@tbu}{4B}
1301 \DeclareMathSymbol{\Mu}{\mst@mathord}{\mst@font@tbu}{4D}
1302 \DeclareMathSymbol{\Nu}{\mst@mathord}{\mst@font@tbu}{4E}
1303 \DeclareMathSymbol{\Omicron}{\mst@mathord}{\mst@font@tbu}{4F}
1304 \DeclareMathSymbol{\Rho}{\mst@mathord}{\mst@font@tbu}{50}
1305 \DeclareMathSymbol{\Tau}{\mst@mathord}{\mst@font@tbu}{54}
1306 \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 Φ).

```

1307 \ifx\mst@mathord\mathord
      symbolgreek but neither eulergreek nor selfGreek
1308 %% attention le P de Symbol est un \Pi pas un \Rho
1309 \DeclareMathSymbol{\Rho}{\mathord}{mtpsymbol}{52}
1310 %% attention le X de Symbol est un \Xi pas un \Chi
1311 \DeclareMathSymbol{\Chi}{\mathord}{mtpsymbol}{43}
1312 %% attention le F de Symbol est un \Phi. Il n'y a pas de \Digamma
1313 \DeclareMathSymbol{\Gamma}{\mathord}{mtpsymbol}{47}
1314 \DeclareMathSymbol{\Delta}{\mathord}{mtpsymbol}{44}
1315 \DeclareMathSymbol{\Theta}{\mathord}{mtpsymbol}{51}
1316 \DeclareMathSymbol{\Lambda}{\mathord}{mtpsymbol}{4C}
1317 \DeclareMathSymbol{\Xi}{\mathord}{mtpsymbol}{58}
1318 \DeclareMathSymbol{\Pi}{\mathord}{mtpsymbol}{50}
1319 \DeclareMathSymbol{\Sigma}{\mathord}{mtpsymbol}{53}
1320 \DeclareMathSymbol{\Upsilon}{\mathord}{mtpsymbol}{A1}
1321 \DeclareMathSymbol{\Phi}{\mathord}{mtpsymbol}{46}
1322 \DeclareMathSymbol{\Psi}{\mathord}{mtpsymbol}{59}
1323 \DeclareMathSymbol{\Omega}{\mathord}{mtpsymbol}{57}
1324 \else
      not symbolgreek but eulergreek or selfGreek or both
1325 \DeclareMathSymbol{\Digamma}{\mathalpha}{\mst@font@tbu}{46}
1326 \DeclareMathSymbol{\Gamma}{\mathalpha}{\mst@font@tbu}{00}
1327 \DeclareMathSymbol{\Delta}{\mathalpha}{\mst@font@tbu}{01}
1328 \DeclareMathSymbol{\Theta}{\mathalpha}{\mst@font@tbu}{02}
1329 \DeclareMathSymbol{\Lambda}{\mathalpha}{\mst@font@tbu}{03}
1330 \DeclareMathSymbol{\Xi}{\mathalpha}{\mst@font@tbu}{04}
1331 \DeclareMathSymbol{\Pi}{\mathalpha}{\mst@font@tbu}{05}
1332 \DeclareMathSymbol{\Sigma}{\mathalpha}{\mst@font@tbu}{06}
1333 \DeclareMathSymbol{\Upsilon}{\mathalpha}{\mst@font@tbu}{07}
1334 \DeclareMathSymbol{\Phi}{\mathalpha}{\mst@font@tbu}{08}
1335 \DeclareMathSymbol{\Psi}{\mathalpha}{\mst@font@tbu}{09}
1336 \DeclareMathSymbol{\Omega}{\mathalpha}{\mst@font@tbu}{0A}
1337 \fi
1338 \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 L^AT_EX are of type **mathord**. If we use the Euler font it is perhaps better to have them be of type **mathalpha**

```

1339 \ifmst@goahead
1340 \ifmst@eulergreek
1341 \DeclareMathSymbol{\alpha}{\mathalpha}{mteulervm}{0B}
1342 \DeclareMathSymbol{\beta}{\mathalpha}{mteulervm}{0C}
1343 \DeclareMathSymbol{\gamma}{\mathalpha}{mteulervm}{0D}
1344 \DeclareMathSymbol{\delta}{\mathalpha}{mteulervm}{0E}
1345 \DeclareMathSymbol{\epsilon}{\mathalpha}{mteulervm}{0F}
1346 \DeclareMathSymbol{\zeta}{\mathalpha}{mteulervm}{10}

```

```

1347 \DeclareMathSymbol{\eta}      {\mathalpha}{mteulervm}{11}
1348 \DeclareMathSymbol{\theta}    {\mathalpha}{mteulervm}{12}
1349 \DeclareMathSymbol{\iota}      {\mathalpha}{mteulervm}{13}
1350 \DeclareMathSymbol{\kappa}     {\mathalpha}{mteulervm}{14}
1351 \DeclareMathSymbol{\lambda}    {\mathalpha}{mteulervm}{15}
1352 \DeclareMathSymbol{\mu}        {\mathalpha}{mteulervm}{16}
1353 \DeclareMathSymbol{\nu}        {\mathalpha}{mteulervm}{17}
1354 \DeclareMathSymbol{\xi}        {\mathalpha}{mteulervm}{18}
1355 \let\omicron\undefined
1356 \DeclareMathSymbol{\omicron}     {\mathalpha}{mteulervm}{6F}
1357 \DeclareMathSymbol{\pi}        {\mathalpha}{mteulervm}{19}
1358 \DeclareMathSymbol{\rho}       {\mathalpha}{mteulervm}{1A}
1359 \DeclareMathSymbol{\sigma}     {\mathalpha}{mteulervm}{1B}
1360 \DeclareMathSymbol{\tau}       {\mathalpha}{mteulervm}{1C}
1361 \DeclareMathSymbol{\upsilon}   {\mathalpha}{mteulervm}{1D}
1362 \DeclareMathSymbol{\phi}       {\mathalpha}{mteulervm}{1E}
1363 \DeclareMathSymbol{\chi}       {\mathalpha}{mteulervm}{1F}
1364 \DeclareMathSymbol{\psi}       {\mathalpha}{mteulervm}{20}
1365 \DeclareMathSymbol{\omega}     {\mathalpha}{mteulervm}{21}
1366 \DeclareMathSymbol{\varepsilon}{\mathalpha}{mteulervm}{22}
1367 \DeclareMathSymbol{\vartheta}{\mathalpha}{mteulervm}{23}
1368 \DeclareMathSymbol{\varpi}     {\mathalpha}{mteulervm}{24}
1369 \let\varrho=\rho
1370 \let\varsigma=\sigma
1371 \DeclareMathSymbol{\varphi}     {\mathalpha}{mteulervm}{27}
1372 \DeclareMathSymbol{\partial}    {\mathalpha}{mteulervm}{40}
1373 \DeclareMathSymbol{\wp}        {\mathalpha}{mteulervm}{7D}
1374 \DeclareMathSymbol{\ell}       {\mathalpha}{mteulervm}{60}
1375 \else
1376 \ifmst@symbolgreek
1377 \DeclareMathSymbol{\alpha}     {\mathord}{mtpsymbol}{61}
1378 \DeclareMathSymbol{\beta}     {\mathord}{mtpsymbol}{62}
1379 \DeclareMathSymbol{\gamma}    {\mathord}{mtpsymbol}{67}
1380 \DeclareMathSymbol{\delta}    {\mathord}{mtpsymbol}{64}
1381 \DeclareMathSymbol{\epsilon}   {\mathord}{mtpsymbol}{65}
1382 \DeclareMathSymbol{\zeta}     {\mathord}{mtpsymbol}{7A}
1383 \DeclareMathSymbol{\eta}      {\mathord}{mtpsymbol}{68}
1384 \DeclareMathSymbol{\theta}    {\mathord}{mtpsymbol}{71}
1385 \DeclareMathSymbol{\iota}     {\mathord}{mtpsymbol}{69}
1386 \DeclareMathSymbol{\kappa}    {\mathord}{mtpsymbol}{6B}
1387 \DeclareMathSymbol{\lambda}   {\mathord}{mtpsymbol}{6C}
1388 \DeclareMathSymbol{\mu}       {\mathord}{mtpsymbol}{6D}
1389 \DeclareMathSymbol{\nu}       {\mathord}{mtpsymbol}{6E}
1390 \DeclareMathSymbol{\xi}       {\mathord}{mtpsymbol}{78}
1391 \let\omicron\undefined
1392 \DeclareMathSymbol{\omicron}    {\mathord}{mtpsymbol}{6F}
1393 \DeclareMathSymbol{\pi}       {\mathord}{mtpsymbol}{70}
1394 \DeclareMathSymbol{\rho}      {\mathord}{mtpsymbol}{72}
1395 \DeclareMathSymbol{\sigma}    {\mathord}{mtpsymbol}{73}

```

```

1396 \DeclareMathSymbol{\tau}{\mathord}{mtpsymbol}{"74}
1397 \DeclareMathSymbol{\upsilon}{\mathord}{mtpsymbol}{"75}
1398 \DeclareMathSymbol{\phi}{\mathord}{mtpsymbol}{"66}
1399 \DeclareMathSymbol{\chi}{\mathord}{mtpsymbol}{"63}
1400 \DeclareMathSymbol{\psi}{\mathord}{mtpsymbol}{"79}
1401 \DeclareMathSymbol{\omega}{\mathord}{mtpsymbol}{"77}
1402 \let\varepsilon=\epsilon
1403 \DeclareMathSymbol{\vartheta}{\mathord}{mtpsymbol}{"4A}
1404 \DeclareMathSymbol{\varpi}{\mathord}{mtpsymbol}{"76}
1405 \let\varrho=\rho
1406 \DeclareMathSymbol{\varsigma}{\mathord}{mtpsymbol}{"56}
1407 \DeclareMathSymbol{\varphi}{\mathord}{mtpsymbol}{"6A}
1408 \DeclareMathSymbol{\partial}{\mathord}{mtpsymbol}{"B6}
1409 \DeclareMathSymbol{\wp}{\mathord}{mtpsymbol}{"C3}
1410 \fi\fi\fi
1411 %%
1412 \ifmst@LGRgreek % internal names for subdued implementation (1.15c-1.15e)
1413 \let\mst@origAlpha\Alpha
1414 \let\mst@origBeta\Beta
1415 \let\mst@origGamma\Gamma
1416 \let\mst@origDelta\Delta
1417 \let\mst@origEpsilon\Epsilon
1418 \let\mst@origZeta\Zeta
1419 \let\mst@origEta\Eta
1420 \let\mst@origTheta\Theta
1421 \let\mst@origIota\Iota
1422 \let\mst@origKappa\Kappa
1423 \let\mst@origLambda\Lambda
1424 \let\mst@origMu\Mu
1425 \let\mst@origNu\Nu
1426 \let\mst@origXi\Xi
1427 \let\mst@origOmicron\Omicron
1428 \let\mst@origPi\Pi
1429 \let\mst@origRho\Rho
1430 \let\mst@origSigma\Sigma
1431 \let\mst@origTau\Tau
1432 \let\mst@origUpsilon\Upsilon
1433 \let\mst@origPhi\Phi
1434 \let\mst@origChi\Chi
1435 \let\mst@origPsi\Psi
1436 \let\mst@origOmega\Omega
1437 %
1438 \let\mst@origalpha\alpha
1439 \let\mst@origbeta\beta
1440 \let\mst@origgamma\gamma
1441 \let\mst@origdelta\delta
1442 \let\mst@origepsilon\epsilon
1443 \let\mst@origzeta\zeta
1444 \let\mst@origeta\eta

```

```

1445 \let\mst@origtheta\theta
1446 \let\mst@origiota\iota
1447 \let\mst@origkappa\kappa
1448 \let\mst@origlambda\lambda
1449 \let\mst@origmu\mu
1450 \let\mst@orignu\nu
1451 \let\mst@origxi\xi
1452 \let\mst@origomicron\omicron
1453 \let\mst@origpi\pi
1454 \let\mst@origrho\rho
1455 \let\mst@origvarsigma\varsigma
1456 \let\mst@origsigma\sigma
1457 \let\mst@origtau\tau
1458 \let\mst@origupsilon\upsilon
1459 \let\mst@origphi\phi
1460 \let\mst@origchi\chi
1461 \let\mst@origpsi\psi
1462 \let\mst@origomega\omega
1463 \let\mst@origDigamma\Digamma
1464 \let\mst@origdigamma\digamma
1465 %%
1466 \def\mst@restoregreek{%
1467 \let\Alpha\mst@origAlpha
1468 \let\Beta\mst@origBeta
1469 \let\Gamma\mst@origGamma
1470 \let\Delta\mst@origDelta
1471 \let\Epsilon\mst@origEpsilon
1472 \let\Zeta\mst@origZeta
1473 \let\Eta\mst@origEta
1474 \let\Theta\mst@origTheta
1475 \let\Iota\mst@origIota
1476 \let\Kappa\mst@origKappa
1477 \let\Lambda\mst@origLambda
1478 \let\Mu\mst@origMu
1479 \let\Nu\mst@origNu
1480 \let\Xi\mst@origXi
1481 \let\Omicron\mst@origOmicron
1482 \let\Pi\mst@origPi
1483 \let\Rho\mst@origRho
1484 \let\Sigma\mst@origSigma
1485 \let\Tau\mst@origTau
1486 \let\Upsilon\mst@origUpsilon
1487 \let\Phi\mst@origPhi
1488 \let\Chi\mst@origChi
1489 \let\Psi\mst@origPsi
1490 \let\Omega\mst@origOmega
1491 %
1492 \let\alpha\mst@origalpha
1493 \let\beta\mst@origbeta

```

```

1494 \let\gamma\mst@origgamma
1495 \let\delta\mst@origdelta
1496 \let\epsilon\mst@origepsilon
1497 \let\zeta\mst@origzeta
1498 \let\eta\mst@origeta
1499 \let\theta\mst@origtheta
1500 \let\iota\mst@origiota
1501 \let\kappa\mst@origkappa
1502 \let\lambda\mst@origlambda
1503 \let\mu\mst@origmu
1504 \let\nu\mst@orignu
1505 \let\xi\mst@origxi
1506 \let\omicron\mst@origomicron
1507 \let\pi\mst@origpi
1508 \let\rho\mst@origrho
1509 \let\varsigma\mst@origvarsigma
1510 \let\sigma\mst@origsigma
1511 \let\tau\mst@origtau
1512 \let\upsilon\mst@origupsilon
1513 \let\phi\mst@origphi
1514 \let\chi\mst@origchi
1515 \let\psi\mst@origpsi
1516 \let\omega\mst@origomega
1517 \let\Digamma\mst@origDigamma
1518 \let\digamma\mst@origdigamma
1519 }
1520 %%
1521 \def\mst@setuplgrgreek{%
1522 \let\Alpha\mst@Alpha
1523 \let\Beta\mst@Beta
1524 \let\Gamma\mst@Gamma
1525 \let\Delta\mst@Delta
1526 \let\Epsilon\mst@Epsilon
1527 \let\Zeta\mst@Zeta
1528 \let\Eta\mst@Eta
1529 \let\Theta\mst@Theta
1530 \let\Iota\mst@Iota
1531 \let\Kappa\mst@Kappa
1532 \let\Lambda\mst@Lambda
1533 \let\Mu\mst@Mu
1534 \let\Nu\mst@Nu
1535 \let\Xi\mst@Xi
1536 \let\Omicron\mst@Omicron
1537 \let\Pi\mst@Pi
1538 \let\Rho\mst@Rho
1539 \let\Sigma\mst@Sigma
1540 \let\Tau\mst@Tau
1541 \let\Upsilon\mst@Upsilon
1542 \let\Phi\mst@Phi

```

```

1543 \let\Chi\mst@Chi
1544 \let\Psi\mst@Psi
1545 \let\Omega\mst@Omega
1546 %
1547 \let\alpha\mst@alpha
1548 \let\beta\mst@beta
1549 \let\gamma\mst@gamma
1550 \let\delta\mst@delta
1551 \let\epsilon\mst@epsilon
1552 \let\zeta\mst@zeta
1553 \let\eta\mst@eta
1554 \let\theta\mst@theta
1555 \let\iota\mst@iota
1556 \let\kappa\mst@kappa
1557 \let\lambda\mst@lambda
1558 \let\mu\mst@mu
1559 \let\nu\mst@nu
1560 \let\xi\mst@xi
1561 \let\omicron\mst@omicron
1562 \let\pi\mst@pi
1563 \let\rho\mst@rho
1564 \let\varsigma\mst@varsigma
1565 \let\sigma\mst@sigma
1566 \let\tau\mst@tau
1567 \let\upsilon\mst@upsilon
1568 \let\phi\mst@phi
1569 \let\chi\mst@chi
1570 \let\psi\mst@psi
1571 \let\omega\mst@omega
1572 \let\Digamma\mst@Digamma
1573 \let\digamma\mst@digamma
1574 }
1575 %%
1576 \DeclareMathSymbol{\mst@Alpha}{\mathalpha}{mtlgrfontupper}{65}
1577 \DeclareMathSymbol{\mst@Beta}{\mathalpha}{mtlgrfontupper}{66}
1578 \DeclareMathSymbol{\mst@Gamma}{\mathalpha}{mtlgrfontupper}{71}
1579 \DeclareMathSymbol{\mst@Delta}{\mathalpha}{mtlgrfontupper}{68}
1580 \DeclareMathSymbol{\mst@Epsilon}{\mathalpha}{mtlgrfontupper}{69}
1581 \DeclareMathSymbol{\mst@Zeta}{\mathalpha}{mtlgrfontupper}{90}
1582 \DeclareMathSymbol{\mst@Eta}{\mathalpha}{mtlgrfontupper}{72}
1583 \DeclareMathSymbol{\mst@Theta}{\mathalpha}{mtlgrfontupper}{74}
1584 \DeclareMathSymbol{\mst@Iota}{\mathalpha}{mtlgrfontupper}{73}
1585 \DeclareMathSymbol{\mst@Kappa}{\mathalpha}{mtlgrfontupper}{75}
1586 \DeclareMathSymbol{\mst@Lambda}{\mathalpha}{mtlgrfontupper}{76}
1587 \DeclareMathSymbol{\mst@Mu}{\mathalpha}{mtlgrfontupper}{77}
1588 \DeclareMathSymbol{\mst@Nu}{\mathalpha}{mtlgrfontupper}{78}
1589 \DeclareMathSymbol{\mst@Xi}{\mathalpha}{mtlgrfontupper}{88}
1590 \DeclareMathSymbol{\mst@Omicron}{\mathalpha}{mtlgrfontupper}{79}
1591 \DeclareMathSymbol{\mst@Pi}{\mathalpha}{mtlgrfontupper}{80}

```

```

1592 \DeclareMathSymbol{\mst@Rho}{\mathalpha}{mtlgrfontupper}{82}
1593 \DeclareMathSymbol{\mst@Sigma}{\mathalpha}{mtlgrfontupper}{83}
1594 \DeclareMathSymbol{\mst@Tau}{\mathalpha}{mtlgrfontupper}{84}
1595 \DeclareMathSymbol{\mst@Upsilon}{\mathalpha}{mtlgrfontupper}{85}
1596 \DeclareMathSymbol{\mst@Phi}{\mathalpha}{mtlgrfontupper}{70}
1597 \DeclareMathSymbol{\mst@Chi}{\mathalpha}{mtlgrfontupper}{81}
1598 \DeclareMathSymbol{\mst@Psi}{\mathalpha}{mtlgrfontupper}{89}
1599 \DeclareMathSymbol{\mst@Omega}{\mathalpha}{mtlgrfontupper}{87}
1600 %
1601 \DeclareMathSymbol{\mst@alpha}{\mathalpha}{mtlgrfontlower}{97}
1602 \DeclareMathSymbol{\mst@beta}{\mathalpha}{mtlgrfontlower}{98}
1603 \DeclareMathSymbol{\mst@gamma}{\mathalpha}{mtlgrfontlower}{103}
1604 \DeclareMathSymbol{\mst@delta}{\mathalpha}{mtlgrfontlower}{100}
1605 \DeclareMathSymbol{\mst@epsilon}{\mathalpha}{mtlgrfontlower}{101}
1606 \DeclareMathSymbol{\mst@zeta}{\mathalpha}{mtlgrfontlower}{122}
1607 \DeclareMathSymbol{\mst@eta}{\mathalpha}{mtlgrfontlower}{104}
1608 \DeclareMathSymbol{\mst@theta}{\mathalpha}{mtlgrfontlower}{106}
1609 \DeclareMathSymbol{\mst@iota}{\mathalpha}{mtlgrfontlower}{105}
1610 \DeclareMathSymbol{\mst@kappa}{\mathalpha}{mtlgrfontlower}{107}
1611 \DeclareMathSymbol{\mst@lambda}{\mathalpha}{mtlgrfontlower}{108}
1612 \DeclareMathSymbol{\mst@mu}{\mathalpha}{mtlgrfontlower}{109}
1613 \DeclareMathSymbol{\mst@nu}{\mathalpha}{mtlgrfontlower}{110}
1614 \DeclareMathSymbol{\mst@xi}{\mathalpha}{mtlgrfontlower}{120}
1615 \DeclareMathSymbol{\mst@omicron}{\mathalpha}{mtlgrfontlower}{111}
1616 \DeclareMathSymbol{\mst@pi}{\mathalpha}{mtlgrfontlower}{112}
1617 \DeclareMathSymbol{\mst@rho}{\mathalpha}{mtlgrfontlower}{114}
1618 \DeclareMathSymbol{\mst@varsigma}{\mathalpha}{mtlgrfontlower}{99}
1619 \DeclareMathSymbol{\mst@sigma}{\mathalpha}{mtlgrfontlower}{115}
1620 \DeclareMathSymbol{\mst@tau}{\mathalpha}{mtlgrfontlower}{116}
1621 \DeclareMathSymbol{\mst@upsilon}{\mathalpha}{mtlgrfontlower}{117}
1622 \DeclareMathSymbol{\mst@phi}{\mathalpha}{mtlgrfontlower}{102}
1623 \DeclareMathSymbol{\mst@chi}{\mathalpha}{mtlgrfontlower}{113}
1624 \DeclareMathSymbol{\mst@psi}{\mathalpha}{mtlgrfontlower}{121}
1625 \DeclareMathSymbol{\mst@omega}{\mathalpha}{mtlgrfontlower}{119}
1626 \DeclareMathSymbol{\mst@digamma}{\mathalpha}{mtlgrfontlower}{147}
1627 \DeclareMathSymbol{\mst@Digamma}{\mathalpha}{mtlgrfontlower}{195}
1628 % cf http://milde.users.sourceforge.net/LGR/lgrxenc.def.html
1629 % et greek.ldf du package babel
1630 \ifmst@subdued\else\mst@setuplgrgreek\fi
1631 \fi

```

\inodot In 1.0, I had them of type mathord, here I choose mathalpha. If I used \i and \j from the text
\jnodot 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.

```

1632 \edef\mst@tmp@enc{\encodingdefault}
1633 \mst@goaheadtrue
1634 \ifx\mst@tmp@enc\mst@eui % xetex and unicode font

```



```

1635 \XeTeXmathchardef\inodot="7 \symmtletterfont "0131 \relax
1636 \XeTeXmathchardef\jnodot="7 \symmtletterfont "0237 \relax
1637 \else
1638 \ifx\mst@tmp@enc\mst@euui % luatex and unicode font
1639 \luatexUmathchardef\inodot="7 \symmtletterfont "0131 \relax
1640 \luatexUmathchardef\jnodot="7 \symmtletterfont "0237 \relax
1641 \else
1642 \ifx\mst@tmp@enc\mst@ti % T1
1643 \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{25}
1644 \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{26}
1645 \else
1646 \ifx\mst@tmp@enc\mst@oti % OT1
1647 \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
1648 \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
1649 \else
1650 \ifx\mst@tmp@enc\mst@lyi % LY1
1651 \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
1652 \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
1653 \else
1654 \ifXeTeX\mst@goaheadfalse\else\ifLuaTeX\mst@goaheadfalse\else
1655 \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}
1656 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\i\endcsname}
1657 \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}
1658 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\j\endcsname}
1659 \fi\fi
1660 \fi\fi\fi\fi\fi
1661 \ifmst@defaultimath\else
1662 \ifmst@goahead
1663 \renewcommand{\imath}{\inodot}
1664 \renewcommand{\jmath}{\jnodot}
1665 \let\mst@oldi\i \let\mst@oldj\j
1666 \DeclareRobustCommand{\i}{\ifmmode\inodot\else\mst@oldi\fi}
1667 \DeclareRobustCommand{\j}{\ifmmode\jnodot\else\mst@oldj\fi}
1668 \fi
1669 \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 L^AT_EX, 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

```

1670 \edef\mst@tmp@enc{\encodingdefault} %% rather one too many than sorry
1671 \ifmst@mathaccents
1672 \ifx\mst@ti\mst@tmp@enc
1673 \DeclareMathAccent{\acute}{\mathalpha}{mtooperatorfont}{1}
1674 \DeclareMathAccent{\grave}{\mathalpha}{mtooperatorfont}{0}
1675 \DeclareMathAccent{\ddot}{\mathalpha}{mtooperatorfont}{4}
1676 \DeclareMathAccent{\tilde}{\mathalpha}{mtooperatorfont}{3}

```

```

1677 \DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{9}
1678 \DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{8}
1679 \DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{7}
1680 \DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{2}
1681 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{10}
1682 \DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{6}
1683 \else
1684 \DeclareMathAccent{\acute}{\mathalpha}{mtoperatorfont}{19}
1685 \DeclareMathAccent{\grave}{\mathalpha}{mtoperatorfont}{18}
1686 \DeclareMathAccent{\ddot}{\mathalpha}{mtoperatorfont}{127}
1687 \DeclareMathAccent{\tilde}{\mathalpha}{mtoperatorfont}{126}
1688 \DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{22}
1689 \DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{21}
1690 \DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{20}
1691 \DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{94}
1692 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{95}
1693 \DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{23}
1694 \ifx\mst@lyi\mst@tmp@enc % LY1 encoding
1695 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{5}
1696 \else
1697 \ifx\mst@oti\mst@tmp@enc\else
1698 \typeout{** mathastext: math accents have been assumed to be~J%
1699 ** as in OT1 encoding.}
1700 \fi
1701 \fi
1702 \fi\fi

```

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

```

1703 \ifmst@defaultsizes\else
1704 \providecommand\@xxxpt{29.86}
1705 \providecommand\@xxxvpt{35.83}
1706 \ifmst@twelve
1707 \def\Huge{\@setfontsize\Huge\@xxxpt{36}}
1708 \def\HUGE{\@setfontsize\HUGE\@xxxvpt{43}}
1709 \typeout{** \protect\Huge\space and \protect\HUGE\space have been (re)-defined.}
1710 \else
1711 \def\HUGE{\@setfontsize\HUGE\@xxxpt{36}}
1712 \typeout{** \protect\HUGE\space has been (re)-defined.}
1713 \fi

```

I choose rather big subscripts.

```

1714 \def\defaultscriptratio{.8333}
1715 \def\defaultscriptscriptratio{.7}
1716 \DeclareMathSizes{9}{9}{7}{5}
1717 \DeclareMathSizes{\@xpt}{\@xpt}{8}{6}
1718 \DeclareMathSizes{\@xipt}{\@xipt}{9}{7}
1719 \DeclareMathSizes{\@xiipt}{\@xiipt}{10}{8}
1720 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xiipt}{10}
1721 \DeclareMathSizes{\@xvipt}{\@xvipt}{\@xivpt}{\@xiipt}
1722 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xvipt}{\@xivpt}

```

```

1723 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xvipt}
1724 \DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxvpt}{\@xxpt}
1725 \DeclareMathSizes{\@xxxvpt}{\@xxxvpt}{\@xxxpt}{\@xxvpt}
1726 \typeout{** mathastext has declared larger sizes for subscripts.^^J%
1727 ** To keep LaTeX defaults, use option 'defaultmathsizes'.}
1728 \fi

```

Scaling mechanism for the Symbol font and the end of the package.

```

1729 \AtBeginDocument{
1730 \ifmst@needsymbol
1731   \DeclareFontFamily{U}{psy}{\}
1732   \DeclareFontShape{U}{psy}{m}{n}{<->s*[\psy@scale] psy}{\}
1733   \fi
1734 }
1735 \endinput

```