

The **mathastext** package

JEAN-FRANÇOIS BURNOL
jfbu (at) free (dot) fr
Version 1.3a of 2013/09/04¹

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.

`'mathastext'` is a LaTeX package

```
\usepackage{mathastext}
```

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

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

The command `\MTsetmathskips` allows to set up extra spacings around each given letter.

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

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

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

¹Documentation generated (via `latex+dvipdfmx`) from the source file with timestamp "05-09-2013 at 11:03:03 CEST".

Contents

Recent changes and installation instructions	1, p. 2	Extended scope of the math alphabets commands	2.8, p. 16
Recent changes	1.1, p. 2	Greek letters	2.9, p. 18
Installation	1.2, p. 3	Shape of Greek letters.	
What <code>mathastext</code> does	2, p. 3	Unicode engines	2.10, p. 20
Examples	2.1, p. 3	Compatibility issues	2.11, p. 23
Overview	2.2, p. 5	Package options and commands	3, p. 25
Basic use—sans in math—always load <code>mathastext</code> last—option <code>LGRgreek</code> —avoid OT1 encoding.		Summary of main options	3.1, p. 25
Main options	2.3, p. 8	Miscellaneous	3.2, p. 26
The italic option—The defaultmathsizes option—The subdued option.		Commands	3.3, p. 28
Math versions	2.4, p. 9	Preamble-only commands—Commands usable everywhere—Body-only commands.	
Extra spaces around letters	2.5, p. 11	Complete list of options	3.4, p. 35
Italic corrections	2.6, p. 12	Change log	4, p. 37
Other extra glue in math mode	2.7, p. 15	Implementation	5, p. 40

1 Recent changes and installation instructions

1.1 Recent changes

This is the documentation for version 1.3a of 2013/09/04.

1. (see [subsection 2.5](#)) new command `\MTsetmathskips` allows to specify, on a letter by letter basis, extra spaces (expressed in terms of ‘mu’ units, or as `\thinmuskip` for example) to be inserted automatically in math mode around the specified letter. This is provided for text fonts whose use in math mode creates glyph collisions or excessive crampedness,
2. use of the latest version of `\luatexUmathcodenum` now allows equal treatment of = and - by `mathastext` under both unicode engines LuaTeX and XeTeX. So `mathastext` version 1.3 under LuaLaTeX *requires* the version of `lualatex` (binaries and format) as included for example in the TeXLive 2013 distribution, or later,
3. a few minor changes and code improvements for better compatibility in various contexts,
4. blue keywords in the documentation are now (mostly) hyperlinked to their more detailed descriptions,
5. various other documentation improvements.

Version 1.2f: addition of the “change log” at the end of this user manual, and some minor code improvements not changing neither features nor user interface.

Version 1.2e made additions: in this user manual to the section 2.11 describing the compatibility issues, and to the test files illustrating various package features. Important changes to the source code were done to fix compatibility problems with active characters dating back to the 1.2 version. Also, an oversight in the implementation of the italic correction features from 1.2b was corrected.

Version 1.2d introduced the `asterisk` option and addressed a problem of compatibility with `amsmath`.

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

Earlier, version 1.15 introduced the concept of the [subdued math versions](#).

1.2 Installation

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

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

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

2 What `mathastext` does

2.1 Examples

`mathastext`'s basic aim is to have the same font for text and mathematics. With hundreds of free text fonts packaged for 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. Note that `mathastext` was initially developed for the traditional T_EX fonts and engines, and that compatibility with Unicode engines and OpenType fonts is partial.

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 \TeX distribution). Let us now be a bit more original and have our mathematics with italic letters from the sans serif font Helvetica, while the letters in text use New Century Schoolbook.

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

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

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

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

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

And after all that, we may wish to return to the default math typesetting (let's shorten the extract here in case the reader makes an indigestion ...):

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

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

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 **cmmtt**, **cmss**, and **txr**, which are part of standard T_EX distributions.²

2.2 Overview

2.2.1 Basic use

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

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}
```

²The 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 **txr** font (again no package loading is necessary) is part of the files of the **txfontsb** package.

The package records which font is set up for text, at the time it is loaded,³ and then arranges things in order for this text font to be used in math mode as well. So, with the preamble as above all letters, digits, and punctuation signs inside math mode will then be typeset in Times.⁴ The exact list of characters concerned by `mathastext` is a subset of the basic ASCII set:

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

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

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

- the derivative sign ' is left to its default as the text font glyph ' is not, as a rule, a satisfying alternative.⁵
- for the minus sign `mathastext` uses the endash character –, if available, and not the hyphen character -,
- the option `asterisk` is necessary for `mathastext` to replace the binary math operator * (and the equivalent control sequence `\ast`) with a version which uses the text asterisk * suitably lowered⁶ (and with the correct spaces around it as binary operator). The reason for making it optional is that after this `R^*` or `$R^{\ast}` do not work and have to be written `$R^{\text{*}}` or `$R^{\text{\ast}}`.

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

2.2.2 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.

³alternatively it is possible to configure the text font after loading `mathastext`, and then the command `\Mathastext` will accomplish the necessary changes to the font for letters, digits and a few extra ascii symbols in math mode.

⁴let'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.

⁵v1.2 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' .

⁶the amount of lowering can be customized.

```

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

```

2.2.3 always load `mathastext` last

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

2.2.4 option `LGRgreek`

There is the issue of Greek letters. Sometimes the text font has Greek glyphs, in LGR encoding (this will be mentioned in the documentation of the font package). Then option `LGRgreek` tells `mathastext` to pick up these Greek letters. And it is possible to specify whether the Greek letters should be upright, or “italic”.⁷ Of course it is always possible to leave the responsibility to set up Greek letters in math mode to packages loaded previously to `mathastext`.

2.2.5 avoid OT1 encoding

We specified in our minimal working example a T1 encoding (LY1 would have been ok, too) because the default OT1 does not have the `<>|{}&` and `\` glyphs. If `mathastext` detects OT1 as the default encoding it will leave these characters to their defaults from the math fonts.⁸

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.

⁷a 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.

⁸the `subdued` option, described next, acts a bit otherwise, it forces, contrarily to its usual low-key character, the replacement of OT1 by T1 for the fonts ultimately used with letters and digits in math mode.

2.3 Main options

2.3.1 The `italic` option

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

Nevertheless, soon after I posted the initial version of the package to CTAN, I was overwhelmed by numerous⁹ questions¹⁰ 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.¹¹ There is also an option `frenchmath` to make the uppercase letters nevertheless upright, because this is the way of traditional French mathematical typography.¹²

2.3.2 The `defaultmathsizes` option

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

2.3.3 The `subdued` option

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

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

⁹this means “more than one.”

¹⁰I thank in particular Tariq PERWEZ and Kevin KLEMENT for their kind remarks (chronological order).

¹¹more precisely stated, the value of `\itdefault` is used.

¹²more precisely stated, the value of `\shapedefault` is used.

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

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

2.4 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*¹³: `lmvtt`,

¹³math versions are discussed in the document `fntguide.pdf` from your T_EX distribution.

`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, `\MTversion{name_of_version}`, or equivalently `\Mathastextversion{name_of_version}`, enacts the font switches in the body of the document. As is usual with L^AT_EX one can limit the scope to the inside of a group, or also switch back to the main set-up through issuing `\Mathas-textversion{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 T_EX 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 L^AT_EX command `\mathversion{<version_name>}` will change only the fonts used in math mode. It is recommended to use instead the package command `\MTversion` (or its synonyms `\mathastextversion`, `\Mathas-textversion`, `\MTVersion`), with a mandatory argument `{<version_name>}`. It

- checks in case the `subdued` option was specified whether the asked-for math version is `<normal>` or `<bold>`, and adapts the following to that case,
- sets the font which will be used in math mode for letters (including math operator names), digits, punctuations and other ascii symbols,
- sets the font of the document text (if another version name is additionally passed as optional argument, it uses instead the corresponding font for text),
- resets the `\(family,rm,sf,tt)defaults` to their values as registered at the time of definition of the version,
- (see sections 2.5 and 2.6) re-issues the command `\MTmathactiveletters` to let a to z, A to Z, be mathematically active in order to automatically insert the skips as defined by the user with `\MTsetmathskips`, and the italic corrections (if the font is not italic or slanted),

- (see section 2.7) resets the extra spaces after the symbols \exists , \forall and before the derivative $'$ to the values as decided by the user in the preamble on a *per version* basis,
- (see section 2.8) makes the math operator names as well as the ‘easy’ non letter characters (and the asterisk) obey the math alphabets,
- does some additional set up for Greek letters when the `LGRgreek` option was used.

The scope is limited to the current L^AT_EX 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.¹⁴ But one can use instead `\renewcommand{\rmdefault}{epigrafica}`,¹⁵ followed with `\Mathastext`, or also `\MTfamily{epigrafica}\Mathastext` which will only change the font in math.

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

2.5 Extra spaces around letters

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

```
% this may be anywhere in the document (also within a math group):
\MTsetmathskips{x}{20.33mu}{15.66mu}% 20.33mu before all x's and 15.66mu after.
\MTsetmathskips{y}{\thickmuskip}{\thickmuskip}%
\MTsetmathskips{z}{10mu plus 10mu minus 5mu}{5mu plus 2mu minus 3mu}%
\MTsetmathskips{A}{\muexpr \thickmuskip*2}{\muexpr \medmuskip-\thinmuskip/2}%

```

Here is what `$\wxytztz\wxytztz=BAC\BAC$` then gives using the Times font:
 $w \ x \ t \ y \ t \ z \ w \ x \ t \ y \ t \ z = B \ A \ C^B \ A \ C$. Any T_EX group or L^AT_EX environment limits as usual the scope of this command. Furthermore the command `\MTunsetmathskips` cancels previous use of `\MTsetmathskips` for a given letter (in a more complete manner than re-using it to specify 0mu as the before and after skips).

The implementation relies on the ‘mathematical activation’ of letters, which is done by default by the package since release 1.2b. Should this cause compatibility

¹⁴may typically give a ‘too many math alphabets’ error message.

¹⁵sometimes 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 T_EX 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.

problems, the command `\MTmathstandardletters` cancels it entirely. To reactivate it, there is `\MTmathactiveletters`. Note that `\MTmathactiveletters` is done automatically by `mathastext` when loaded, and also each time the package enhanced math-version-switch command `\MTversion` is used, except for the normal and bold math versions under the `subdued` option.

Earlier this ‘mathematical activation’ of letters had been used by the package only to add automatically italic corrections, as described in the next section.

2.6 Italic corrections

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

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

For the built-in placement routines of $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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. 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 $f(x)$ (21.31474pt) than it is with the text italic font used in math.¹⁷ $f(x)$ (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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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. f .

Without the italic corrections¹⁹ it is f vs. f . I said that \mathbf{f} included the italic correction automatically, but if we tell $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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:

$abcdefghijklmnopqrstuvwxy$	text italic in text
$abcdefghijklmnopqrstuvwxy$	text italic in math
$abcdefghijklmnopqrstuvwxy$	math italic in math
$abcdefghijklmnopqrstuvwxy$	math italic in text

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

¹⁷we used simply $\mathbf{\mathit{f(x)}}$.

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!

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 $\mathbf{f}(\mathbf{x})$. 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 very different from $f^{\textit{j}}$... where the latter was coded with `\hbox{\itshape f}j`. The inputs `\mathit{\hbox{\itshape f}j}` and `\mathit{f}j` give almost identical results: $f^{\textit{j}}$ vs. $f^{\textit{j}}$. Close examination reveals that the horizontal spacing is exactly identical, however the exponent in the second case is a bit lower. Anyway, the point is that in the second case the italic correction for f was indeed used.

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

We learn in Appendix G of the T_EXbook 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^j for more accurate comparison of the positioning of the sub- and superscript.

If we try something like this: \mathbf{f}_i^i we obtain f_i^i . Our overlapping game with `\rlap{\mathbf{f}_i^i}\mathbf{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 T_EX 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 T_EX fonts may also have italic correction data! And what I just said about

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

the shift of the superscript with respect to the subscript apply equally well to such a font, if T_EX has been told to use it. Let's try Latin Modern Upright for letters in math: `f_i` now gives²¹ f_i . We see the italic correction in action for the positioning of the superscript! Compare with `$\mathrm{\hbox{f}}_i$`: f_i . Overlapping with `\rlap{f_i}$\mathrm{\hbox{f}}_i$` gives f_i and shows that the upright f has an italic correction which was used to shift the superscript to the right (and it is now in a slightly lower position). Let's now do `$\mathrm{f\!/}_i$`: this gives f_i and the subscript is shifted to the right, and is now on the same vertical axis as the superscript. There are also some slight vertical displacements, `\rlap{f_i}$\mathrm{f\!/}_i$` gives f_i .

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

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

Starting with version v1.2b, `mathastext` adds the italic correction automatically after each letter of the Latin alphabet in math mode, *except* when these letters are italic or slanted.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 3.3.2.

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

²¹we just use `f_i`.

²²the situation is rather ironical! by the way, the warnings in section 2.8 with `$x^?$` or similar are less of an issue here, because the letter is only *followed* by `\/` and anyhow the whole is put by `mathastext` within group braces, so no surprises with `x^y` or `$\mathbin{x}`. Nevertheless it is still true that (in math mode only) the letters a-z, A-Z, expand to composite objects, something which could surprise other packages. The command `\MTmathstandardletters` cancels this mechanism.

(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
$\mathit{f_i}$\par $\mathrm{f_i}$
\end{document}
```

Comment out the `fontspec` line and use pdfL^AT_EX. All three outputs are different on my T_EX installation. X_YL^AT_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.

2.7 Other extra glue in math mode

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

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

These three commands may be used throughout the document, or also in the preamble, in which case the declared math versions will record the then current values of the skips. `mathastext` applies the following (small) default skips: `0.6667mu` for the skip after \forall , `1mu` for the skip after \exists , and `0.5mu` for the skip before the prime. The examples above become $\forall B$, $\exists N$ and f' .²³

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

Also, when the `subdued` option has been used, the normal and bold math versions have by default zero length skips. Note though that there is no `\MTversion{normal}`

²³the derivative glyph from the txfonts math symbols adapts itself better to an upright letter, no skip seems to be needed then.

done implicitly by the package when the document body starts²⁴ (*i.e.* at the level of `\begin{document}`), even when the `subdued` option is in force. As a result the last use in the preamble of the `\MT...skip` commands decides the skips which will be initially used; this is cancelled if `\begin{document}` is followed with `\MTversion{normal}`, in which case the skips used are the ones in force at the latest `\Mathastext` (without optional argument) preamble use, or just zero skips under the `subdued` regime.

2.8 Extended scope of the math alphabets commands

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

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

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

1. the pre-declared or user-declared (using the `amsmath \DeclareMathOperator` or equivalent) operator names obey the math alphabet commands,²⁵
2. and, *optionally*, all non alphabetical characters²⁶ treated by `mathastext`, *i.e.*, if not disabled by options, `! ? , : ; + - = () [] < > { }`, the asterisk `*`, and `.` `/` `|` `\` `#` `$` `%` `&`²⁷ will also obey the math alphabet commands (when not used as delimiters). The important thing is that the spaces added by `TeX` before and after are not modified.

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

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

²⁴when `\MTversion` is used, `mathastext` resets all font defaults, so for example a `\renewcommand\sfdefault` at the end of the preamble would be overruled if `\MTversion{normal}` was done automatically. Perhaps this behavior of `\MTversion` will change in a future version of the package.

²⁵contrarily to the next feature, this one is not likely to create incompatibilities with other packages, so it is activated by default.

²⁶of course some of them are input preceded by a backslash, and the backslash itself is input as `\backslash`.

²⁷`#` `$` `%` `&` obey the math alphabets since the initial version of `mathastext`; the dot `.`, the slash `/`, the vertical bar `|` and the backslash `\` do not have specific spacings inserted by `TeX` around them, and the procedure is then not a devilish one, this is why it is made the default for these characters which are listed apart. The math symbols `\mid` (which is `|` with type `\mathrel`) and `\setminus` (`\` with type `\mathbin`) are counted among the ‘difficult’ cases, not the ‘easy non-letters’.

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

```
\MTmathoperatorsdonotobeymathxx
\MTnonlettersdonotobeymathxx
\MTeasyonlettersdonotobeymathxx
```

for deactivation and

```
\MTmathoperatorsobeymathxx
\MTnonlettersobeymathxx
\MTeasyonlettersobeymathxx
```

for activation.^{28 29}

Important: the package does `\MTnonlettersdonotobeymathxx` by default. The reason is that activating the mechanism adds some constraints to the way things must be input, adding `\usepackage{mathastext}\MTnonlettersobeymathxx` to a pre-existing document might well create errors: all these characters treated by `mathastext`, such as `?`, `[`, `<` now represent (in math mode only!) *two* ‘tokens’ and this will utterly confuse $\mathrm{T\!E\!X}$ if some precautions are not taken: `$x^?$`, `$R^+ $` or `$\mathopen{A\mathclose{}}$` *must* now be coded as `$x^{\?}$`, `$R^{\+}$` and `$\mathopen{\<}A\mathclose{\>}$` (the rule is to do as if `?`, `+`, `<` or `>` were each really *two* characters).

Even if this rule is respected in the document source, it is still a possibility that incompatibilities with other packages will arise because `mathastext` does a ‘mathematical activation’ of the characters which could be unexpected and unchecked for by other packages. This is precisely the case with the `amsmath`

²⁸these commands are to be used outside of math mode. Their scope is limited to the current $\mathrm{L\!A\!T\!E\!X}$ environment or group. They use the `\everymath` and `\everydisplay` mechanism so if the document needs to modify these token lists it has to do so in a responsible manner, extending not annihilating their previous contents.

²⁹when in subdued mode, the math alphabets are the default ones, not the ones modified by `mathastext` to use the document text fonts. As a result, matters of font encodings may then give unexpected results, for example for `-`. On the present document page we switched to a math version to escape from the subdued mode and avoid the problem with `\mathbf{-}` giving in the normal (subdued) math version `˘`, when ‘non-letters’ are declared to obey math alphabets.

package, and the problem goes away by just making sure that `amsmath` is loaded before `mathastext` (generally speaking, `mathastext` should be loaded last after all packages dealing with math things).

The brace control sequences `\{` and `\}` have their own (supplementary) switch, which is (even) less activated by default. The reason is that after `\MTexplic-itbracesobeymathxx`, they regrettably can not be used anymore as delimiters: `\big\{` or `\big\}` cease to function and must be replaced by `\big\lbrace` and `\big\rbrace`. But we can now enjoy $\{a, a > b\}$, $\{\mathbf{a}, \mathbf{a} > \mathbf{b}\}$, $\{a, a > b\}$, or even $\{a, a > b\}$.³⁰

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{\mathopen{\langle}a,b\mathclose{\rangle}}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. The L^AT_EX standard behavior for `\mathbf{\mathopen{\langle}a,b\mathclose{\rangle}}` is $\langle \mathbf{a}, \mathbf{b} \rangle$.

2.9 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

³⁰this last example uses the `\mathnormalbold` additional alphabet defined by `mathastext`.

txfontsb package which extends the txfonts package with LGR-encoded Greek letters; the Droid fonts, the DejaVu fonts, the Comfortaa font, and the Open Sans font. The LGR encoded CM/LM fonts (in serif, sans-serif and typewriter family) give the nice Greek letters in upright shape from the **cbfonts** package. To get these letters in your **mathastext** math mode, you can do the following:

```
% instructions to load the document fonts:
\usepackage{nice_font}
% and then the following:
\renewcommand{\familydefault}{\cmr} % or cmss or cmtt for sans resp. mono
\usepackage[LGRgreek]{mathastext}
\renewcommand{\familydefault}{\rmdefault}
\Mathastext % this re-initializes mathastext with the nice_font,
% without changing the LGR font cmr/cmss/cmtt used for Greek letters
% in math mode.
\begin{document}
```

If you use the **italic** option note that the italic Greek letters from the **cbfonts** are not the same glyphs as the default Greek letters from the OML encoded font **cmmi**.

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

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

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

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

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

1. in its presence, the **LGRgreek** and **LGRgreeks** options cease to have any effect in the normal and bold math versions,
2. **selfGreek(s)**, **eulergreek** and **symbolgreek** act normally, they are not affected by the presence or absence of **subdued**.

2.9.1 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 T_EX 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 `LGRgrees` or `selfGrees` (notice the additional “s”), each math version is assumed to have its text font available in LGR (or OT1 encoding) and also the shapes will be local to the math version.

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

2.10 Unicode engines

`mathastext` has been made minimally unicode-aware and can be used with X_YT_EX or Lua_YT_EX, 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_YTeX), 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`.

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

When using `mathastext` with either X_YTeX or Lua_YTeX 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 work correctly. 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`. For example, when using `polyglossia` one should presumably say:

```
\PassOptionsToPackage{no-math}{fontspec}
```

before the `\usepackage{polyglossia}` as `fontspec` will then be loaded in a manner compatible with `mathastext`.

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

I already mentioned in the section 2.6 the fact that the italic corrections were not available for OpenType fonts under the X_YTeX engine and only partially available for the Lua_YTeX 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 T_EX 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 T_EX fonts in a fixed encoding (T1, or LY1 for example).

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

a Linux machine, the second one on a Mac OS X machine.³¹

```
\documentclass{article}
\usepackage[hscale=0.8]{geometry}
\usepackage{multicol}
\usepackage[no-math]{fontspec}
\usepackage{lmodern}
\usepackage[subdued,italic]{mathastext}
\setmainfont[Color=999999]{Verdana} \Mathastext[Verdana]
\setmainfont[Color=0000FF]{Arial} \Mathastext[Arial]
\setmainfont[Color=00FF00]{DejaVu Serif} \Mathastext[DejaVu]
\MTDeclareVersion{times}{T1}{ptm}{m}{n}
\setmainfont[Color=FF0000]{Andale Mono} \Mathastext[Andale]
\begin{document}
\newcommand\TEST[1]{\MTversion{#1}%
\begin{multicols}{2}
\hbox to\columnwidth{\hbox to\columnwidth{\hfil
$abcdefghijklmnopqrstuvwxyz$\hfil}\kern-2.5em{#1}}
\centerline{ $ABCDEFGHIJKLMNQRSTUvwxyz$ }
\centerline{ $0123456789$ }
\centerline{ $!\,?\,*\,,\,\.\,:\,;\,\,+\,-\,=\,(\,,)\,,\,[\,,]\,,/\,,\#\,,\%
\$,\\,\,\&\,<\\,>\\,|\\,\{\\,\}\\,\backslash$ }
\columnbreak
\centerline{ abcdefghijklmnopqrstuvwxyz }
\centerline{ ABCDEFGHIJKLMNOPQRSTUVWXYZ }
\centerline{ 0123456789}
\centerline{ !\,?\,*\,,\,\.\,:\,;\,\,+\,-\,=\,(\,,)\,,\,[\,,]\,,/\,,\#\,,\%
\$,\\,\,\&\,<\\,>\\,|\\,\{\\,\}\\,\char92 }
\end{multicols}}
\begin{multicols}{2}
\centerline{\textbf{math mode}}
\columnbreak
\centerline{ \textbf{text} }
\end{multicols}
\TEST{DejaVu}\TEST{Verdana}\TEST{times}\TEST{Andale}
\TEST{Arial}\TEST{bold}\TEST{normal}
\end{document}
```

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

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

³¹running latex (in a temporary repertory) on a copy of the file mathastext.dtx will extract extended versions of these examples as test files.

```

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

```

2.11 Compatibility issues

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

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

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

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

This is compatible with using `\label` and `\ref` in and outside of math mode. But a difficulty arises when some other package has made the character ‘globally active’ everywhere in the document. The action of **mathastext** is made anew at each mathematical inline or displayed formula. If it is detected that a character has been activated then nothing further will be done (so the **mathastext** feature³² for that character is lost) *except* if it appears that this activation was done by the Babel system. In that case **mathastext** does not make the character mathematically active but it modifies in the appropriate manner the action of Babel for that character in math mode. Furthermore **mathastext** makes the character mathematically *inactive*.³³

³²italic correction insertion for the latin letters, receptivity to the math alphabet action for the other characters.

³³only the characters `,:! ? + - = < > () [] *` mentioned in section 2.8 as ‘difficult non letters’ (and the right quote ‘) and the latin letters are concerned here; it seems highly improbable that a latin letter $\in \{a-z, A-Z\}$ will have been made globally active (only letters never being used in command names

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

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

DO NOT DO THIS AT HOME: it creates an infinite loop.³⁴ This is due to the fact that the colon is simultaneously active (this is made by `babel+frenchb` at begin document) and mathematically active (done by `mathtools` in the preamble). The interaction gives an infinite loop. Such a situation will be cured by `mathastext`, even loaded before `mathtools`, if use is made of `\MTnonlettersobeymathxx`. At each math formula `mathastext` will detect that Babel has activated the colon, and will cancel the mathematical activation (the precise definition done by `mathtools` was already lost at begin document due to overwriting by `babel` but the fact that the character was mathematically active remained true).

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

Always load `amsmath` before `mathastext`.

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

Important! As is mentioned in the section 2.8, with the “non letters obey math alphabets” mechanism, characters such as `?`, or `[`, now represent *two* ‘tokens’ and this will utterly confuse \TeX if some precautions are not taken. Examples: `$0^+ $` or `$x\mathrel?y $` or `$R^* $` *must* be input now as `$0^{+} $` and, respectively, `$x\mathrel{?}y $` or `$R^{*} $`. This is why the package does `\MTnonlettersdonotobeymathxx` by default.

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

Doing (after the `\begin{document}`) `\everymath={}` and `\everydisplay={}` (preemptively) annihilates all of the `mathastext` (evil) doing with math active

are possible candidates), but `mathastext` has been designed to cope with it, should it happen ...

³⁴This seems to still be the case with Babel 3.9f and frenchb.lfd 2.6e, as tested on Sep. 2, 2013.

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` (already done by default), `\MTmath-standardletters` (cancels the math activation of letters), `\MTnormalprime` (cancels the redefinition done by `mathastext` of the action of the right quote ' in math mode).

Regarding \exists and \forall : the old definitions could be saved before loading the package but just `\MTexistsskip{0mu}` and `\MTforallskip{0mu}` should be sufficient, except if it is important that the `\exists` and `\forall` control sequences always expand to only one token each.

3 Package options and commands

3.1 Summary of main options

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

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

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

symbolmax: all characters other than letters and digits, are taken from the Symbol font. This option also makes a number of further glyphs available, such as some basic mathematical arrows, and the sum and product signs. For documents with very simple needs in mathematical symbols, `mathastext` with option `symbolmax` may give in the end a PDF file quite smaller than the one one would get without the package.

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

defaultalphabets: by default, `mathastext` redeclares the math alphabets `\mathrm`, `\mathit`, `\mathtt` etc... (but not `\mathcal` of course) to refer to the current document text fonts (at the time of loading the package and in each `mathastext` math version). Use this option to prevent it from doing so (each alphabet also has its own disabling option).

3.2 Miscellaneous

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

amsmath: the behavior of the `\DeclareMathOperator` command of `amsmath` is slightly modified by `mathastext`. This command initially allows crazy things like `\DeclareMathOperator\crazy{m.ch-in'tr/u:c}` and then the ., -, ', / and : will be typeset in the roman font. But the font number was hardcoded in the macro and furthermore the code of `amsmath` would cause an error with Unicode engine as soon as some Unicode code is assigned to the minus character.³⁵ 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 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).

asterisk: previous versions of `mathastext` did not do anything with the `\ast` control sequence but did pick the asterisk `*` in the document text font, and this

³⁵To 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 `XYLATEX`/`LuaALATEX`. 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.

often was a rather silly thing as the text asterisk is generally in a raised position. Furthermore, the `*` lost its status of a binary operator and was treated as an ‘ordinary’ symbol. An option `noasterisk` turned this feature off. Starting with 1.2d, the `noasterisk` option is deprecated and the new default is to do nothing. But when option `asterisk` is received by the package, then both `\ast` and `*` are simultaneously modified to use (as binary operators) the text asterisk, slightly lowered. The amount of lowering³⁶ is decided by the mandatory argument to the command `\MTlowerast{<dimen>}`. The package initially does `\MTlowerast{.3\height}`. Doing `\MTlowerast{.5ex}` is not a good idea as it does not scale properly in the script and scriptscript styles. With an argument given as a multiple of `\height`, the asterisk will behave as expected in subscripts and subscripts of subscripts. But `*` is now ‘mathematically active’³⁷ and `R^*` or `$R^{\ast}` *must* be input as `$R^{\ast}` and `$R^{\ast}`. Furthermore, they will obey the math alphabet commands.

X_YTeX and Lua^ATeX: 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^ATeX (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 ^ATeX and Pdf^ATeX, 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^ATeX 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 `amsbsy` package). As it does not exist in the default ^ATeX 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`.

³⁶with the option `symbolmisc`, the asterisk is picked from the Symbol font, and the amount of lowering is non-customizable; however if a math alphabet command is used, the asterisk is then again from a text font and the lowering will be as specified by `\MTlowerast`.

³⁷in a hopefully safe way, for example `$\label{eq*1}` is ok.

- `\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.

3.3 Commands

3.3.1 Preamble-only commands

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

- `\Mathastext` (or `\mathastext`): reinitializes `mathastext` according to the current defaults of encoding, family, series and shape.³⁸
 - It can also be preceded optionally by one or more of³⁹ `\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 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.

³⁸updates 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*.

³⁹these commands exist also with long names: `\Mathastextencoding`, etc. . . The same applies to the other commands mentioned in this section.

- **math versions:** starting with version 1.12 `\Mathastext` accepts an optional argument `[\langle version_name \rangle]`, 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.⁴⁰
- `\MTWillUse[\langle ltsh \rangle]{\langle enc \rangle}{\langle fam \rangle}{\langle ser \rangle}{\langle sh \rangle}`: 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 `\langle ltsh \rangle` specifies a shape for the letters, for example `\itdefault`, or directly `\langle it \rangle` or `\langle sc \rangle`.
- `\MTDeclareVersion[\langle ltsh \rangle]{\langle name \rangle}{\langle enc \rangle}{\langle fam \rangle}{\langle ser \rangle}{\langle sh \rangle}`: declares that the document will have access to the font with the specified characteristics, under the math version name `\langle name \rangle`. For example:

$$\text{\MTDeclareVersion[sc]{palatino}{T1}{ppl}{b}{sl}}$$
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).⁴¹ When the optional argument is absent, and `mathastext` was loaded with the `italic` option, then the default letter shape will be `it`,⁴² else letters will have the same shape as used for digits and operator-names.
- `\MTboldvariant{\langle var \rangle}`: 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{\langle factor \rangle}`: scales the Euler font by `\langle factor \rangle`.
- `\MTSymbolScale{\langle factor \rangle}`: scales the Symbol font by `\langle factor \rangle`.
- `\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.
- `\MTgreekfont{\langle fontfamily \rangle}`: 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.

⁴⁰The 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.

⁴¹I do not especially recommend to use this in real life!

⁴²more 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.

3.3.2 Commands usable everywhere

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

```
\MTsetmathskips{x}{\medmuskip}{\thickmuskip}
```

```
\MTsetmathskips{A}{.5mu}{2.3mu plus .5mu minus .5mu}
```

and the effect: $vw\ x\ yzA\ BC^{vw\ x\ yz}A\ BC$. The effect obeys the usual L^AT_EX scoping rules.

The first argument of `\MTsetmathskips` may be a macro or, more generally, any expandable code, giving a letter; this facilitates use of `\MTsetmathskip` in `\@for` loops such as this one:

```
\makeatletter
```

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

```
\makeatother
```

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

- `\MTunsetmathskips{⟨a-z/A-Z⟩}`: cancels the effect of a previous `\MTsetmathskips` for that letter (this is better than setting the skips to 0mu, which would still prevent formation of ligatures when the letter is used inside a `\mathbf` for example). Example (using a version with upright Latin Modern): `\MTsetmathskips{f}{8mu}{0mu}$gffg\MTunsetmathskips{f}ff\mathbf{ff}$` gives: $g\ f\ fgff\mathbf{ff}$. The ligature was correctly formed.

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

- `\MTmathstandardletters`: this command cancels the mechanism (used by the package to insert the extra skips as specified by `\MTsetmathskips`, and the italic corrections) of ‘math activation’ of the letters.
- `\MTmathactiveletters`: this is done by default by the package (initially and each time a switch of math version occurs, except for the normal and bold

math versions under the `subdued` option). It allows the mathematical activation of the Latin letters a to z, A to Z which is used to add user-specified optional spaces around the letters, or (automatically if the font is not italic nor slanted) the italic corrections. Canceled by `\MTmathstandardletters`.

- `\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.⁴³
- `\MTnormalprime`: the prime $'$ is kept unchanged, it does not take into account the math glue specified by `\MTprimeskip`.
- `\MTlowerast{⟨dimen⟩}`: a `\raisebox` command is used to lower the text asterisk to produce a reasonable math asterisk. The package uses this command initially with argument `0.3\height`, this will have to be fine-tuned for each given text font but worked out ok with the fonts we tried. Note that the dimension argument will be used also in sub-scripts and sub-sub-scripts, so it is best not to use an absolute dimension.
- `\MTmathoperatorsobeymathxx`, `\MTmathoperatorsdonotobeymathxx`, `\MT-nonlettersobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTexplicitbrace-sobeymathxx`, `\MTexplicitbracesdonotobeymathxx`, `\MTeasynonlettersobey-`

⁴³if `\begin{document}` is not followed with `\MTversion{normal}`, the latest values set in the preamble by these commands are used at the beginning of the document, even under option `subdued`.

`mathxx`, `\MTeasynonlettersdonotobeymathxx`: these commands without argument must be used *outside* of math mode. `mathastext` issues automatically `\MTmathoperatorsobeymathxx`, `\MTeasynonlettersobeymathxx` and `\MTnonlettersdonotobeymathxx` at the time of `\usepackage{mathastext}`. See the discussion in section 2.8. The ‘explicit braces’ activation works only when `\MTnonlettersobeymathxx` has already been used.

- `\MTicinmath`: this command tells `mathastext` to add italic corrections after all letters in math mode. Its effect is cancelled inside math alphabets (except `\mathnormal`). The command is issued by default in the preamble when the shape of the letters used in math mode is neither italic nor slanted. It is reissued each time a math version is entered, under the same conditions (and except of course for the normal and bold subdued math versions). Its effect is local to the group or environment in which it has been issued.
- `\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 activate 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.⁴⁴ To cancel its effect either enclose it in a group or environment or re-issue `\MTicinmath` after it.

3.3.3 Body-only commands

- `\MTversion` [`\langle nametext \rangle`] {`\langle namemath \rangle`}, also known as `\Mathastextversion` (and as `\MTVersion`, and `\mathastextversion`): in the absence of the optional argument changes simultaneously the text and the math fonts to be the fonts corresponding to the version `\langle namemath \rangle`. If there is an optional argument then the text fonts will use `\langle nametext \rangle` and the math mode will use `\langle namemath \rangle`.

It is worth repeating that `mathastext` `\MTversion` *also* modifies the *text* font: this is due to the initial logic of the package to keep text and math in sync. To change only the math fonts, one may use the L^AT_EX command

⁴⁴it is provided for the `mathastext` defined math alphabet `\mathnormalbold` which contrarily to `\mathnormal` is not treated apart from the other math alphabets.

`\mathversion`, but this is not recommended as `\MTversion` goes through quite a few additional configuration steps (cf. [subsection 2.4](#)).

`\MTversion[⟨nametext⟩]{⟨namemath⟩}` does `\MTmathactiveletters` (except for `\MTversion{normal}` and `\MTversion{bold}` under package option `subdued`), and, if the font used in math mode is not *italic* nor *slanted*, it furthermore activates the automatic insertion of italic corrections (for better positioning of subscripts; see the discussion in [subsection 2.6](#)). Under the `frenchmath` option the package checks separately the letter shape for lower-case and uppercase.

Under the `subdued` option, `\MTversion{normal}` and `\MTversion{bold}` do `\MTmathoperatorsdonotobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTeasynonlettersdonotobeymathxx`, and `\MTmathstandardletters`.

All further commands are for math mode only.

- `\hbar`: this is constructed (in a way compatible with the `italic` option) from the `h` letter and the `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 `mathastext` 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) by option `symbolmisc`.⁴⁵ They are `\prod` \prod `\sum` \sum `\implies` \Rightarrow `\impliedby` \Leftarrow `\iff` \iff `\shortiff` \Leftrightarrow `\to` \rightarrow `\longto` \longrightarrow `\mapsto` \mapsto `\longmapsto` \longmapsto `\aleph` \aleph `\infty` ∞ `\emptyset` \emptyset `\surd` \surd `\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 \sum in inline math, and in display math the Computer Modern ones (or whatever is set up by other packages) :

$$\prod \sum$$

The package provides `\prod` and `\sum`: if one really wants in all situations the Symbol glyphs, one can do `\let\prod\prod` and `\let\sum\sum`. 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.

⁴⁵option `asterisk` is also required to treat the `*`. Recall from subsection 2.8 that the asterisk in math mode (also when using the control sequence `\ast`) appears then to T_EX to be a composite object.

3.4 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` `\MTversion{normal}` and `\MTversion{bold}` do `\MTmathoperatorsdonotobeymathxx`, `\MTeasynonlettersdonotobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTmathstandardletters`.
- **italic**: the letters default to italic shape in math mode.
- **frenchmath**: italic lowercase Latin letters, but uppercase Latin letters in the same font as for digits and operator names. In general this means that they will be upright. In case of the `LGRgreek` option, `frenchmath` influences also the shape of the Greek letters.
- **endash**, **emdash**: use the text font en-dash (–) or even the em-dash (—, but this seems crazy) for the minus sign rather than -. **endash** option is default for the package.
- **asterisk**: use the text font (or the Symbol font) asterisk in math mode.
- **noendash**: the minus sign will be the - from the text font, not the en-dash –.
- **nohbar**: prevents `mathastext` from defining its own `\hbar`.
- **nolessnomore**: besides `!?`, `,.:`, `+–=()`, `[]/`, `#$%&` `mathastext` treats also `<>|`, `{ }` and `\`. Use this option to let it not do it. This is the default in case of OT1-encoding.
- further excluding options: **noexclam** `!?`, **nopunctuation**, `,.:`, **noplus**, **nominus**, **noplusnominus** `+–`, **noequal** `=`, **noparenthesis** `()[]/`, **nospecials** `#$%&` and **nodigits**.
- **alldelims**: true by default, means that the characters excluded by **nolessnomore** are treated. Use this option in case of a mono-width OT1-encoded font.
- **nosmalldelims**: this prevents `mathastext` from trying to pick up in the text font the ‘small variants’ of some math delimiters; it only affects what happens when a character such as a left parenthesis (or [is used as a delimiter, and \TeX has chosen the smallest sized variant. This has no impact on what happens when they are not used as delimiters: then, and if not disabled by the corresponding options, these characters are always picked up from the text font.

- `symbolgreek`, `symboldigits`: to let Greek letters (digits) use the Symbol font.
- `symbolre`: replaces `\Re` and `\Im` by the Symbol glyphs \Re , \Im and defines a `\DotTriangle` command ($\dot{\triangle}$).
- `symbolmisc`: takes quite a few glyphs, including logical arrows, product and sum signs from Symbol. They are listed *supra*. Doing `\renewcommand{\int}{\smallint}` will maximize even more the use of the Symbol font.
- `symboldelimiters`: the characters apart from letters and digits will be taken from the Symbol font.
- `symbol`: combines `symbolgreek`, `symbolre`, and `symbolmisc`.
- `symbolmax`: combines `symbol` and `symboldelimiters`.
- `eulergreek`, `eulerdigits`: to let Greek letters (digits) use the Euler font.
- `LGRgreek`: this is for a font which is also available in LGR-encoding. It is possible to change the font per math version, via the use of the `\MTgreekfont` command in the preamble.
- `LGRgreeks`: each declared math version will be supposed to be with a font which is also available in LGR-encoding.
- `selfGreek`: this is for a font which is also available in OT1-encoding and contains the glyphs for the default eleven capital Greek letters.
- `selfGreeks`: each declared math version will be supposed to be with a font with the eleven capital Greek letters in its OT1-encoded version.
- `upgreek`, `itgreek`, `upGreek`, `itGreek`: options to tell to use `\itdefault` or `\updefault` for the lowercase and uppercase (or only the uppercase) Greek letters. Only operant in the case of the `LGRgreek(s)` and `selfGreek(s)` options.
- `mathaccents`: use the text font also for the math accents. As in vanilla \LaTeX , they are taken from the font for the digits and `\log`-like names. Obey the alphabet changing commands. Will work only for T1, LY1, or OT1-compatible encodings.
- `defaultbf`, `defaultit`, `defaultsf`, `defaulttt`: do not set up, respectively, the `\mathbf`, `\mathit`, `\mathsf`, and `\mathtt` commands to use the mathastext-ified font. This also prevents `mathastext` to create internally `\Mathxx` alphabets (it never overwrites the original `\mathxx` things but let `\mathxx` point to `\Mathxx` instead), so one can use these options if one encounters a ‘too many math alphabets’ \LaTeX error.

- `defaulnormal`, `defaultrm`: do not identify the default `\mathnormal` (resp. `\mathrm`) with the newly created `\Mathnormal` (resp. `\Mathrm`) commands which use the mathastextified fonts in each math version.
- `defaultalphabets`: all the `defaultxx` options together, and additionally tells `mathastext` not to create the `\mathnormalbold` alphabet either.
- `defaultimath`: do not overwrite `\imath` and `\jmath`, do not extend `\i` and `\j` to math mode use.
- `defaultmathsizes`: do not change the 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>

4 Change log

1.3a [2013/09/04]

- * the somewhat silly `\string's` are removed from the `\MTsetmathskips` command of release 1.3, thus allowing its first argument to be a macro, or any expandable code, giving a letter.
- * the `amsmath \resetMathstrut@`, which is incompatible with a mathematically active parenthesis (is now modified only if necessary (i.e. only when `\MTnonlettersobeymathxx` is issued) and is restored to its original value if not needed anymore (i.e. after `\MTnonlettersdonotobeymathxx`, as for example when switching to the normal version under option `subdued`).
- * improved documentation.

1.3 [2013/09/02]

- * commands `\MTsetmathskips` and `\MTunsetmathskips` added.
- * commands `\MTmathactiveletters` and `\MTmathstandardletters` to govern the math activation of letters independently of its use for insertion of the italic corrections (`\MTicinmath` and `\MT-`

`noicinmath` correspondingly modified).

- * the new `\luatexUmathcodenum` as available since TL2013 allows identical treatment by `mathastext` of `=` and `-` under both LuaTeX and XeTeX.
- * `\newmcodes@` of `amsmath` is left untouched in case of option `basic`.
- * a sentence containing `|` which was written to the log during the loading caused a problem if `|` was active (typically if `\MakeShortVerb{\|}` was added to the preamble prior to the loading of `mathastext`).
- * some preemptive measures taken regarding things such as `\mid`, `\lbrace`, and `\rbrace`, as some packages define these things in manners which made the re-definitions done by `mathastext` issue errors.

1.2f [2013/01/21]

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

1.2e [2013/01/10]

This version should be the last one in the 1.2 se-

ries as it seems to correct most of the main problems which were introduced with the massive use of mathematically active characters in versions 1.2 and 1.2b.

- * It is indeed a thorny point when one wants to modify only in math mode how an active character acts, without breaking things. The package now does that /only/ if the activation appears to originate in the Babel system, as it is then possible to modify appropriately the Babel macros `\user@active<char>` and `\normal@char<char>`. The relevant issues are discussed in section 2.10 of the user manual, in the test file `mathastexttestalphabets.tex`, and in the source code comments to the macro `\mst@mathactivate`. The inherent incompatibility of Babel with packages having made mathematically active the characters itself makes document active is circumvented by this interference of `mathastext`. A generally applicable Babel patch could be derived from the method used by `mathastext`.

- * The technique of mathematical activation is maintained only for the characters which are not catcode active (at the entrance in math mode, as `mathastext` does all its activation job at `everymath` and `everydisplay`).

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

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

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

- * The code is better documented.

1.2d [2013/01/02]

- * an incompatibility with `amsmath` (its macro `\resetMathstrut@`), exists since version 1.2 of the package. This is fixed here.

- * various improvements in dealing with the asterisk and in the mechanism of letting non-letter symbols obey the math alphabet commands.

- * documentation extended and improved.

1.2c [2012/12/31]

- * `mathastext` now inserts automatically after all (latin) letters in math mode their italic corrections, if the font used is upright (sic). This improves the spacings for the positioning of subscripts. The feature is de-activated inside the math alphabets com-

mands (apart from `\mathnormal`), so as to not prohibit the formation of ligatures,

- * the documentation has been extended to explain in detail the issues which are relevant to the new feature of added italic corrections,

- * version 1.2 had some bad bugs when confronted to active characters. This is corrected and additionally `\MTnonlettersdonotobeymathxx` is made the default, as the user input is too much constrained in its absence.

- * a less fatal, but still annoying, typo had made the dot in 1.2 of type `\mathpunct` rather than `\mathord`

- * the inner namespace has been rationalized a bit.

1.2 [2012/12/20]

- * a new command sets up the amount of space to be automatically inserted before the derivative glyph (useful when using an upright font).

- * the scope of the math alphabets has been extended to apply to the non-alphabetical characters, and also to operator names.

- * the format of the dtx file has changed. The package file is self-extracting from the dtx, and four additional test files are also produced during 'latex `mathastext.dtx`'.

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

- * `\$`, `\#`, `\&`, and `\%` had been re-defined by `mathastext` since its inception in a rather strange (but working) way, which could cause surprises to other packages. Fixed.

- * the subdued mechanism for the math alphabets is implemented in a simpler and more efficient manner than in 1.15e.

- * the 'defaultxx' options act a bit differently, and are more useful in case of a 'too many math alphabets' situation.

- * various improvements in the documentation.

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

1.15e [2012/10/22]

- * new user commands to specify skip or glue to be inserted after the math symbols `\exists` and `\forall`

- * complete (user transparent) rewrite of the code implementing the subdued option; and its action has been extended to apply also to the `\mathbf`, `\mathit`, `\mathsf`, `\mathtt` alphabets and not only to `\mathrm` and `\mathnormal` as in the previous versions.

- * improvements in the documentation.
- 1.15d [2012/10/13]
- * the Unicode situation is now correctly treated, throughout the code (this had been left in a half-done way from version 1.14 of April 2011).
 - this includes an issue related to amsmath and its DeclareMathOperator macro which has been fixed,
 - and the code related to `\relbar` and `\Relbar` (and `\models`) has been revised.
- 1.15c [2012/10/05]
- * it is now possible to use distinct fonts in LGR encoding for the Greek letters according to the current math version.
 - * improvements to the documentation.
- 1.15b
- * corrected a 'feature' of 1.15 which was backward- incompatible
 - * improvements to the pdf documentation
- 1.15 [2012/09/26]
- * the subdued option allows the mathastextification to act only locally.
 - * some measures taken to deal with amsmath related issues when using xetex or luatex.
- 1.14c
- * a bug is fixed: the `\Mathastext` macro reinitializes the fonts in the normal and bold math versions, but it also erroneously redeclared the math alphabet changing commands which could have been set up in previously defined math versions (via earlier calls to `\Mathastext[version_name]`).
- 1.14b [2011/04/03]
- * there was a bug with `\$, \#, \&, \%` in math mode which showed up when ten or more math families had been declared. This bug affected also the minus sign under the same circumstances, when Unicode engines were used. Fixed.
 - * the options LGRgreek and selfGreek act now a bit differently, and new options LGRgreeks and selfGreeks have been defined.
 - * I also cleaned up a bit the code, for a more structured namespace.
- 1.14
- * mathastext now modifies also the math alphabets `\mathit`, `\mathsf` and `\mathtt`, thus making it a quite generic complete manner to adapt the math configuration to fonts provided with no math support.
- 1.13b
- * when the Symbol font is used for `\prod` and `\sum` this will be only for inline math; display math will use the default glyphs
- 1.13 [2011/03/11]
- * the LGRgreek option is added.
 - * internal changes for better readability of the code.
- 1.12
- * various bugs have been corrected.
 - * the endash and alldelims options are active by default.
 - * the package is more Unicode aware.
 - * the `\Mathastext` command has been improved to facilitate the mechanism of math versions also when using XeTeX or LuaTeX (with package fontspec.)
 - * the en-dash and dotless i and j now work with all encodings, Unicode inclusive.
- 1.11 [2011/02/06] optional argument to `\Mathastext` macro.
- 1.1 [2011/02/01] options italic and frenchmath.
- 1.0 [2011/01/25] Initial version.

5 Implementation

```
1 \ProvidesPackage{mathastext}
2 [2013/09/04 v1.3a Use the text font in math mode (jfB)]
```

The usual catcode regime for letters and digits is assumed and some characters such as *, ‘, ", = are supposed to be of catcode other at the time of loading of **mathastext**. The source of **mathastext** takes precautions for some other characters such as the right quote ’, which may thus be active with no harm at the time of loading. By the way, I think L^AT_EX₂ε should have provided to authors a standard macro to be used at the beginning of a style file to make sure the catcodes are standard. Shorthands created by Babel should be mostly no problem as Babel does the activation only at the `\begin{document}`.

The comments have been accumulating through successive versions with only partial efforts to achieve some sort of coherence; as a result some are a bit strange or obsolete to various degrees.

```
3 \NeedsTeXFormat{LaTeX2e}
```

Testing for XeTeX and LuaLaTeX. I should load some package for this code which I copied somewhere (perhaps from the `iftex` package? or some post on `comp.text.tex`). 2013/01/01: at least the code now does not overwrite previous `\ifXeTeX` or `\ifLuaTeX`...

```
4 \newif\ifmst@XeTeX
5 \begingroup\expandafter\expandafter\expandafter\endgroup
6 \expandafter\ifx\csname XeTeXinterchartoks\endcsname\relax
7   \mst@XeTeXfalse
8 \else
9   \mst@XeTeXtrue
10 \fi
11 \newif\ifmst@LuaTeX
12 \begingroup\expandafter\expandafter\expandafter\endgroup
13 \expandafter\ifx\csname directlua\endcsname\relax
14   \mst@LuaTeXfalse
15 \else
16   \mst@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
\mst@ser
\mst@opsh
\mst@bold
\mst@ltsh
23 \def\mst@enc{\encodingdefault}
24 \def\mst@fam{\familydefault}
25 \def\mst@ser{\seriesdefault}
26 \def\mst@opsh{\shapedefault} %% will be default shape for operator names
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@asterisk
66   \DeclareOption{noasterisk}{\PackageWarningNoLine{mathastext}
67     {option ‘noasterisk\string’ is deprecated.^^J\space\space\space
68       Check the documentation}}
69   \DeclareOption{asterisk}{\mst@asterisktrue}
70   \AtBeginDocument{
71     \everymath\expandafter{\the\everymath \mst@doasterisk

```

```

72 \let\mst@doasterisk\relax}
73 \everydisplay\expandafter{\the\everydisplay \mst@doasterisk
74 \let\mst@doasterisk\relax}}
75 \newif\ifmst@nospecials
76 \DeclareOption{nospecials}{\mst@nospecialstrue}
77 \newif\ifmst@basic % 1.3 to avoid unnecessary patch of amsmath \newmcodes@
78 \DeclareOption{basic}{\mst@basictrue
79 \ExecuteOptions{noparenthesis,nopunctuation,%
80 noplusnominus,noequal,noexclam,nospecials,nolessnomore}}
81 %%
82 \newif\ifmst@nohbar
83 \DeclareOption{nohbar}{\mst@nohbartrue}
84 \newif\ifmst@nodigits
85 \DeclareOption{nodigits}{\mst@nodigitstrue}
86 \newif\ifmst@defaultimath
87 \DeclareOption{defaultimath}{\mst@defaultimathtrue}
88 \newif\ifmst@mathaccents
89 \DeclareOption{mathaccents}{\mst@mathaccentstrue}
90 %%
91 \newif\ifmst@needsymbol
92 \newif\ifmst@symboldelimiters
93 \DeclareOption{symboldelimiters}{\mst@needsymboltrue\mst@symboldelimiterstrue}
94 \newif\ifmst@symboldigits
95 \DeclareOption{symboldigits}{\mst@needsymboltrue\mst@symboldigitstrue}
96 \newif\ifmst@symbolgreek
97 \DeclareOption{symbolgreek}{\mst@needsymboltrue\mst@symbolgreektrue}
98 \newif\ifmst@symbolre
99 \DeclareOption{symbolre}{\mst@needsymboltrue\mst@symbolretrue}
100 \newif\ifmst@symbolmisc
101 \DeclareOption{symbolmisc}{\mst@needsymboltrue\mst@symbolmisctrue}
102 \DeclareOption{symbol}{\ExecuteOptions{symbolgreek,symbolmisc,symbolre}}
103 \DeclareOption{symbolmax}{\ExecuteOptions{symbol,symboldelimiters}}
104 %%
105 \newif\ifmst@needeuler
106 \newif\ifmst@eulerdigits
107 \DeclareOption{eulerdigits}{\mst@needeulertrue\mst@eulerdigitstrue}
108 \newif\ifmst@eulergreek
109 \DeclareOption{eulergreek}{\mst@needeulertrue\mst@eulergreektrue}
110 %%
111 \newif\ifmst@selfGreek
112 \DeclareOption{selfGreek}{\mst@selfGreektrue}
113 \newif\ifmst@selfGreeks
114 \DeclareOption{selfGreeks}{\mst@selfGreektrue\mst@selfGreektrue}
115 \newif\ifmst@LGRgreek
116 \DeclareOption{LGRgreek}{\mst@LGRgreektrue}
117 \newif\ifmst@LGRgreeks
118 \DeclareOption{LGRgreeks}{\mst@LGRgreektrue\mst@LGRgreektrue}
119 %%
120 \def\mst@greek@select{0}

```

```

121 \newif\ifmst@itgreek
122 \newif\ifmst@upgreek
123   \DeclareOption{itgreek}{\mst@itgreektrue}
124   \DeclareOption{upgreek}{\mst@upgreektrue}
125   \DeclareOption{itGreek}{\def\mst@greek@select{1}}
126   \DeclareOption{upGreek}{\def\mst@greek@select{2}}
127 %%

```

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.

```

128 \newif\ifmst@defaultnormal
129   \DeclareOption{defaultnormal}{\mst@defaultnormaltrue}
130 \newif\ifmst@defaultterm
131   \DeclareOption{defaultterm}{\mst@defaulttermtrue}
132 \newif\ifmst@defaultbf
133   \DeclareOption{defaultbf}{\mst@defaultbftrue}
134 \newif\ifmst@defaultit
135   \DeclareOption{defaultit}{\mst@defaultittrue}
136 \newif\ifmst@defaultsf
137   \DeclareOption{defaultsf}{\mst@defaultsftrue}
138 \newif\ifmst@defaulttt
139   \DeclareOption{defaulttt}{\mst@defaulttttrue}
140 \newif\ifmst@nonnormalbold
141 \DeclareOption{defaultalphabets}{\ExecuteOptions{defaultnormal,defaultterm,%
142 defaultbf,defaultit,defaultsf,defaulttt}\mst@nonnormalboldtrue}

```

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

```

143 \newif\ifmst@defaultsizes
144   \DeclareOption{defaultmathsizes}{\mst@defaultsizestrue}
145 \newif\ifmst@twelve
146   \DeclareOption{12pt}{\mst@twelvetrue}
147 \newif\ifmst@fouriervec
148   \DeclareOption{fouriervec}{\mst@fouriervectrue}

```

1.15: the subdued option.

```

149 \newif\ifmst@subdued
150   \DeclareOption{subdued}{\mst@subduedtrue}
151 \DeclareOption*{\PackageWarningNoLine{mathastext}
152   {Unknown option ‘\CurrentOption\string’}}

```

```

153 \ProcessOptions\relax

```

<pre> \exists \mst@exists@skip \forall \mst@forall@skip </pre>	<p>1.15e [2012/10/21]: <code>math skip/glue</code> <i>after</i> <code>\exists</code> and <code>\forall</code>, 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 <code>symbol</code> option will also want to redefine these math symbols.</p>
--	---

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.

```

154 \newmuskip\mst@exists@muskip %% v 1.15e
155 \newmuskip\mst@forall@muskip
156 \def\mst@exists@skip{1mu}
157 \def\mst@forall@skip{.6667mu}
158 \ifmst@italic\ifmst@frenchmath\else
159     \def\mst@exists@skip{0mu}\def\mst@forall@skip{0mu}\def\mst@prime@skip{0mu}
160 \fi\fi
161 \AtEndOfPackage{% must be at end of code
162 \let\mst@stdexists\exists
163 \let\mst@stdforall\forall
164 \renewcommand\exists{\mst@stdexists\mskip\mst@exists@muskip}
165 \renewcommand\forall{\mst@stdforall\mskip\mst@forall@muskip}
166 }

```

`\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).
`\mst@prime@skip` glyph and upright letters (in contrast the prime from `txfonts` works fine with upright letters).
`\active@math@prime` For this we replace the L^AT_EX kernel `\active@math@prime` with our own skip-enhanced version
`\MTnormalprime` `\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\bb1@prim@s` when `'` is made active via its services (the czech and slovak languages also store the initial version of `\prim@s`, else the quote would not work correctly when being again of `catcode 12`), and it doesn't matter if `mathastext` is loaded before or after this happens, as the `\mst@mathactivate` does its job only as part of the `\everymath` and `\everydisplay` token lists.

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

```

167 \newmuskip\mst@prime@muskip %% v 1.2
168 \def\mst@prime@skip{.5mu}
169 \ifmst@italic\ifmst@frenchmath\else\def\mst@prime@skip{0mu}\fi\fi
170 \def\mst@active@math@prime{\sp\bgroup\mskip\mst@prime@muskip\prim@s}
171 {\catcode'\'=12
172 \gdef\mst@modifyprime{\mst@mathactivate'}\mst@active@math@prime}}
173 \newcommand*\MTnormalprime{\let\mst@modifyprime\relax}
174 \AtBeginDocument{
175 \everymath\expandafter{\the\everymath \mst@modifyprime \MTnormalprime}
176 \everydisplay\expandafter{\the\everydisplay \mst@modifyprime \MTnormalprime}
177 }

```

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

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

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

```

178 \newcommand*\MTexistsskip[1]{\edef\mst@exists@skip{#1}%
179   \mst@exists@muskip\mst@exists@skip\relax}
180 \newcommand*\MTforallskip[1]{\edef\mst@forall@skip{#1}%
181   \mst@forall@muskip\mst@forall@skip\relax}
182 \newcommand*\MTprimeskip[1]{\edef\mst@prime@skip{#1}%
183   \mst@prime@muskip\mst@prime@skip\relax}
184 \let\Mathastextexistsskip\MTexistsskip
185 \let\Mathastextforallskip\MTforallskip
186 \let\Mathastextprimeskip\MTprimeskip
187 \let\mathastextexistsskip\MTexistsskip
188 \let\mathastextforallskip\MTforallskip
189 \let\mathastextprimeskip\MTprimeskip

```

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

```

\mathchardef\@tempa\mathcode'\(\relax

```

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

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

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

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

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

```

190 \ifmst@noparen\else
191 \AtBeginDocument{%
192   \@ifundefined{resetMathstrut@}{% nothing to do, no amsmath
193   }{% amsmath loaded, and possibly patched by things such as lualatex-math
194     \let\mst@savdresetMathstrut@\resetMathstrut@
195     \PackageWarning{mathastext}{current meaning of amsmath
196     \string\resetMathstrut@\space saved}%
197     \ifx\mst@the\the % means that \MTnonlettersobeymathxx was used in preamble
198       \let\mst@the@gobble\MTnonlettersobeymathxx

```

```
199 \fi}}
200 \fi
```

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

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

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

1.2c [2012/12/31]: hence, this current implementation puts constraints on the input: `$x^?$` or `$x\mathrel?y$` now create errors. They must be input `$x^{?}$`, respectively `$x\mathrel{?}y$`.

The disactivating macro `\MTnonlettersdonotobeymathxx` is made the default.

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

`\mst@mathactivate`

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

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

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

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

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

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

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

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

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

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

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

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

(*) it is still important to not do twice the thing when the character is active, in which case the `babel` macros are patched.

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

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

1.2e [2013/01/10] corrects a bad oversight of 1.2b in `\mst@mathactivate` which made the reproduction of the user manual illustrations with `f_i` impossible. As `\mst@mathactivate` was used also to get the non-letters obey math alphabet while maintaining the TeX spacings, I could not use group braces. I should have added them later in the case of letters but then forgot. Fixed. (the group braces do not prevent ligatures when the letters are arguments to the math alphabet commands, the added macro `\mst@itcorr` then expands to nothing).

`\MTnoicinmath` can also be used from inside math mode.

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

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

allowed.

<code>\MTmath-</code> <code>activeletters</code> <code>\MTmath-</code> <code>standardletters</code>	<p>1.3 [2013/09/02] extends the use of mathematically active letters to allow the user to specify <code>\muglue</code> before and after the letter itself (see <code>\MTsetmathskips</code>, below). Mathematically active letters were previously used only to add the italic correction; the math activation has now been separated and put in <code>\MTmathactiveletters</code>. There is also <code>\MTmathactiveLetters</code> to allow math activation only for the uppercase letters. To cancel the (now default, even with option <code>italic</code>) math activation of letters, there is <code>\MTmathstandardletters</code>. Version 1.3a removes some silly <code>\string</code>'s from the code, which prevented to pass macros as first argument to the command.</p>
<code>\MTnonletters-</code> <code>obeymathxx</code> <code>\MTnonletters-</code> <code>donotobeymathxx</code>	<p>These macros are modified in version 1.3a [2013/09/04] in order to cleverly adjust, or not, the <code>\amsmath\resetMathstrut@</code>. When used in the preamble, they just modify <code>\mst@the</code>. And there is code at begin document to check the status there of <code>\mst@the</code> and if its meaning is <code>\the</code>, then <code>\MTnonlettersobeymathxx</code> is activated again to do the patch. When used in the body they adjust <code>\resetMathstrut@</code>. Notice that the saved meaning is the one at begin document (thus, possibly patched by <code>\lualatex-math</code>) but modifications done after that would not be seen in <code>\mst@saveresetMathstrut@</code>.</p>

```

201 \newtoks\mst@do@nonletters
202 \newtoks\mst@do@easynonletters
203 \newtoks\mst@do@az
204 \newtoks\mst@do@AZ
205 \let\mst@the\@gobble
206 \newcommand*\MTnonlettersdonotobeymathxx{%
207     \ifx\mst@the\@gobble
208     \else
209         \@ifundefined{mst@saveresetMathstrut@}{}{%
210             \PackageWarning{mathastext}{restoring (for this group or environment) ams-
math \string\resetMathstrut@}%
211             \let\resetMathstrut@\mst@saveresetMathstrut@}%
212         \fi
213         \let\mst@the\@gobble
214 }
215 \newcommand*\MTnonlettersobeymathxx{%
216     \ifx\mst@the\the
217     \else
218         \@ifundefined{mst@saveresetMathstrut@}{}{%
219             \ifmst@symboldelimiters
220                 \def\resetMathstrut@{%
221                     \setbox\z@\hbox{\the\textfont\symmtpsymbol\char40
222                         \the\textfont\symmtooperatorfont\char40
223                         \the\textfont\symoperators\char40}%
224                     \ht\Mathstrutbox@ \ht\z@ \dp\Mathstrutbox@ \dp\z@}%
225             \else
226                 \def\resetMathstrut@{%
227                     \setbox\z@\hbox{\the\textfont\symmtooperatorfont\char40
228                         \the\textfont\symoperators\char40}%
229                     \ht\Mathstrutbox@ \ht\z@ \dp\Mathstrutbox@ \dp\z@}%
230             \fi
231             \PackageWarning{mathastext}{\string\resetMathstrut@\space

```

```

232     from amsmath replaced (for this group or environment)}}%
233     \fi
234     \let\mst@the\the
235 }
236 \newcommand*\MTeasynonlettersdonotobeymathxx{\let\mst@theeasy@gobble}
237 \newcommand*\MTeasynonlettersobeymathxx{\let\mst@theeasy\the}
238 \MTeasynonlettersobeymathxx
239 \newcommand*\MTmathactiveletters{\let\mst@thef\the \let\mst@theF\the}
240 \newcommand*\MTmathactiveLetters{\let\mst@theF\the}
241 \newcommand*\MTmathstandardletters{\let\mst@thef@gobble \let\mst@theF@gobble}
242 \newcommand*\MTicinmath{%
243     \MTmathactiveletters
244     \def\mst@itcorr{\ifnum\fam=\m@ne\else\ifnum\fam=\symmtletterfont\fi}%
245     \let\mst@ITcorr\mst@itcorr}
246 \newcommand*\MTICinmath{%
247     \MTmathactiveLetters
248     \def\mst@ITcorr{\ifnum\fam=\m@ne\fi}}
249 \newcommand*\MTnoicinmath{\def\mst@itcorr{}\def\mst@ITcorr{}}
250 \newcommand*\MTnoICinmath{\def\mst@ITcorr{}}
251 \newcommand*\MTicalsoinmathxx{%
252     \ifx\mst@itcorr@empty\else\def\mst@itcorr{\/}\fi
253     \ifx\mst@ITcorr@empty\else\def\mst@ITcorr{\/}\fi}
254 \AtBeginDocument{
255 \everymath\expandafter{\the\everymath
256     \mst@the\mst@do@nonletters \let\mst@the@gobble
257     \mst@theeasy\mst@do@easynonletters \let\mst@theeasy@gobble
258     \mst@thef\mst@do@az \let\mst@thef@gobble
259     \mst@theF\mst@do@AZ \let\mst@theF@gobble}
260 \everydisplay\expandafter{\the\everydisplay
261     \mst@the\mst@do@nonletters \let\mst@the@gobble
262     \mst@theeasy\mst@do@easynonletters \let\mst@theeasy@gobble
263     \mst@thef\mst@do@az \let\mst@thef@gobble
264     \mst@theF\mst@do@AZ \let\mst@theF@gobble}
265 }

```

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

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

The command `\MTmathstandardletters` turns off math activation and its effects for all letters.

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

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

```
266 \newcommand*\MTsetmathskips[3]{%
267   \@namedef{mst@before#1}{\mskip#2\relax}%
268   \@namedef{mst@after#1}{\mskip#3\relax}%
269 }
270 \newcommand*\MTunsetmathskips[1]{%
271   \@namedef{mst@before#1}{\relax}%
272   \@namedef{mst@after#1}{\relax}%
273 }
```

```
\mst@mathactivate
\addtodo@nonletters 274 \def\mst@magic@v #1#2#3#4#5{#1#3#4}
\do@easynonletters 275 \def\mst@magic@vi #1#2#3#4#5#6{#1#2#4#5}
\mst@addtodo@az 276 \def\mst@fork{\ifmmode\mst@magic@v\fi\@thirdofthree}
\mst@addtodo@AZ 277 \def\mst@safefork{\ifmmode\if@safe@actives\else\mst@magic@vi\fi\fi\@thirdofthree}
278 \def\mst@do@activecase#1#2#3{% #1 is a category 11 or 12 character token
279   \@ifundefined{active@char#1}{}%
280     \ifcat #1a\mathcode'#1=#2\else
281     \ifx\relax #2\relax\mathcode'#1='#1 \else\mathcode'#1=#3\fi\fi
282   \expandafter\expandafter\expandafter\let\expandafter\expandafter
283     \csname mst@orig@user@active#1\endcsname
284     \csname user@active#1\endcsname
285   \expandafter\expandafter\expandafter\let\expandafter\expandafter
286     \csname mst@orig@normal@char#1\endcsname
287     \csname normal@char#1\endcsname
288   \ifcat #1a%
289     \expandafter\edef\csname user@active#1\endcsname
290       {\noexpand\mst@fork {{#2}\noexpand#3}}}\expandafter
291       \noexpand\csname mst@orig@user@active#1\endcsname
292     }%
293   \expandafter\edef\csname normal@char#1\endcsname
294     {\noexpand\mst@safefork {{#2}\noexpand#3}}}\expandafter
295     \noexpand\csname mst@orig@normal@char#1\endcsname
296   }%
297   \else
298     \expandafter\edef\csname user@active#1\endcsname
299       {\noexpand\mst@fork {#2}\noexpand#3}\expandafter
300       \noexpand\csname mst@orig@user@active#1\endcsname
301     }%
302   \expandafter\edef\csname normal@char#1\endcsname
303     {\noexpand\mst@safefork {#2}\noexpand#3}\expandafter
304     \noexpand\csname mst@orig@normal@char#1\endcsname
305   }%
306   \fi}}
307 \begingroup
```

```

308 \catcode'\~=\active
309 \def\x{\endgroup
310 \def\mst@mathactivate##1##2##3{% ##1 guaranteed of cat 11 or 12
311 \begingroup
312 \lccode'\~='##1
313 \lccode'##1='##1
314 \lowercase{\endgroup
315 \ifnum\catcode'##1=\active
316 \mst@do@activecase ##1{##2}##3
317 % careful as ##2 is empty in the asterisk and
318 % prime case!
319 \else
320 \mathcode'##1="8000
321 % version 1.3 adds the possibility of extra skips around letters,
322 % (only if non catcode active at the time of use).
323 \ifcat##1a\def~{{\csname mst@before##1\endcsname ##2##3%
324 \csname mst@after##1\endcsname}}}%
325 \else\def~{##2##3}\fi
326 \fi}}
327 \x
328 \def\mst@addtodo@nonletters#1#2#3{%
329 % #1 will be of cat 11 or 12.
330 % #2 is empty for asterisk and right quote
331 \mst@do@nonletters\expandafter
332 {\the\mst@do@nonletters \mst@mathactivate#1{#2}#3}%
333 }
334 \def\mst@addtodo@easynonletters#1#2{% #1 is a one char control sequence
335 \mst@do@easynonletters\expandafter{\the\mst@do@easynonletters\mathcode'#1=#2}%
336 }
337 \def\mst@addtodo@az#1#2#3{%
338 \mst@do@az\expandafter{\the\mst@do@az\mst@mathactivate#1#2#3}
339 }
340 \def\mst@addtodo@AZ#1#2#3{%
341 \mst@do@AZ\expandafter{\the\mst@do@AZ\mst@mathactivate#1#2#3}%
342 }

```

`\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).

1.3: now tests if 'basic' option was used.

```

343 \ifmst@basic\else
344 \ifpackageloaded{amsmath}
345 {\let\newmcodes@relax} % brutal, but bye bye and don't come back.
346 {}
347 \fi

```

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 L^AT_EX 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 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 `\XeTeX-mathcodenum` always return a Unicode mathcode.

UPDATE for `mathastext` 1.3: since the release of `lualatex` as included in TL2013, `\luatexUmathcodenum` behaves as `\XeTeXmathcodenum` so `mathastext` 1.3 treats identically under both unicode engines the equal and minus signs (and the vertical bar).

1.15e: Oct 22, 2012. I add the necessary things to also subdue the `\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^LA^TE^X 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.

```

348 \ifmst@subdued
349   \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
350     \def\mst@normalmv@letter{#2}}
351   \expandafter\mst@reserved\mv@normal\@nil
352   \def\mst@reserved#1\getanddefine@fonts\symletters#2#3\@nil{%
353     \def\mst@boldmv@letter{#2}}
354   \expandafter\mst@reserved\mv@bold\@nil
355   \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
356     \def\mst@normalmv@operator{#2}}
357   \expandafter\mst@reserved\mv@normal\@nil
358   \def\mst@reserved#1\getanddefine@fonts\symoperators#2#3\@nil{%
359     \def\mst@boldmv@operator{#2}}
360   \expandafter\mst@reserved\mv@bold\@nil
361 %
362   \edef\mst@tmp{\encodingdefault}
363   \def\mst@reserved#1/#2/#3/#4/{\gdef\mst@debut{#1}\gdef\mst@reste{#2/#3/#4}}
364   \begingroup\escapechar\m@ne
365     \xdef\mst@funnyoti{\expandafter\string\cename OT1\endcename}
366     \expandafter\expandafter\expandafter

```

```

367         \mst@reserved\expandafter\string\mst@normalmv@operator/
368     \endgroup
369     \ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
370     \edef\mst@normalmv@operator{\expandafter\noexpand\csname
371         \mst@tmp/\mst@reste\endcsname}
372     \begingroup\escapechar\m@ne
373         \expandafter\expandafter\expandafter
374         \mst@reserved\expandafter\string\mst@boldmv@operator/
375     \endgroup
376     \ifx\mst@debut\mst@funnyoti\ifx\mst@tmp\mst@oti\def\mst@tmp{T1}\fi\fi
377     \edef\mst@boldmv@operator{\expandafter\noexpand\csname
378         \mst@tmp/\mst@reste\endcsname}
379     \AtEndOfPackage{
380         \typeout{** ...entering subdued mode...}
381         \expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@letter\symmtletterfont
382         \expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@letter\symmtletterfont
383         \expandafter\SetSymbolFont@ \expandafter\mv@normal\mst@normalmv@operator\symmtoperatorfont
384         \expandafter\SetSymbolFont@ \expandafter\mv@bold\mst@boldmv@operator\symmtoperatorfont
385         \typeout{** ...done.}
386     }
387 \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.

```

388 \DeclareSymbolFont{mtoperatorfont}
389     {\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-
donot-
obeymathxx

```

390 \newcommand*{\MTmathoperatorsobeymathxx}
391 {\def\operator@font{\mathgroup\ifnum\fam=\m@ne\symmoperatorfont\else\fam\fi}}
392 \newcommand*{\MTmathoperatorsdonotobeymathxx}
393 {\def\operator@font{\mathgroup\symmoperatorfont}}
394 \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.

```

395 \DeclareSymbolFont{mtletterfont}
396      {\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 alphabets is recorded for the sake of the subdued option.

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

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

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

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

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

```

397 \let\mst@alph@omicron\mathnormal
398 \@ifpackageloaded{fourier}{\ifslowed\else\let\mst@alph@omicron\mathrm\fi}{}
399 \DeclareSymbolFontAlphabet{\Mathnormal}{mtletterfont}
400 \DeclareSymbolFontAlphabet{\Mathrm}{moperatorfont}
401 \ifmst@nonormalbold\else
402   \DeclareMathAlphabet{\mathnormalbold}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@ltsh}
403 \fi\ifmst@defaultbf\else
404   \DeclareMathAlphabet{\Mathbf}{\mst@enc}{\mst@fam}{\mst@bold}{\mst@opsh}
405 \fi\ifmst@defaultit\else
406   \DeclareMathAlphabet{\Mathit}{\mst@enc}{\mst@fam}{\mst@ser}{\itdefault}
407 \fi\ifmst@defaultsf\else
408   \DeclareMathAlphabet{\Mathsf}{\mst@enc}{\sfdefault}{\mst@ser}{\mst@opsh}
409 \fi\ifmst@defaulttt\else
410   \DeclareMathAlphabet{\Mathtt}{\mst@enc}{\ttdefault}{\mst@ser}{\mst@opsh}
411 \fi
412 \let\mst@original@normal\mathnormal
413 \let\mst@original@rm\mathrm
414 \let\mst@original@bf\mathbf
415 \let\mst@original@it\mathit
416 \let\mst@original@sf\mathsf
417 \let\mst@original@tt\mathtt
418 \def\mst@restorealphabets{% for subdued
419   \let\mathnormal\mst@original@normal
420   \let\mathrm\mst@original@rm
421   \let\mathbf\mst@original@bf
422   \let\mathit\mst@original@it
423   \let\mathsf\mst@original@sf
424   \let\mathtt\mst@original@tt}

```

```

425 \def\mst@setalphabets{%
426   \ifmst@defaultnormal\else\renewcommand{\mathnormal}{\Mathnormal}\fi
427   \ifmst@defaulttrm\else\renewcommand{\mathrm}{\Mathrm}\fi
428   \ifmst@defaultbf\else\renewcommand{\mathbf}{\Mathbf}\fi
429   \ifmst@defaultit\else\renewcommand{\mathit}{\Mathit}\fi
430   \ifmst@defaultsf\else\renewcommand{\mathsf}{\Mathsf}\fi
431   \ifmst@defaultttt\else\renewcommand{\mathtt}{\Mathtt}\fi}
432 \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 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.

```

433 \def\mst@update@greeksh{
434   \def\mst@greek@lsh{\mst@ltsh} %% default behavior
435   \def\mst@greek@ush{\mst@opsh}
436   \ifmst@itgreek\def\mst@greek@lsh{\itdefault}
437   \def\mst@greek@ush{\itdefault}\fi
438   \ifmst@upgreek\def\mst@greek@lsh{\updefault}
439   \def\mst@greek@ush{\updefault}\fi
440   \ifmst@frenchmath
441     \ifmst@itgreek\else
442       \ifmst@upgreek\else
443         \def\mst@greek@lsh{\mst@opsh}
444         \def\mst@greek@ush{\mst@opsh}
445       \fi\fi
446   \fi
447   \ifcase\mst@greek@select
448     \or\def\mst@greek@ush{\itdefault}
449     \or\def\mst@greek@ush{\updefault}
450   \fi}
451 \mst@update@greeksh
452 \ifmst@LGRgreek
453   \DeclareFontEncoding{LGR}{\LGR}{}
454   \DeclareSymbolFont{mtlgrfontlower}{LGR}
455     {\mst@fam}{\mst@ser}{\mst@greek@lsh}
456   \DeclareSymbolFont{mtlgrfontupper}{LGR}

```

```

457          {\mst@fam}{\mst@ser}{\mst@greek@ush}
458 \else
459 \ifmst@selfGreek
460     \DeclareSymbolFont{mtselfGreekfont}{OT1}
461     {\mst@fam}{\mst@ser}{\mst@greek@ush}
462 \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`

```

463 \ifmst@needeuler\typeout{** will use Euler font; command \string\MTEulerScale}
464 \DeclareSymbolFont{mteulervm}{U}{zeur}{m}{n}
465 \DeclareSymbolFontAlphabet{\MathEuler}{mteulervm}
466 \DeclareMathAlphabet{\MathEulerBold}{U}{zeur}{\mst@bold}{n}
467 \fi
468 \newcommand*\MTEulerScale[1]{\edef\zeu@Scale{#1}}
469 \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`, where there are NO lowercase greek letters. This set-up is ok for the Capital Greek letters which are together with the Latin letters in both `cmmi` and `cmr`.

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

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

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

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

```

470 \ifmst@needsymbol\typeout{** will use Symbol font; command \string\MTSymbolScale}
471 \def\psy@scale{1}
472 \DeclareSymbolFont{mtpsymbol}{U}{psy}{m}{n}
473 \DeclareSymbolFontAlphabet{\MathPSymbol}{mtpsymbol}
474 \AtBeginDocument{
475     \DeclareFontFamily{U}{psy}{}
476     \DeclareFontShape{U}{psy}{m}{n}{<->s*[\psy@scale] psy}{r}{}
477 }
478 \fi
479 \newcommand*\MTSymbolScale[1]{\edef\psy@scale{#1}}
480 \let\MathastextSymbolScale\MTSymbolScale

```

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

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

```
481 \DeclareRobustCommand*\pmvec[1]{\mathord{\stackrel{\scriptstyle}{\vec}}{#1}}
482 {\tiny\boldmath$\mathord{\rightarrow}$}}{\tiny\boldmath$}
```

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

```
483 \ifmst@fouriervec
484 \DeclareFontEncoding{FML}{F}{F}
485 \DeclareFontSubstitution{FML}{F}{F}
486 \DeclareSymbolFont{mathastextfourier}{FML}{F}{F}
487 \SetSymbolFont{mathastextfourier}{bold}{FML}{F}{F}
488 \DeclareMathAccent{\fouriervec}{\mathord}{mathastextfourier}{"7E}
489 \fi
```

`\MTencoding` Some public macros to modify our private internals, and we will use them also ourself.

`\MTfamily` In version 1.1 we add the possibility to have two distinct font shapes for letters and digits.

`\MTseries` So in fact we could as well have two really unrelated fonts but this is really not the spirit of the package.

`\MTshape` Note that using these macros in the preamble allows `\Mathastext` to set up math versions

`\MTboldvariant` with a given font for math mode, and at the same time not modifying the `\familydefault` or

`\romandefault` etc...

```
490 \newcommand*\MTencoding[1]{\def\mst@enc{#1}}
491 \newcommand*\MTfamily[1]{\def\mst@fam{#1}}
492 \newcommand*\MTseries[1]{\def\mst@ser{#1}}
493 \newcommand*\MTshape[1]{\def\mst@opsh{#1}}
494 \ifmst@italic\else\def\mst@ltsh{#1}\fi
495 \newcommand*\MTboldvariant[1]{\def\mst@bold{#1}}
496 \newcommand*\MTlettershape[1]{\def\mst@ltsh{#1}}
497 \@onlypreamble\MTencoding
498 \@onlypreamble\MTfamily
499 \@onlypreamble\MTseries
500 \@onlypreamble\MTshape
501 \@onlypreamble\MTboldvariant
502 \@onlypreamble\MTlettershape
503 \let\Mathastextencoding\MTencoding
504 \let\Mathastextfamily\MTfamily
505 \let\Mathastextseries\MTseries
506 \let\Mathastextshape\MTshape
507 \let\Mathastextboldvariant\MTboldvariant
508 \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

`\MTupGreek` or selfGreeks.

`\MTgreekfont`

```

509 \newcommand*\MTitgreek{%
510 \mst@itgreektrue\mst@upgreekfalse\def\mst@greek@select{0}}
511 \newcommand*\MTupgreek{%
512 \mst@upgreektrue\mst@itgreekfalse\def\mst@greek@select{0}}
513 \newcommand*\MTitGreek{\def\mst@greek@select{1}}
514 \newcommand*\MTupGreek{\def\mst@greek@select{2}}
515 \@onlypreamble\MTitgreek
516 \@onlypreamble\MTupgreek
517 \@onlypreamble\MTitGreek
518 \@onlypreamble\MTitGreek
519 \let\Mathastextitgreek\MTitgreek
520 \let\Mathastextupgreek\MTupgreek
521 \let\MathastextitGreek\MTitGreek
522 \let\MathastextupGreek\MTupGreek
523 \newcommand*\MTgreekfont[1]{\def\mst@greekfont{#1}}
524 \@onlypreamble\MTgreekfont
525 \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**.

```

526 \newcommand*\MTWillUse[5][\empty]{
527 \MTencoding{#2}
528 \MTfamily{#3}
529 \MTseries{#4}
530 \MTshape{#5}
531 \ifmst@italic\MTlettershape{\itdefault}\fi % was missing in v 1.14 and prior
532 \ifx\empty#1
533 \else
534 %% we check if optional argument was [], this does not detect [ ]
535 \def\mst@tmp{#1}
536 \ifx\empty\mst@tmp\else\MTlettershape{#1}\fi
537 \fi
538 \Mathastext}
539 \@onlypreamble\MTWillUse
540 \let\MathastextWillUse\MTWillUse
541 \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.

```

542 \newcommand*\Mathastext[1][\empty]{
543 \ifx\empty#1 % no optional argument
544 \mst@update@greeksh
545 \edef\mst@encoding@normal{\mst@enc}

```

```

546 \edef\mst@family@normal{\mst@fam}
547 \edef\mst@series@normal{\mst@ser}
548 \edef\mst@shape@normal{\mst@opsh}
549 \edef\mst@ltshape@normal{\mst@ltsh}
550 \edef\mst@itdefault@normal{\itdefault}
551 \edef\mst@rmdefault@normal{\rmdefault}
552 \edef\mst@sfdefault@normal{\sfdefault}
553 \edef\mst@ttdefault@normal{\ttdefault}
554 \edef\mst@boldvariant@normal{\mst@bold}
555 \edef\mst@exists@skip@normal{\mst@exists@skip}
556 \edef\mst@forall@skip@normal{\mst@forall@skip}
557 \edef\mst@prime@skip@normal{\mst@prime@skip}
558 \edef\mst@encoding@bold{\mst@enc}
559 \edef\mst@family@bold{\mst@fam}
560 \edef\mst@series@bold{\mst@bold}
561 \edef\mst@shape@bold{\mst@opsh}
562 \edef\mst@ltshape@bold{\mst@ltsh}
563 \edef\mst@boldvariant@bold{\mst@bold}
564 \edef\mst@itdefault@bold{\itdefault}
565 \edef\mst@rmdefault@bold{\rmdefault}
566 \edef\mst@sfdefault@bold{\sfdefault}
567 \edef\mst@ttdefault@bold{\ttdefault}
568 \edef\mst@exists@skip@bold{\mst@exists@skip}
569 \edef\mst@forall@skip@bold{\mst@forall@skip}
570 \edef\mst@prime@skip@bold{\mst@prime@skip}
571 \MTicinmath % does \MTmathactiveletters, \MTnoicinmath only cancels ics
572 \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@ltshape@normal\MTnoicinmath\fi}%
573 \ifmst@frenchmath
574     \MTICinmath
575     \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@shape@normal\MTnoICinmath\fi}%
576 \fi
577 \ifmst@subdued
578     \MTeasyonlettersdonotobeymathxx
579     \MTnonlettersdonotobeymathxx
580     \MTmathoperatorsdonotobeymathxx
581     \MTmathstandardletters % prior to 1.3, was \MTnoicinmath
582     \ifmst@asterisk\let\mst@doasterisk\relax\fi
583     \def\mst@exists@skip@normal{0mu}
584     \def\mst@forall@skip@normal{0mu}
585     \def\mst@prime@skip@normal{0mu}
586     \def\mst@exists@skip@bold{0mu}
587     \def\mst@forall@skip@bold{0mu}
588     \def\mst@prime@skip@bold{0mu}
589 \else % not subdued
590     \ifmst@italic
591         \ifmst@frenchmath
592             \mst@exists@muskip\mst@exists@skip\relax
593             \mst@forall@muskip\mst@forall@skip\relax
594             \mst@prime@muskip\mst@prime@skip\relax

```

```

595     \else
596     \def\mst@exists@skip@normal{0mu}
597     \def\mst@forall@skip@normal{0mu}
598     \def\mst@prime@skip@normal{0mu}
599     \def\mst@exists@skip@bold{0mu}
600     \def\mst@forall@skip@bold{0mu}
601     \def\mst@prime@skip@bold{0mu}
602     \fi
603   \else
604     \mst@exists@muskip\mst@exists@skip\relax
605     \mst@forall@muskip\mst@forall@skip\relax
606     \mst@prime@muskip\mst@prime@skip\relax
607   \fi
608 \fi
609 %% v1.15f
610 \ifmst@nonormalbold\else
611   \SetMathAlphabet{\mathnormalbold}{normal}{\mst@encoding@normal}
612                                     {\mst@family@normal}
613                                     {\mst@boldvariant@normal}
614                                     {\mst@ltshape@normal}
615   \SetMathAlphabet{\mathnormalbold}{bold}{\mst@encoding@bold}
616                                     {\mst@family@bold}
617                                     {\mst@boldvariant@bold}
618                                     {\mst@ltshape@bold}
619 \fi
620 %% v1.15f adds \ifmst@default.. checks
621 \ifmst@subdued\else
622   \SetSymbolFont{mtletterfont}{normal}{\mst@encoding@normal}
623                                     {\mst@family@normal}
624                                     {\mst@series@normal}
625                                     {\mst@ltshape@normal}
626   \SetSymbolFont{mtletterfont}{bold}{\mst@encoding@bold}
627                                     {\mst@family@bold}
628                                     {\mst@series@bold}
629                                     {\mst@ltshape@bold}
630   \SetSymbolFont{mtooperatorfont}{normal}{\mst@encoding@normal}
631                                     {\mst@family@normal}
632                                     {\mst@series@normal}
633                                     {\mst@shape@normal}
634   \SetSymbolFont{mtooperatorfont}{bold}{\mst@encoding@bold}
635                                     {\mst@family@bold}
636                                     {\mst@series@bold}
637                                     {\mst@shape@bold}
638 \ifmst@defaultbf\else
639   \SetMathAlphabet{\Mathbf}{normal}{\mst@encoding@normal}
640                                     {\mst@family@normal}
641                                     {\mst@series@bold}
642                                     {\mst@shape@normal}
643   \SetMathAlphabet{\Mathbf}{bold}{\mst@encoding@bold}

```

```

644             {\mst@family@bold}
645             {\mst@series@bold}
646             {\mst@shape@bold}
647 \fi\ifmst@defaultit\else
648   \SetMathAlphabet{\Mathit}{normal}{\mst@encoding@normal}
649             {\mst@family@normal}
650             {\mst@series@normal}
651             {\mst@itdefault@normal}
652   \SetMathAlphabet{\Mathit}{bold}{\mst@encoding@bold}
653             {\mst@family@bold}
654             {\mst@series@bold}
655             {\mst@itdefault@bold}
656 \fi\ifmst@defaultsf\else
657   \SetMathAlphabet{\Mathsf}{normal}{\mst@encoding@normal}
658             {\mst@sfdefault@normal}
659             {\mst@series@normal}
660             {\mst@shape@normal}
661   \SetMathAlphabet{\Mathsf}{bold}{\mst@encoding@bold}
662             {\mst@sfdefault@bold}
663             {\mst@series@bold}
664             {\mst@shape@bold}
665 \fi\ifmst@defaulttt\else
666   \SetMathAlphabet{\Mathtt}{normal}{\mst@encoding@normal}
667             {\mst@ttdefault@normal}
668             {\mst@series@normal}
669             {\mst@shape@normal}
670   \SetMathAlphabet{\Mathtt}{bold}{\mst@encoding@bold}
671             {\mst@ttdefault@bold}
672             {\mst@series@bold}
673             {\mst@shape@bold}
674 \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.

```

675 \ifmst@needeuler
676   \SetSymbolFont{mteulervm}{bold}{U}{zeur}{\mst@boldvariant@normal}{n}
677   \SetMathAlphabet{\MathEulerBold}{normal}
678             {U}{zeur}{\mst@boldvariant@normal}{n}
679   \SetMathAlphabet{\MathEulerBold}{bold}
680             {U}{zeur}{\mst@boldvariant@normal}{n}
681 \fi

682 \ifmst@needsymbol\SetSymbolFont{mtpsymbol}{bold}
683             {U}{psy}{\mst@boldvariant@normal}{n}
684 \fi

```

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

selfGreek* 685 \ifmst@subdued\else

```

686 \ifmst@LGRgreek
687   \SetSymbolFont{mtlgrfontlower}{normal}{LGR}
688     {\mst@greekfont}{\mst@series@normal}{\mst@greek@lsh}
689   \SetSymbolFont{mtlgrfontlower}{bold}{LGR}
690     {\mst@greekfont}{\mst@boldvariant@normal}{\mst@greek@lsh}
691   \SetSymbolFont{mtlgrfontupper}{normal}{LGR}
692     {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
693   \SetSymbolFont{mtlgrfontupper}{bold}{LGR}
694     {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
695 \else
696 \ifmst@selfGreek
697   \SetSymbolFont{mtselfGreekfont}{normal}{OT1}
698     {\mst@greekfont}{\mst@series@normal}{\mst@greek@ush}
699   \SetSymbolFont{mtselfGreekfont}{bold}{OT1}
700     {\mst@greekfont}{\mst@boldvariant@bold}{\mst@greek@ush}
701 \fi\fi\fi

702 \ifmst@subdued
703 \typeout{** subdued mode will be activated for the normal and bold math versions}
704 \else
705 \typeout{** Latin letters in the normal (resp. bold) math versions are now^^J%
706   ** set up to use the fonts
707 \mst@encoding@normal/\mst@family@normal/\mst@series@normal%
708 (\mst@boldvariant@normal)/\mst@ltshape@normal}
709 \ifmst@LGRgreek\typeout{** Greek letters (\mst@greek@lsh/\mst@greek@ush)
710 will use LGR/\mst@greekfont}
711 \fi
712 \ifmst@nodigits\else\typeout{** Other characters (digits, ...)
713 and \protect\log-like names will be^^J%
714 ** typeset with the \mst@shape@normal\space shape.}
715 \fi
716 \fi
717 \else % optional argument, then we rather do DeclareVersion
718 \edef\mst@tmp{#1}
719 \ifx\mst@tmp\empty
720   \Mathastext % no infinite loop!
721 \else
722   \MTDeclareVersion[\mst@ltsh]{#1}
723     {\mst@enc}{\mst@fam}{\mst@ser}{\mst@opsh}
724 \fi
725 \fi
726 }
727 \@onlypreamble\Mathastext
728 \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!

```

729 \newcommand*{MTDeclareVersion}[6][\empty]{
730   \edef\mst@version{#2}
731   \DeclareMathVersion{\mst@version}
732   \expandafter\edef\csname mst@encoding@\mst@version\endcsname{#3}
733   \expandafter\edef\csname mst@family@\mst@version\endcsname{#4}
734   \expandafter\edef\csname mst@series@\mst@version\endcsname{#5}
735   \expandafter\edef\csname mst@shape@\mst@version\endcsname{#6}
736   \expandafter\edef\csname mst@boldvariant@\mst@version\endcsname{\mst@bold}
737   \expandafter\edef\csname mst@itdefault@\mst@version\endcsname{\itdefault}
738   \expandafter\edef\csname mst@rmdefault@\mst@version\endcsname{\rmdefault}
739   \expandafter\edef\csname mst@sfdefault@\mst@version\endcsname{\sfdefault}
740   \expandafter\edef\csname mst@ttdefault@\mst@version\endcsname{\ttdefault}
741   \expandafter\edef\csname mst@exists@skip@\mst@version\endcsname
742     {\mst@exists@skip}
743   \expandafter\edef\csname mst@forall@skip@\mst@version\endcsname
744     {\mst@forall@skip}
745   \expandafter\edef\csname mst@prime@skip@\mst@version\endcsname
746     {\mst@prime@skip}
747   \mst@optionalparamfalse
748   \ifx\empty#1
749     \else % we check nevertheless for [] situation, will not work with [ ]
750       \def\mst@tmp{#1}
751       \ifx\mst@tmp\empty\else\mst@optionalparamtrue\fi
752     \fi
753   \ifmst@optionalparam
754     \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#1}
755     \typeout{** Latin letters in math version ‘#2\string’ will use the font
756       #3/#4/#5/#1~J%
757     ** Other characters (digits, ...) and \protect\log-like
758       names will be in #6 shape.}
759     \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#1}
760   \else
761     \ifmst@italic
762       \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{\mst@ltsh}
763       \typeout{** Latin letters in math version ‘#2\string’ will use the font
764         #3/#4/#5/\mst@ltsh~J%
765       ** Other characters (digits, ...) and \protect\log-like names
766         will be in #6 shape.}
767       \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{\mst@ltsh}
768     \else
769       \SetSymbolFont{mtletterfont}{#2}{#3}{#4}{#5}{#6}
770       \typeout{** Latin letters in math version ‘#2\string’ will use the fonts
771         #3/#4/#5(\mst@bold)/#6}
772       \expandafter\edef\csname mst@ltshape@\mst@version\endcsname{#6}

```

```

773 \fi
774 \fi
775 \ifmst@nonormalbold\else
776 \SetMathAlphabet{\mathnormalbold}{#2}{#3}{#4}{\mst@bold}
777 \csname
778 mst@ltshape@\mst@version\endcsname}
779 \fi
780 \SetSymbolFont{mtoperatorfont}{#2}{#3}{#4}{#5}{#6}
781 \ifmst@defaultbf\else\SetMathAlphabet{\Mathbf}{#2}{#3}{#4}{\mst@bold}{#6}\fi
782 \ifmst@defaultit\else\SetMathAlphabet{\Mathit}{#2}{#3}{#4}{#5}{\itdefault}\fi
783 \ifmst@defaultsf\else\SetMathAlphabet{\Mathsf}{#2}{#3}{\sfdefault}{#5}{#6}\fi
784 \ifmst@defaulttt\else\SetMathAlphabet{\Mathtt}{#2}{#3}{\ttdefault}{#5}{#6}\fi
785 \ifmst@needeuler
786 \SetMathAlphabet{\MathEulerBold}{#2}{U}{zeur}{\mst@bold}{n}
787 \fi

```

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

```

788 \def\mst@greek@lsh@loc{\csname mst@ltshape@\mst@version\endcsname}
789 \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
790 \ifmst@itgreek\def\mst@greek@lsh@loc{\itdefault}
791 \def\mst@greek@ush@loc{\itdefault}\fi
792 \ifmst@upgreek\def\mst@greek@lsh@loc{\updefault}
793 \def\mst@greek@ush@loc{\updefault}\fi
794 \ifmst@frenchmath
795 \ifmst@itgreek\else
796 \ifmst@upgreek\else
797 \def\mst@greek@lsh@loc{\csname mst@shape@\mst@version\endcsname}
798 \def\mst@greek@ush@loc{\csname mst@shape@\mst@version\endcsname}
799 \fi\fi
800 \fi
801 \ifcase\mst@greek@select
802 \or\def\mst@greek@ush@loc{\itdefault}
803 \or\def\mst@greek@ush@loc{\updefault}
804 \fi
805 \ifmst@LGRgreek
806 \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{#4}{#5}{\mst@greek@lsh@loc}
807 \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{#4}{#5}{\mst@greek@ush@loc}
808 \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
809 LGR font #4}
810 \else
811 \ifmst@selfGreek
812 \SetSymbolFont{mtselfGreekfont}{#2}{OT1}{#4}{#5}{\mst@greek@ush@loc}
813 \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
814 will use OT1 font #4}
815 \else
816 \ifmst@LGRgreek

```

```

817     \SetSymbolFont{mtlgrfontlower}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@lsh@loc}
818     \SetSymbolFont{mtlgrfontupper}{#2}{LGR}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
819     \typeout{** Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc) will use
820         LGR font \mst@greekfont}
821 \else
822 \ifmst@selfGreek
823     \SetSymbolFont{mtselfGreekfont}{#2}{OT1}{\mst@greekfont}{#5}{\mst@greek@ush@loc}
824     \typeout{** Capital Greek letters (\mst@greek@lsh@loc/\mst@greek@ush@loc)
825         will use OT1 font \mst@greekfont}
826 \fi\fi\fi\fi
827 }
828 \@onlypreamble\MTDeclareVersion

829 \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...

1.3: activation of italic corrections is now separated from actual math activation of letters.

```

830 \newcommand*{\MTVersion}[2][\empty]{%
831     \mathversion{#2}%
832     \ifx\empty#1% no optional argument
833         \def\mst@tmp{#2}%
834     \else% we check nevertheless for [] situation, will not work with [ ]
835         \def\mst@tmpa{#1}%
836         \ifx\mst@tmpa\empty\def\mst@tmp{#2}%
837         \else\def\mst@tmp{#1}%
838         \fi%
839     \fi%
840     \usefont{\csname mst@encoding@\mst@tmp\endcsname}%
841         {\csname mst@family@\mst@tmp\endcsname}%
842         {\csname mst@series@\mst@tmp\endcsname}%
843         {\csname mst@shape@\mst@tmp\endcsname}%
844     \edef\mst@@encoding{\csname mst@encoding@\mst@tmp\endcsname}%
845     \renewcommand{\encodingdefault}{\mst@@encoding}%
846 %% some modifications for v1.15b
847     \edef\mst@@famdefault{\csname mst@family@\mst@tmp\endcsname}%
848     \renewcommand{\familydefault}{\mst@@famdefault}%
849     \edef\mst@@series{\csname mst@series@\mst@tmp\endcsname}%
850     \renewcommand{\mddefault}{\mst@@series}%

```

```

851 \edef\mst@@shape{\csname mst@shape@\mst@tmp\endcsname}%
852 \renewcommand{\shapedefault}{\mst@@shape}%
853 \edef\mst@@boldvariant{\csname mst@boldvariant@\mst@tmp\endcsname}%
854 \renewcommand{\bfdefault}{\mst@@boldvariant}%
855 \edef\mst@@itdefault{\csname mst@itdefault@\mst@tmp\endcsname}%
856 \renewcommand{\itdefault}{\mst@@itdefault}%
857 \edef\mst@@rmdefault{\csname mst@rmdefault@\mst@tmp\endcsname}%
858 \renewcommand{\rmdefault}{\mst@@rmdefault}%
859 \edef\mst@@sfdefault{\csname mst@sfdefault@\mst@tmp\endcsname}%
860 \renewcommand{\sfdefault}{\mst@@sfdefault}%
861 \edef\mst@@ttdefault{\csname mst@ttdefault@\mst@tmp\endcsname}%
862 \renewcommand{\ttdefault}{\mst@@ttdefault}%
863 \edef\mst@tmp{#2}%
864 %% v1.15e: muskips
865 \expandafter
866 \mst@exists@muskip\csname mst@exists@skip@\mst@tmp\endcsname\relax
867 \expandafter
868 \mst@forall@muskip\csname mst@forall@skip@\mst@tmp\endcsname\relax
869 %% v1.2: muskip for \cs{prime}
870 \expandafter
871 \mst@prime@muskip\csname mst@prime@skip@\mst@tmp\endcsname\relax
872 %% v1.2b: italic corrections except for italic/slanted (sic) letters, and of
873 %% course except in the subdued normal and bold math versions
874 %% v1.3: by default, letters are made mathematically active, even if italic
875 %% corrections are not used, to allow the action of \MTsetmathskips.
876 \edef\mst@tmpa{\csname mst@ltshape@\mst@tmp\endcsname}%
877 \edef\mst@tmpb{\csname mst@shape@\mst@tmp\endcsname}%
878 \MTicinmath
879 \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@tmpa\MTnoicinmath\fi}%
880 \ifmst@frenchmath
881 \MTICinmath
882 \@for\mst@tmp:=it,sl\do{\ifx\mst@tmp\mst@tmpb\MTnoICinmath\fi}%
883 \fi
884 %% v1.15c: extending subdued to LGRgreek
885 %% v1.15f: subdueing math alphabets in a simpler way than in 1.15e
886 %% v1.2b: subdueing the activation of characters in math mode
887 %% v1.2d: special treatment of the asterisk
888 \MTmathoperatorsobeymathxx
889 \MTeasyonlettersobeymathxx
890 \ifmst@asterisk\let\mst@doasterisk\mst@@doasterisk\fi
891 \ifmst@subdued
892 \edef\mst@tmp{#2}%
893 \def\mst@tmpa{normal}%
894 \ifx\mst@tmp\mst@tmpa
895 \mst@restorealphabets
896 \MTmathoperatorsdonotobeymathxx
897 \MTnonlettersdonotobeymathxx
898 \MTeasyonlettersdonotobeymathxx
899 \MTmathstandardletters

```

```

900     \ifmst@asterisk\let\mst@doasterisk\relax\fi
901     \ifmst@LGRgreek\mst@restoregreek\fi
902     \else
903     \def\mst@tmpa{bold}%
904     \ifx\mst@tmp\mst@tmpa
905         \mst@restorealphabets
906         \MTmathoperatorsdonotobeymathxx
907         \MTnonlettersdonotobeymathxx
908         \MTeasynonlettersdonotobeymathxx
909         \MTmathstandardletters
910     \ifmst@asterisk\let\mst@doasterisk\relax\fi
911     \ifmst@LGRgreek\mst@restoregreek\fi
912     \else
913     \mst@setalphabets
914     \ifmst@LGRgreek\mst@setuplrgreek\fi
915     \fi\fi\fi
916 }
917 \let\MathastextVersion\MTVersion
918 \let\Mathastextversion\MTVersion
919 \let\MTversion\MTVersion
920 \let\mathastextversion\MTVersion

```

`\Mathastext` Initialization call:

```
921 \Mathastext
```

Additional appropriate messages to the terminal and the log.

```

922 \ifmst@eulergreek
923     \typeout{** Greek letters will use the Euler font. Use^^J%
924 ** \protect\MathastextEulerScale{<factor>} to scale the font.}
925 \else
926 \ifmst@symbolgreek
927     \typeout{** Greek letters will use the PostScript Symbol font. Use^^J%
928 ** \protect\MathastextSymbolScale{<factor>} to scale the font.}
929 \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 initiated the use of mathematically active letters to insert the italic corrections. With version 1.3 the use of math active letters is also for extra muglue added before and after the letters. Use of `\@for` to shorten the code initiated with release 1.3.

```

930 \def\mst@DeclareMathLetter #1{%
931     \DeclareMathSymbol {#1}{\mathalpha}{mtletterfont}{‘#1}%
932     \expandafter
933     \DeclareMathSymbol \csname mst@#1\endcsname{\mathalpha}{mtletterfont}{‘#1}%
934     \expandafter\mst@addtodo@az
935     \expandafter #1\csname mst@#1\endcsname\mst@itcorr}
936 \@for\mst@tmp:=a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\do
937     {\expandafter\mst@DeclareMathLetter\mst@tmp}
938 \ifmst@frenchmath \def\mst@font@tbu{mtoperatorfont}
939 \else \def\mst@font@tbu{mtletterfont}

```

```

940 \fi
941 \def\mst@DeclareMathLetter #1{%
942   \DeclareMathSymbol {#1}{\mathalpha}{\mst@font@tbu}{‘#1}%
943   \expandafter
944   \DeclareMathSymbol \csname mst@#1\endcsname{\mathalpha}{\mst@font@tbu}{‘#1}%
945   \expandafter\mst@addtodo@AZ
946   \expandafter #1\csname mst@#1\endcsname\mst@ITcorr}
947 \@for\mst@tmp:=A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z\do
948   {\expandafter\mst@DeclareMathLetter\mst@tmp}
949 \let\mst@DeclareMathLetter\relax
950 %%
951 \ifmst@nodigits\else
952 \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.

```

953 \ifmst@symboldigits \def\mst@font@tbu{mtpsymbol} \fi
954 \ifmst@eulerdigits \def\mst@font@tbu{mteulervm} \fi
955 \DeclareMathSymbol{0}{\mathalpha}{\mst@font@tbu}{‘0}
956 \DeclareMathSymbol{1}{\mathalpha}{\mst@font@tbu}{‘1}
957 \DeclareMathSymbol{2}{\mathalpha}{\mst@font@tbu}{‘2}
958 \DeclareMathSymbol{3}{\mathalpha}{\mst@font@tbu}{‘3}
959 \DeclareMathSymbol{4}{\mathalpha}{\mst@font@tbu}{‘4}
960 \DeclareMathSymbol{5}{\mathalpha}{\mst@font@tbu}{‘5}
961 \DeclareMathSymbol{6}{\mathalpha}{\mst@font@tbu}{‘6}
962 \DeclareMathSymbol{7}{\mathalpha}{\mst@font@tbu}{‘7}
963 \DeclareMathSymbol{8}{\mathalpha}{\mst@font@tbu}{‘8}
964 \DeclareMathSymbol{9}{\mathalpha}{\mst@font@tbu}{‘9}
965 \fi

```

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

```

966 \ifmst@symboldelimiters
967 \def\mst@font@tbu{mtpsymbol}
968 \mst@endashfalse
969 \mst@emdashfalse
970 \else
971 \def\mst@font@tbu{mtoperatorfont}
972 \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`.

```

973 \ifmst@noexclam\else\typeout{** \string! and \string?}
974 \DeclareMathSymbol{!}{\mathclose}{\mst@font@tbu}{"21}
975 \DeclareMathSymbol{\mst@varfam@exclam}{\mathalpha}{\mst@font@tbu}{"21}
976 \expandafter\mst@addtodo@nonletters\string!\mathclose\mst@varfam@exclam
977 \DeclareMathSymbol{?}{\mathclose}{\mst@font@tbu}{"3F}
978 \DeclareMathSymbol{\mst@varfam@question}{\mathalpha}{\mst@font@tbu}{"3F}
979 \expandafter\mst@addtodo@nonletters\string?\mathclose\mst@varfam@question
980 \fi

```

`\MTlowerast` 1.12d The `\ast` or `*` is defined in `fontmath.ltx` as a binary operator from the `symbols` font.
`\mst@doasterisk` Usually the asterisk from the text font is in a raised position. Previous versions of `mathastext` did
`\mst@@doasterisk` nothing with `\ast` but strangely defined `*` to be the one from the text font, with type `\mathalpha`.
The package now leaves by default both `*` and `\ast` untouched, and if passed option `asterisk` replaces both of them with a lowered text asterisk (or the one from the Symbol font), and of type `\mathbin`. A trick is used to optionally get both `*` and `\ast` obey the math alphabets.
The user macro `\MTlowerast` sets the amount of lowering to be applied to the text asterisk.
1.12e Somehow there was a big omission in 1.12d, the command `\MTlowerast` as described in the manual was missing!
nota bene: it is assumed that `*` is of type `other` when `mathastext` is loaded... it should neither be active, nor of type `letter`!

```

981 \def\mst@@doasterisk{\let\ast\mst@ast\mst@mathactivate*{}\mst@ast}
982 \ifmst@asterisk\typeout{** asterisk: \string\ast\space and *}
983 \ifmst@symbolmisc
984   \def\mst@bin@ast{%
985     \mathbin{\mathchoice{\raisebox{-\mst@lowerast}{\the\textfont\symmtpsymbol\char42}}
986                       {\raisebox{-\mst@lowerast}{\the\textfont\symmtpsymbol\char42}}
987                       {\raisebox{-\mst@lowerast}{\the\textfont\symmtpsymbol\char42}}
988                       {\raisebox{-\mst@lowerast}{\the\scriptfont\symmtpsymbol\char42}}
989                       {\raisebox{-\mst@lowerast}{\the\scriptscriptfont\symmtpsymbol\char42}}}}
990   \else
991     \def\mst@bin@ast{%
992       \mathbin{\mathchoice{\raisebox{-\mst@lowerast}{\the\textfont\symmtooperatorfont\char42}}
993                         {\raisebox{-\mst@lowerast}{\the\textfont\symmtooperatorfont\char42}}
994                         {\raisebox{-\mst@lowerast}{\the\textfont\symmtooperatorfont\char42}}
995                         {\raisebox{-\mst@lowerast}{\the\scriptfont\symmtooperatorfont\char42}}
996                         {\raisebox{-\mst@lowerast}{\the\scriptscriptfont\symmtooperatorfont\char42}}}}
997   \fi
998 \def\mst@varfam@ast{\ifnum\fam=\m@ne\mst@bin@ast\else
999   \mathbin{\mathchoice{\raisebox{-\mst@lowerast}{\the\textfont\fam\char42}}
1000                    {\raisebox{-\mst@lowerast}{\the\textfont\fam\char42}}
1001                    {\raisebox{-\mst@lowerast}{\the\textfont\fam\char42}}
1002                    {\raisebox{-\mst@lowerast}{\the\scriptfont\fam\char42}}
1003                    {\raisebox{-\mst@lowerast}{\the\scriptscriptfont\fam\char42}}}\fi}
1004 \let\mst@doasterisk\mst@@doasterisk
1005 \DeclareRobustCommand*\mst@ast{\mst@bin@ast}
1006 \newcommand*\MTlowerast[1]{\def\mst@lowerast{#1}}
1007 \MTlowerast{.3\height}
1008 \mst@do@easynonletters\expandafter{\the\mst@do@easynonletters
1009   \expandafter\def\csname mst@ast \endcsname{\mst@varfam@ast}}

```

1019 \fi

(2011) I renounced to try to do things with all the various dots, they are defined in many different ways, and there is the amsmath also. Dealing with this issue would mean a lot a time for a minuscule result. Better to leave the user use the `mathdots` package and accept that we can not avoid the default fonts in that case. So here I just treat `.` (in the hope to really lessen by 1 the number of fonts embedded at the end in the PDF).

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

```

1020 \ifmst@nopunct\else\typeout{** punctuation\string: \string. \string. \string: \string; and
1021 \DeclareMathSymbol{,}{\mathpunct}{\mst@font@tbu}{"2C}
1022 \DeclareMathSymbol{\mst@varfam@comma}{\mathalpha}{\mst@font@tbu}{"2C}
1023 \expandafter\mst@addtodo@nonletters\string,\mathpunct\mst@varfam@comma
1024 \DeclareMathSymbol{.}{\mathord}{\mst@font@tbu}{"2E}
1025 \DeclareMathSymbol{\mst@varfam@dot}{\mathalpha}{\mst@font@tbu}{"2E}
1026 \mst@addtodo@easynonletters\.\mst@varfam@dot
1027 \DeclareMathSymbol{:}{\mathrel}{\mst@font@tbu}{"3A} % relation spacing
1028 \DeclareMathSymbol{\mst@varfam@colon}{\mathalpha}{\mst@font@tbu}{"3A}
1029 \expandafter\mst@addtodo@nonletters\string:\mathrel\mst@varfam@colon
1030 \@ifpackageloaded{amsmath}
1031 {} % \colon defined in amsmath.sty in terms of : with some enlarged explicit
1032 % spacing. No need to intervene.
1033 {% no amsmath, use standard punctuation spacing
1034 \let\colon\undefined
1035 % the reason is if some package has redefined \colon which then
1036 % can not be used in \cs{DeclareMathSymbol} anymore (we
1037 % shamelessly overwrite...)
1038 \DeclareMathSymbol{\colon}{\mathpunct}{\mst@font@tbu}{"3A}
1039 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1040 \def\colon{\mathpunct{\mst@varfam@colon}}}}
1041 \DeclareMathSymbol{;}{\mathpunct}{\mst@font@tbu}{"3B}
1042 \DeclareMathSymbol{\mst@varfam@pointvirgule}{\mathalpha}{\mst@font@tbu}{"3B}
1043 \expandafter\mst@addtodo@nonletters\string;\mathpunct\mst@varfam@pointvirgule
1044 \fi

```

\relbar Due to the way `=` and `-` are used by L^AT_EX in arrows, we will have to redefine `\Relbar` and `\relbar` in order for them to preserve their original meanings.

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

```

1045 \ifmst@nominus\else
1046 \ifmst@XeTeX
1047 \XeTeXmathcharnumdef\mst@minus@sign=\XeTeXmathcodenum'\-\relax
1048 \else
1049 \ifmst@LuaTeX
1050 \luatexUmathcharnumdef\mst@minus@sign=\luatexUmathcodenum'\-\relax
1051 %% I used this prior to the new \luatexUmathcodenum, as available with TL2013:
1052 %% \mathchardef\mst@minus@sign=8704\relax % "2200
1053 \else

```

```

1054 \mathchardef\mst@minus@sign=\mathcode'\- \relax
1055 \fi\fi
1056 \@ifpackageloaded{amsmath}
1057 {\def\relbar{\mathrel{\mathpalette\mathsm@sh\mst@minus@sign}}}
1058 {\DeclareRobustCommand\relbar{\mathrel{\smash\mst@minus@sign}}}
1059 \def\rightarrowfill{\mathord{\relbar}\mkern-7mu%
1060   \cleaders\hbox{$\mkern-2mu\relbar\mkern-2mu$}\hfill
1061   \mkern-7mu\mathord\rightarrow$}
1062 \def\leftarrowfill{\mathord{\leftarrow\mkern-7mu%
1063   \cleaders\hbox{$\mkern-2mu\relbar\mkern-2mu$}\hfill
1064   \mkern-7mu\mathord{\relbar}$}
1065 \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.

```

1066 \def\do@the@endashstuff{\edef\mst@tmp@enc{\encodingdefault}
1067   \ifx\mst@tmp@enc\mst@eui % xetex+fontspec
1068   \AtBeginDocument{%
1069     \XeTeXmathcode'\-="2 \symmoperatorfont "2013\relax
1070     \XeTeXmathchardef\mst@varfam@minus="7 \symmoperatorfont "2013\relax
1071   }
1072 \else
1073   \ifx\mst@tmp@enc\mst@euii % luatex+fontspec
1074   \AtBeginDocument{%
1075     \luatexUmathcode'\-="2 \symmoperatorfont "2013\relax
1076     \luatexUmathchardef\mst@varfam@minus="7 \symmoperatorfont "2013\relax
1077   }
1078 \else
1079   \ifx\mst@tmp@enc\mst@ti % T1
1080   \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{21}
1081   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{21}
1082 \else
1083   \ifx\mst@tmp@enc\mst@oti % OT1
1084   \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{123}
1085   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{123}
1086 \else
1087   \ifx\mst@tmp@enc\mst@lyi % LY1
1088   \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{150}
1089   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{150}
1090 \else % make sure we have neither xetex nor luatex
1091   \ifmst@XeTeX\mst@aborttrue\else\ifmst@LuaTeX\mst@aborttrue\else
1092   \DeclareMathSymbol{-}{\mathbin}{moperatorfont}
1093   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}

```

```

1094 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\textendash\endcsname}
1095   \fi\fi
1096 \fi\fi\fi\fi\fi}
1097 \def\do@the@emdashstuff{\edef\mst@tmp@enc{\encodingdefault}
1098 \ifx\mst@tmp@enc\mst@eui % xetex+fontspec
1099   \AtBeginDocument{%
1100     \XeTeXmathcode'\-="2 \symmoperatorfont "2014\relax
1101     \XeTeXmathchardef\mst@varfam@minus="7 \symmoperatorfont "2014\relax
1102   }
1103 \else
1104 \ifx\mst@tmp@enc\mst@euii % luatex+fontspec
1105   \AtBeginDocument{%
1106     \luatexUmathcode'\-="2 \symmoperatorfont "2014\relax
1107     \luatexUmathchardef\mst@varfam@minus="7 \symmoperatorfont "2014\relax
1108   }
1109 \else
1110   \ifx\mst@tmp@enc\mst@ti % T1
1111     \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{22}
1112     \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{22}
1113 \else
1114   \ifx\mst@tmp@enc\mst@oti % OT1
1115     \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{124}
1116     \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{124}
1117 \else
1118   \ifx\mst@tmp@enc\mst@lyi % LY1
1119     \DeclareMathSymbol{-}{\mathbin}{moperatorfont}{151}
1120     \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}{151}
1121 \else % make sure we have neither xetex nor luatex
1122   \ifmst@XeTeX\mst@aborttrue\else\ifmst@LuaTeX\mst@aborttrue\else
1123     \DeclareMathSymbol{-}{\mathbin}{moperatorfont}
1124     \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{moperatorfont}
1125 {\expandafter\the\expandafter\csname\mst@tmp@enc\string\textendash\endcsname}
1126   \fi\fi
1127 \fi\fi\fi\fi\fi}
1128 \ifmst@nominus\else\typeout{** minus as endash}
1129 \mst@abortfalse
1130 \ifmst@endash\do@the@emdashstuff\else
1131 \ifmst@emdash\do@the@emdashstuff\else
1132   \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{"2D}
1133   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{\mst@font@tbu}{"2D}
1134 \fi\fi
1135 \ifmst@abort
1136   \DeclareMathSymbol{-}{\mathbin}{\mst@font@tbu}{"2D}
1137   \DeclareMathSymbol{\mst@varfam@minus}{\mathalpha}{\mst@font@tbu}{"2D}
1138 \fi
1139 \expandafter\mst@addtodo@nonletters\string-\mathbin\mst@varfam@minus
1140 \fi

```

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

```
1141 %%\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` ($18\mu=1\text{em}$ 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...)

```
1142 %
1143 \def\mst@mathrlap{\mathpalette\mst@mathrlapinternal}
1144 \def\mst@mathrlapinternal#1#2{\rlap{$\mathsurround=0pt#1{#2}$}}
1145 \def\do@the@hbarstuff{\edef\mst@tmp@enc{\encodingdefault}
1146 \ifx\mst@tmp@enc\mst@eui % xetex and unicode font
1147   \XeTeXmathchardef\hbar="7 \symmtletterfont "0127 \relax
1148 \else
1149 \ifx\mst@tmp@enc\mst@euii % luatex and unicode font
1150   \luatexUmathchardef\hbar="7 \symmtletterfont "0127 \relax %% or 210F?
1151 \else % I must leave open the T1, OT1 possibility also for XeTeX or LuaTeX
1152   \ifx\mst@ti\mst@tmp@enc
1153     \DeclareMathAccent{\mst@ltbar}{\mathalpha}{\mtletterfont}{9}
1154     \else %% assume OT1. Bad idea? Should not provoke an error anyhow
1155     \DeclareMathAccent{\mst@ltbar}{\mathalpha}{\mtletterfont}{22}
1156   \fi
1157   \def\hbar{\mst@mathrlap{\mst@ltbar{\ }h}}
1158 \fi\fi}
1159 \ifmst@nohbar\else\typeout{** \string\hbar}\do@the@hbarstuff\fi
```

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

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

`+,=\,\Relbar`

```
1160 \ifmst@noplus\else\typeout{** \string+ and \string=}
1161 \DeclareMathSymbol{+}{\mathbin}{\mst@font@tbu}{\ "2B}
1162 \DeclareMathSymbol{\mst@varfam@plus}{\mathalpha}{\mst@font@tbu}{\ "2B}
1163 \expandafter\mst@addtodo@nonletters\string+\mathbin\mst@varfam@plus
1164 \fi
1165 \ifmst@noequal\else
1166 \ifmst@XeTeX
1167   \XeTeXmathcharnumdef\mst@equal@sign=\XeTeXmathcodenum'\=\relax
1168 \else
```

```

1169 \ifmst@LuaTeX
1170 \luatexUmathcharnumdef\mst@equal@sign=\luatexUmathcodenum'\=\relax
1171 \else
1172 \mathchardef\mst@equal@sign=\mathcode'\=\relax
1173 \fi\fi
1174 \@ifpackageloaded{amsmath}
1175 {\def\Relbar{\mathrel\mst@equal@sign}}
1176 {\DeclareRobustCommand\Relbar{\mathrel{\mst@equal@sign}}}
1177 \DeclareMathSymbol{=}{\mathrel}{\mst@font@tbu}{\textfont{3D}}
1178 \DeclareMathSymbol{\mst@varfam@equal}{\mathalpha}{\mst@font@tbu}{\textfont{3D}}

```

\nfss@catcodes 2012/12/18: Activating = (only in math mode actually) seems very bad but surprisingly works well. However I had a problem with eu2lmtt.fd which should not be loaded with an active =. 2012/12/25: Since then I had switched to only math activation. And in fact the problematic = from eu2lmtt.fd end up in \csname...\endcsname and I have learnt since that T_EX does not look at the mathcode inside a \csname...\endcsname. Example:

```

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

```

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

```

1179 \typeout{** adding \string= \string; and \string+ to \string\nfss@catcodes}
1180 \g@addto@macro\nfss@catcodes{%
1181   \@makeother\=%
1182   \@makeother\;%
1183   \@makeother\+%
1184 }
1185 \expandafter\mst@addtodo@nonletters\string=\mathrel\mst@varfam@equal
1186 \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.

```

1187 \ifmst@noparen\else\typeout{** parentheses \string( \string) \string[ \string] and slash \s
1188 \ifmst@nosmalldelims
1189   \DeclareMathSymbol{(}{\mathopen}{\mst@font@tbu}{\textfont{28}}
1190   \DeclareMathSymbol{)}{\mathclose}{\mst@font@tbu}{\textfont{29}}

```

```

1191 \DeclareMathSymbol{[}{\mathopen}{\mst@font@tbu}{5B}
1192 \DeclareMathSymbol{]}{\mathclose}{\mst@font@tbu}{5D}
1193 \DeclareMathSymbol{/}{\mathord}{\mst@font@tbu}{2F}
1194 \else
1195 \DeclareMathDelimiter{({\mathopen}{\mst@font@tbu}{28}{largesymbols}{00}
1196 \DeclareMathDelimiter{)}{\mathclose}{\mst@font@tbu}{29}{largesymbols}{01}
1197 \DeclareMathDelimiter{[}{\mathopen}{\mst@font@tbu}{5B}{largesymbols}{02}
1198 \DeclareMathDelimiter{]}{\mathclose}{\mst@font@tbu}{5D}{largesymbols}{03}
1199 \DeclareMathDelimiter{/}{\mathord}{\mst@font@tbu}{2F}{largesymbols}{0E}
1200 \fi
1201 \DeclareMathSymbol{\mst@varfam@lparen}{\mathalpha}{\mst@font@tbu}{40}
1202 \DeclareMathSymbol{\mst@varfam@rparen}{\mathalpha}{\mst@font@tbu}{41}
1203 \DeclareMathSymbol{\mst@varfam@lbrack}{\mathalpha}{\mst@font@tbu}{5B}
1204 \DeclareMathSymbol{\mst@varfam@rbrack}{\mathalpha}{\mst@font@tbu}{5D}
1205 \DeclareMathSymbol{\mst@varfam@slash}{\mathalpha}{\mst@font@tbu}{2F}
1206 \expandafter\mst@addtodo@nonletters\string(\mathopen\mst@varfam@lparen
1207 \expandafter\mst@addtodo@nonletters\string)\mathclose\mst@varfam@rparen
1208 \expandafter\mst@addtodo@nonletters\string[\mathopen\mst@varfam@lbrack
1209 \expandafter\mst@addtodo@nonletters\string]\mathclose\mst@varfam@rbrack
1210 \mst@addtodo@easynonletters\/\mst@varfam@slash
1211 \fi

```

alldelims

```

<,>,\ 1212 \ifmst@alldelims\typeout{** alldelims: \string< \string>
\setminus 1213 \string\backslash\space\string\setminus\space\string\
\backslash 1214 \string\vert\space\string\mid\space\string{\space and \string\}}
1215 \ifmst@nosmallldelims

```

Dec 18, 2012. We then want `\let\backslash\mst@varfam@backslash` to do nothing when the `\backslash` is used as a delimiter. So here the original definition from `latex.ltx` is copied, generally speaking when people use other math symbol fonts they do respect the encoding of the CM symbols and `largesymbols`, so this is 90% safe. But in truth I should extract from the meaning of `\backslash` the `delcode`.

```

1216 \DeclareMathDelimiter{\mst@varfam@backslash}
1217 {\mathalpha}{symbols}{6E}{largesymbols}{0F}
1218 \else
1219 \DeclareMathDelimiter{<}{\mathopen}{\mst@font@tbu}{3C}{largesymbols}{0A}
1220 \DeclareMathDelimiter{>}{\mathclose}{\mst@font@tbu}{3E}{largesymbols}{0B}

```

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

```

1221 \expandafter\DeclareMathDelimiter\@backslashchar
1222 {\mathord}{mtoperatorfont}{5C}{largesymbols}{0F}
1223 \DeclareMathDelimiter{\backslash}
1224 {\mathord}{mtoperatorfont}{5C}{largesymbols}{0F}
1225 \DeclareMathDelimiter{\mst@varfam@backslash}
1226 {\mathalpha}{mtoperatorfont}{5C}{largesymbols}{0F}
1227 \fi
1228 \DeclareMathSymbol{<}{\mathrel}{\mst@font@tbu}{3C}
1229 \DeclareMathSymbol{>}{\mathrel}{\mst@font@tbu}{3E}
1230 \DeclareMathSymbol{\mst@varfam@less}{\mathalpha}{\mst@font@tbu}{3C}

```



```

1269 \fi
1270 \DeclareMathSymbol{\mst@varfam@lbrace}{\mathalpha}{\mst@font@tbu}{123}
1271 \DeclareMathSymbol{\mst@varfam@rbrace}{\mathalpha}{\mst@font@tbu}{125}
1272 \DeclareRobustCommand*\mst@lbrace{
1273     {\ifmmode\mathopen\mst@varfam@lbrace\else\textbraceleft\fi}
1274 \DeclareRobustCommand*\mst@rbrace{
1275     {\ifmmode\mathclose\mst@varfam@rbrace\else\textbraceright\fi}
1276 \mst@do@nonletters\expandafter{\the\mst@do@nonletters
1277     \mst@dobraces{\let\{\mst@lbrace\let\}\mst@rbrace}}
1278 \fi % end of \ifmst@alldelims
1279 \newcommand*\MTexplicitbracesobeymathxx{\let\mst@dobraces\@firstofone}
1280 \newcommand*\MTexplicitbracesdonotobeymathxx{\let\mst@dobraces\@gobble}
1281 \MTexplicitbracesdonotobeymathxx

```

specials 1.14b [2011/04/02]: the redefinitions of #, \$, % and & were buggy (this showed up when 10 or more math families had been created).

1.15f [2012/10/23]: the code, although working, was perhaps a bit insane and had definitions which could surprise other packages. For example, it did:

```
\renewcommand{\%}{\ifmmode\mt@mmode@percent\else\char37\relax\fi}
```

But it seems this provokes a problem with `microtype`. Perhaps the problem was that the command was not declared robust? For the dollar \LaTeX itself does

```
\DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
```

So here I just modify `\mathdollar`. Then we have in `latex.ltx` the same definitions as in `plain.tex`: `\chardef\%='\'`, `\chardef\&='&`, and `\chardef\#='#`. It turns out that we can just adjust the mathcodes of these characters and achieve exactly what is wanted for the corresponding one char control sequences. In math mode the control sequence will use the specified mathcode. So here it is *not* a redefinition of the control sequences, purely an adjustment of mathcodes.

1.2d [2013/01/01]: previous versions imposed the variable family type. I hereby make it possible to de-activate this feature with the macro `\MTeasynonlettersdonotobeymathxx`. Besides, I have absolutely no idea why I had different looking code depending on the engine XeTeX, LuaTeX or default. Removed.

```

1282 \ifmst@nospecials\else\typeout{** \string#\space\string\mathdollar\space\string%\space\st.
1283 \count255=\symmtooperatorfont
1284 \multiply\count255 by 256
1285 \advance\count255 by 35
1286 \mathcode'\#\count255
1287 \advance\count255 by 1
1288 \mathchardef\mathdollar\count255
1289 \advance\count255 by 1
1290 \mathcode'\%\count255
1291 \advance\count255 by 1
1292 \mathcode'\&\count255
1293 \count255=\symmtooperatorfont
1294 \multiply\count255 by 256
1295 \advance\count255 by 28707 % = "7023
1296 \mathchardef\mst@varfam@mathhash\count255
1297 \advance\count255 by 1
1298 \mathchardef\mst@varfam@mathdollar\count255

```

```

1299 \advance\count255 by 1
1300 \mathchardef\mst@varfam@mathpercent\count255
1301 \advance\count255 by 1
1302 \mathchardef\mst@varfam@mathampersand\count255
1303 \mst@do@easynonletters\expandafter{\the\mst@do@easynonletters
1304 \mathcode'\#=\mst@varfam@mathhash
1305 \let\mathdollar\mst@varfam@mathdollar
1306 \mathcode'\%=\mst@varfam@mathpercent
1307 \mathcode'\&=\mst@varfam@mathampersand}
1308 \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 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?

```

1309 \ifmst@symbolmisc\typeout{** symbolmisc: miscellaneous math symbols from Symbol font}
1310 \let\mst@prod\prod
1311 \let\defaultprod\prod
1312 \DeclareMathSymbol{\prodpsy}{\mathop}{mtpsymbol}{213}
1313 \renewcommand*{\prod}{\ifinner\prodpsy\else\mst@prod\fi}
1314 \let\mst@sum\sum
1315 \let\defaultsum\sum
1316 \DeclareMathSymbol{\sumpsy}{\mathop}{mtpsymbol}{229}
1317 \renewcommand*{\sum}{\ifinner\sumpsy\else\mst@sum\fi}

1318 \DeclareMathSymbol{\mst@implies}{\mathrel}{mtpsymbol}{222}
1319 \DeclareRobustCommand*\implies{\;\;\mst@implies\;}
1320 \DeclareMathSymbol{\mst@impliedby}{\mathrel}{mtpsymbol}{220}
1321 \DeclareRobustCommand*\impliedby{\;\;\mst@impliedby\;}
1322 \DeclareRobustCommand*\iff{\;\;\mst@impliedby\mathrel{\mkern-3mu}\mst@implies\;}
1323 \DeclareMathSymbol{\mst@iff}{\mathrel}{mtpsymbol}{219}
1324 \DeclareRobustCommand*\shortiff{\;\;\mst@iff\;}
1325 \DeclareMathSymbol{\mst@to}{\mathrel}{mtpsymbol}{174}
1326 \DeclareMathSymbol{\mst@trait}{\mathrel}{mtpsymbol}{190}
1327 \DeclareRobustCommand*\to{\mst@to}
1328 \DeclareRobustCommand*\longto{\mkern2mu\mst@trait\mathrel{\mkern-10mu}\mst@to}
1329 \DeclareRobustCommand*\mapsto{\mapstochar\mathrel{\mkern0.2mu}\mst@to}
1330 \DeclareRobustCommand*\longmapsto{\%
1331 \mapstochar\mathrel{\mkern2mu}\mst@trait\mathrel{\mkern-10mu}\mst@to}

```

```

1332 \DeclareMathSymbol{\aleph}{\mathord}{mtpsymbol}{192}
1333 \DeclareMathSymbol{\infty}{\mathord}{mtpsymbol}{165}
1334 \DeclareMathSymbol{\emptyset}{\mathord}{mtpsymbol}{198}
1335 \let\varnothing\emptyset
1336 \DeclareMathSymbol{\nabla}{\mathord}{mtpsymbol}{209}
1337 \DeclareMathSymbol{\surd}{\mathop}{mtpsymbol}{214}
1338 \let\angle\undefined
1339 \DeclareMathSymbol{\angle}{\mathord}{mtpsymbol}{208}
1340 \DeclareMathSymbol{\forall}{\mathord}{mtpsymbol}{34}
1341 \DeclareMathSymbol{\exists}{\mathord}{mtpsymbol}{36}
1342 \DeclareMathSymbol{\neg}{\mathord}{mtpsymbol}{216}
1343 \DeclareMathSymbol{\clubsuit}{\mathord}{mtpsymbol}{167}
1344 \DeclareMathSymbol{\diamondsuit}{\mathord}{mtpsymbol}{168}
1345 \DeclareMathSymbol{\heartsuit}{\mathord}{mtpsymbol}{169}
1346 \DeclareMathSymbol{\spadesuit}{\mathord}{mtpsymbol}{170}
1347 \DeclareMathSymbol{\smallint}{\mathop}{mtpsymbol}{242}
1348 \DeclareMathSymbol{\wedge}{\mathbin}{mtpsymbol}{217}
1349 \DeclareMathSymbol{\vee}{\mathbin}{mtpsymbol}{218}
1350 \DeclareMathSymbol{\cap}{\mathbin}{mtpsymbol}{199}
1351 \DeclareMathSymbol{\cup}{\mathbin}{mtpsymbol}{200}
1352 \DeclareMathSymbol{\bullet}{\mathbin}{mtpsymbol}{183}
1353 \DeclareMathSymbol{\div}{\mathbin}{mtpsymbol}{184}
1354 \DeclareMathSymbol{\otimes}{\mathbin}{mtpsymbol}{196}
1355 \DeclareMathSymbol{\oplus}{\mathbin}{mtpsymbol}{197}
1356 \DeclareMathSymbol{\pm}{\mathbin}{mtpsymbol}{177}
1357 \DeclareMathSymbol{\times}{\mathbin}{mtpsymbol}{180}
1358 \DeclareMathSymbol{\propto}{\mathrel}{mtpsymbol}{181}
1359 \DeclareMathSymbol{\mid}{\mathrel}{mtpsymbol}{124}
1360 \DeclareMathSymbol{\leq}{\mathrel}{mtpsymbol}{163}
1361 \DeclareMathSymbol{\geq}{\mathrel}{mtpsymbol}{179}
1362 \DeclareMathSymbol{\approx}{\mathrel}{mtpsymbol}{187}
1363 \DeclareMathSymbol{\supset}{\mathrel}{mtpsymbol}{201}
1364 \DeclareMathSymbol{\subset}{\mathrel}{mtpsymbol}{204}
1365 \DeclareMathSymbol{\supseteq}{\mathrel}{mtpsymbol}{202}
1366 \DeclareMathSymbol{\subseteq}{\mathrel}{mtpsymbol}{205}
1367 \DeclareMathSymbol{\in}{\mathrel}{mtpsymbol}{206}
1368 \DeclareMathSymbol{\sim}{\mathrel}{mtpsymbol}{126}
1369 \let\cong\undefined
1370 \DeclareMathSymbol{\cong}{\mathrel}{mtpsymbol}{64}
1371 \DeclareMathSymbol{\perp}{\mathrel}{mtpsymbol}{94}
1372 \DeclareMathSymbol{\equiv}{\mathrel}{mtpsymbol}{186}
1373 \let\notin\undefined
1374 \DeclareMathSymbol{\notin}{\mathrel}{mtpsymbol}{207}
1375 \DeclareMathDelimiter{\rangle}
1376     {\mathclose}{mtpsymbol}{241}{largesymbols}{"0B}
1377 \DeclareMathDelimiter{\langle}
1378     {\mathopen}{mtpsymbol}{225}{largesymbols}{"0A}
1379 \fi

```

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

```

1380 \ifmst@symbolre\typeout{** symbolre: \string\Re\space and \string\Im\space from Sym-
bol font}
1381 \DeclareMathSymbol{\Re}{\mathord}{\mtpsymbol}{\C2}
1382 \DeclareMathSymbol{\Im}{\mathord}{\mtpsymbol}{\C1}
1383 \DeclareMathSymbol{\DotTriangle}{\mathord}{\mtpsymbol}{92}
1384 \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.

```

1385 \let\mst@mathord\mathalpha
1386 \mst@goaheadtrue
1387 \ifmst@selfGreek
1388     \def\mst@font@tbu{\mselfGreekfont}
1389 \else
1390 \ifmst@eulergreek
1391     \def\mst@font@tbu{\mteulervm}
1392 \else
1393 \ifmst@symbolgreek
1394     \def\mst@font@tbu{\mtpsymbol}
1395     \let\mst@mathord\mathord
1396 \else
1397 \ifmst@LGRgreek
1398     \mst@goaheadfalse
1399 \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.

```

1400     \mst@goaheadfalse
1401     \ifx\omicron\undefined
1402         \newcommand*{\omicron}{\mst@alph@omicron{o}}
1403     \fi
1404 \fi\fi\fi\fi
1405 \ifmst@goahead
1406 \DeclareMathSymbol{\Alpha}{\mst@mathord}{\mst@font@tbu}{\C41}
1407 \DeclareMathSymbol{\Beta}{\mst@mathord}{\mst@font@tbu}{\C42}

```

```

1408 \DeclareMathSymbol{\Epsilon}{\mst@mathord}{\mst@font@tbu}{ "45}
1409 \DeclareMathSymbol{\Zeta}{\mst@mathord}{\mst@font@tbu}{ "5A}
1410 \DeclareMathSymbol{\Eta}{\mst@mathord}{\mst@font@tbu}{ "48}
1411 \DeclareMathSymbol{\Iota}{\mst@mathord}{\mst@font@tbu}{ "49}
1412 \DeclareMathSymbol{\Kappa}{\mst@mathord}{\mst@font@tbu}{ "4B}
1413 \DeclareMathSymbol{\Mu}{\mst@mathord}{\mst@font@tbu}{ "4D}
1414 \DeclareMathSymbol{\Nu}{\mst@mathord}{\mst@font@tbu}{ "4E}
1415 \DeclareMathSymbol{\Omicron}{\mst@mathord}{\mst@font@tbu}{ "4F}
1416 \DeclareMathSymbol{\Rho}{\mst@mathord}{\mst@font@tbu}{ "50}
1417 \DeclareMathSymbol{\Tau}{\mst@mathord}{\mst@font@tbu}{ "54}
1418 \DeclareMathSymbol{\Chi}{\mst@mathord}{\mst@font@tbu}{ "58}

```

When we in fact use Symbol, we have to correct \Rho and \Chi. And \Digamma is non-existent in fact (no F in Symbol, F codes a \Phi).

```

1419 \ifx\mst@mathord\mathord

```

symbolgreek but neither eulergreek nor selfGreek

```

1420 %% attention le P de Symbol est un \Pi pas un \Rho
1421 \DeclareMathSymbol{\Rho}{\mathord}{mtpsymbol}{ "52}
1422 %% attention le X de Symbol est un \Xi pas un \Chi
1423 \DeclareMathSymbol{\Chi}{\mathord}{mtpsymbol}{ "43}
1424 %% attention le F de Symbol est un \Phi. Il n'y a pas de \Digamma
1425 \DeclareMathSymbol{\Gamma}{\mathord}{mtpsymbol}{ "47}
1426 \DeclareMathSymbol{\Delta}{\mathord}{mtpsymbol}{ "44}
1427 \DeclareMathSymbol{\Theta}{\mathord}{mtpsymbol}{ "51}
1428 \DeclareMathSymbol{\Lambda}{\mathord}{mtpsymbol}{ "4C}
1429 \DeclareMathSymbol{\Xi}{\mathord}{mtpsymbol}{ "58}
1430 \DeclareMathSymbol{\Pi}{\mathord}{mtpsymbol}{ "50}
1431 \DeclareMathSymbol{\Sigma}{\mathord}{mtpsymbol}{ "53}
1432 \DeclareMathSymbol{\Upsilon}{\mathord}{mtpsymbol}{ "A1}
1433 \DeclareMathSymbol{\Phi}{\mathord}{mtpsymbol}{ "46}
1434 \DeclareMathSymbol{\Psi}{\mathord}{mtpsymbol}{ "59}
1435 \DeclareMathSymbol{\Omega}{\mathord}{mtpsymbol}{ "57}
1436 \else

```

not symbolgreek but eulergreek or selfGreek or both

```

1437 \DeclareMathSymbol{\Digamma}{\mathalpha}{\mst@font@tbu}{ "46}
1438 \DeclareMathSymbol{\Gamma}{\mathalpha}{\mst@font@tbu}{ "00}
1439 \DeclareMathSymbol{\Delta}{\mathalpha}{\mst@font@tbu}{ "01}
1440 \DeclareMathSymbol{\Theta}{\mathalpha}{\mst@font@tbu}{ "02}
1441 \DeclareMathSymbol{\Lambda}{\mathalpha}{\mst@font@tbu}{ "03}
1442 \DeclareMathSymbol{\Xi}{\mathalpha}{\mst@font@tbu}{ "04}
1443 \DeclareMathSymbol{\Pi}{\mathalpha}{\mst@font@tbu}{ "05}
1444 \DeclareMathSymbol{\Sigma}{\mathalpha}{\mst@font@tbu}{ "06}
1445 \DeclareMathSymbol{\Upsilon}{\mathalpha}{\mst@font@tbu}{ "07}
1446 \DeclareMathSymbol{\Phi}{\mathalpha}{\mst@font@tbu}{ "08}
1447 \DeclareMathSymbol{\Psi}{\mathalpha}{\mst@font@tbu}{ "09}
1448 \DeclareMathSymbol{\Omega}{\mathalpha}{\mst@font@tbu}{ "0A}
1449 \fi
1450 \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`

```

1451 \ifmst@goahead
1452 \ifmst@eulergreek
1453 \DeclareMathSymbol{\alpha} {\mathalpha}{mteulervm}{0B}
1454 \DeclareMathSymbol{\beta} {\mathalpha}{mteulervm}{0C}
1455 \DeclareMathSymbol{\gamma} {\mathalpha}{mteulervm}{0D}
1456 \DeclareMathSymbol{\delta} {\mathalpha}{mteulervm}{0E}
1457 \DeclareMathSymbol{\epsilon} {\mathalpha}{mteulervm}{0F}
1458 \DeclareMathSymbol{\zeta} {\mathalpha}{mteulervm}{10}
1459 \DeclareMathSymbol{\eta} {\mathalpha}{mteulervm}{11}
1460 \DeclareMathSymbol{\theta} {\mathalpha}{mteulervm}{12}
1461 \DeclareMathSymbol{\iota} {\mathalpha}{mteulervm}{13}
1462 \DeclareMathSymbol{\kappa} {\mathalpha}{mteulervm}{14}
1463 \DeclareMathSymbol{\lambda} {\mathalpha}{mteulervm}{15}
1464 \DeclareMathSymbol{\mu} {\mathalpha}{mteulervm}{16}
1465 \DeclareMathSymbol{\nu} {\mathalpha}{mteulervm}{17}
1466 \DeclareMathSymbol{\xi} {\mathalpha}{mteulervm}{18}
1467 \let\omicron\undefined
1468 \DeclareMathSymbol{\omicron} {\mathalpha}{mteulervm}{6F}
1469 \DeclareMathSymbol{\pi} {\mathalpha}{mteulervm}{19}
1470 \DeclareMathSymbol{\rho} {\mathalpha}{mteulervm}{1A}
1471 \DeclareMathSymbol{\sigma} {\mathalpha}{mteulervm}{1B}
1472 \DeclareMathSymbol{\tau} {\mathalpha}{mteulervm}{1C}
1473 \DeclareMathSymbol{\upsilon} {\mathalpha}{mteulervm}{1D}
1474 \DeclareMathSymbol{\phi} {\mathalpha}{mteulervm}{1E}
1475 \DeclareMathSymbol{\chi} {\mathalpha}{mteulervm}{1F}
1476 \DeclareMathSymbol{\psi} {\mathalpha}{mteulervm}{20}
1477 \DeclareMathSymbol{\omega} {\mathalpha}{mteulervm}{21}
1478 \DeclareMathSymbol{\varepsilon} {\mathalpha}{mteulervm}{22}
1479 \DeclareMathSymbol{\vartheta} {\mathalpha}{mteulervm}{23}
1480 \DeclareMathSymbol{\varpi} {\mathalpha}{mteulervm}{24}
1481 \let\varrho=\rho
1482 \let\varsigma=\sigma
1483 \DeclareMathSymbol{\varphi} {\mathalpha}{mteulervm}{27}
1484 \DeclareMathSymbol{\partial} {\mathalpha}{mteulervm}{40}
1485 \DeclareMathSymbol{\wp} {\mathalpha}{mteulervm}{7D}
1486 \DeclareMathSymbol{\ell} {\mathalpha}{mteulervm}{60}
1487 \else
1488 \ifmst@symbolgreek
1489 \DeclareMathSymbol{\alpha} {\mathord}{mtpsymbol}{61}
1490 \DeclareMathSymbol{\beta} {\mathord}{mtpsymbol}{62}
1491 \DeclareMathSymbol{\gamma} {\mathord}{mtpsymbol}{67}
1492 \DeclareMathSymbol{\delta} {\mathord}{mtpsymbol}{64}
1493 \DeclareMathSymbol{\epsilon} {\mathord}{mtpsymbol}{65}
1494 \DeclareMathSymbol{\zeta} {\mathord}{mtpsymbol}{7A}
1495 \DeclareMathSymbol{\eta} {\mathord}{mtpsymbol}{68}

```

```

1496 \DeclareMathSymbol{\theta}{\mathord}{\mtpsymbol}{"71}
1497 \DeclareMathSymbol{\iota}{\mathord}{\mtpsymbol}{"69}
1498 \DeclareMathSymbol{\kappa}{\mathord}{\mtpsymbol}{"6B}
1499 \DeclareMathSymbol{\lambda}{\mathord}{\mtpsymbol}{"6C}
1500 \DeclareMathSymbol{\mu}{\mathord}{\mtpsymbol}{"6D}
1501 \DeclareMathSymbol{\nu}{\mathord}{\mtpsymbol}{"6E}
1502 \DeclareMathSymbol{\xi}{\mathord}{\mtpsymbol}{"78}
1503 \let\omicron\undefined
1504 \DeclareMathSymbol{\omicron}{\mathord}{\mtpsymbol}{"6F}
1505 \DeclareMathSymbol{\pi}{\mathord}{\mtpsymbol}{"70}
1506 \DeclareMathSymbol{\rho}{\mathord}{\mtpsymbol}{"72}
1507 \DeclareMathSymbol{\sigma}{\mathord}{\mtpsymbol}{"73}
1508 \DeclareMathSymbol{\tau}{\mathord}{\mtpsymbol}{"74}
1509 \DeclareMathSymbol{\upsilon}{\mathord}{\mtpsymbol}{"75}
1510 \DeclareMathSymbol{\phi}{\mathord}{\mtpsymbol}{"66}
1511 \DeclareMathSymbol{\chi}{\mathord}{\mtpsymbol}{"63}
1512 \DeclareMathSymbol{\psi}{\mathord}{\mtpsymbol}{"79}
1513 \DeclareMathSymbol{\omega}{\mathord}{\mtpsymbol}{"77}
1514 \let\varepsilon=\epsilon
1515 \DeclareMathSymbol{\vartheta}{\mathord}{\mtpsymbol}{"4A}
1516 \DeclareMathSymbol{\varpi}{\mathord}{\mtpsymbol}{"76}
1517 \let\varrho=\rho
1518 \DeclareMathSymbol{\varsigma}{\mathord}{\mtpsymbol}{"56}
1519 \DeclareMathSymbol{\varphi}{\mathord}{\mtpsymbol}{"6A}
1520 \DeclareMathSymbol{\partial}{\mathord}{\mtpsymbol}{"B6}
1521 \DeclareMathSymbol{\wp}{\mathord}{\mtpsymbol}{"C3}
1522 \fi\fi\fi
1523 %%
1524 \ifmst@LGRgreek % internal names for subdued implementation (1.15c-1.15e)
1525 \let\mst@origAlpha\Alpha
1526 \let\mst@origBeta\Beta
1527 \let\mst@origGamma\Gamma
1528 \let\mst@origDelta\Delta
1529 \let\mst@origEpsilon\Epsilon
1530 \let\mst@origZeta\Zeta
1531 \let\mst@origEta\Eta
1532 \let\mst@origTheta\Theta
1533 \let\mst@origIota\Iota
1534 \let\mst@origKappa\Kappa
1535 \let\mst@origLambda\Lambda
1536 \let\mst@origMu\Mu
1537 \let\mst@origNu\Nu
1538 \let\mst@origXi\Xi
1539 \let\mst@origOmicron\Omicron
1540 \let\mst@origPi\Pi
1541 \let\mst@origRho\Rho
1542 \let\mst@origSigma\Sigma
1543 \let\mst@origTau\Tau
1544 \let\mst@origUpsilon\Upsilon

```

```

1545 \let\mst@origPhi\Phi
1546 \let\mst@origChi\Chi
1547 \let\mst@origPsi\Psi
1548 \let\mst@origOmega\Omega
1549 %
1550 \let\mst@origalpha\alpha
1551 \let\mst@origbeta\beta
1552 \let\mst@origgamma\gamma
1553 \let\mst@origdelta\delta
1554 \let\mst@origepsilon\epsilon
1555 \let\mst@origzeta\zeta
1556 \let\mst@origeta\eta
1557 \let\mst@origtheta\theta
1558 \let\mst@origiota\iota
1559 \let\mst@origkappa\kappa
1560 \let\mst@origlambda\lambda
1561 \let\mst@origmu\mu
1562 \let\mst@orignu\nu
1563 \let\mst@origxi\xi
1564 \let\mst@origomicron\omicron
1565 \let\mst@origpi\pi
1566 \let\mst@origrho\rho
1567 \let\mst@origvarsigma\varsigma
1568 \let\mst@origsigma\sigma
1569 \let\mst@origtau\tau
1570 \let\mst@origupsilon\upsilon
1571 \let\mst@origphi\phi
1572 \let\mst@origchi\chi
1573 \let\mst@origpsi\psi
1574 \let\mst@origomega\omega
1575 \let\mst@origDigamma\Digamma
1576 \let\mst@origdigamma\digamma
1577 %%
1578 \def\mst@restoregreek{%
1579 \let\Alpha\mst@origAlpha
1580 \let\Beta\mst@origBeta
1581 \let\Gamma\mst@origGamma
1582 \let\Delta\mst@origDelta
1583 \let\Epsilon\mst@origEpsilon
1584 \let\Zeta\mst@origZeta
1585 \let\Eta\mst@origEta
1586 \let\Theta\mst@origTheta
1587 \let\Iota\mst@origIota
1588 \let\Kappa\mst@origKappa
1589 \let\Lambda\mst@origLambda
1590 \let\Mu\mst@origMu
1591 \let\Nu\mst@origNu
1592 \let\Xi\mst@origXi
1593 \let\Omicron\mst@origOmicron

```

```

1594 \let\Pi\mst@origPi
1595 \let\Rho\mst@origRho
1596 \let\Sigma\mst@origSigma
1597 \let\Tau\mst@origTau
1598 \let\Upsilon\mst@origUpsilon
1599 \let\Phi\mst@origPhi
1600 \let\Chi\mst@origChi
1601 \let\Psi\mst@origPsi
1602 \let\Omega\mst@origOmega
1603 %
1604 \let\alpha\mst@origalpha
1605 \let\beta\mst@origbeta
1606 \let\gamma\mst@origgamma
1607 \let\delta\mst@origdelta
1608 \let\epsilon\mst@origepsilon
1609 \let\zeta\mst@origzeta
1610 \let\eta\mst@origeta
1611 \let\theta\mst@origtheta
1612 \let\iota\mst@origiota
1613 \let\kappa\mst@origkappa
1614 \let\lambda\mst@origlambda
1615 \let\mu\mst@origmu
1616 \let\nu\mst@orignu
1617 \let\xi\mst@origxi
1618 \let\omicron\mst@origomicron
1619 \let\pi\mst@origpi
1620 \let\rho\mst@origrho
1621 \let\varsigma\mst@origvarsigma
1622 \let\sigma\mst@origsigma
1623 \let\tau\mst@origtau
1624 \let\upsilon\mst@origupsilon
1625 \let\phi\mst@origphi
1626 \let\chi\mst@origchi
1627 \let\psi\mst@origpsi
1628 \let\omega\mst@origomega
1629 \let\Digamma\mst@origDigamma
1630 \let\digamma\mst@origdigamma
1631 }
1632 %%
1633 \def\mst@setuplgrgreek{%
1634 \let\Alpha\mst@Alpha
1635 \let\Beta\mst@Beta
1636 \let\Gamma\mst@Gamma
1637 \let\Delta\mst@Delta
1638 \let\Epsilon\mst@Epsilon
1639 \let\Zeta\mst@Zeta
1640 \let\Eta\mst@Eta
1641 \let\Theta\mst@Theta
1642 \let\Iota\mst@Iota

```

```

1643 \let\Kappa\mst@Kappa
1644 \let\Lambda\mst@Lambda
1645 \let\Mu\mst@Mu
1646 \let\Nu\mst@Nu
1647 \let\Xi\mst@Xi
1648 \let\Omicron\mst@Omicron
1649 \let\Pi\mst@Pi
1650 \let\Rho\mst@Rho
1651 \let\Sigma\mst@Sigma
1652 \let\Tau\mst@Tau
1653 \let\Upsilon\mst@Upsilon
1654 \let\Phi\mst@Phi
1655 \let\Chi\mst@Chi
1656 \let\Psi\mst@Psi
1657 \let\Omega\mst@Omega
1658 %
1659 \let\alpha\mst@alpha
1660 \let\beta\mst@beta
1661 \let\gamma\mst@gamma
1662 \let\delta\mst@delta
1663 \let\epsilon\mst@epsilon
1664 \let\zeta\mst@zeta
1665 \let\eta\mst@eta
1666 \let\theta\mst@theta
1667 \let\iota\mst@iota
1668 \let\kappa\mst@kappa
1669 \let\lambda\mst@lambda
1670 \let\mu\mst@mu
1671 \let\nu\mst@nu
1672 \let\xi\mst@xi
1673 \let\omicron\mst@omicron
1674 \let\pi\mst@pi
1675 \let\rho\mst@rho
1676 \let\varsigma\mst@varsigma
1677 \let\sigma\mst@sigma
1678 \let\tau\mst@tau
1679 \let\upsilon\mst@upsilon
1680 \let\phi\mst@phi
1681 \let\chi\mst@chi
1682 \let\psi\mst@psi
1683 \let\omega\mst@omega
1684 \let\Digamma\mst@Digamma
1685 \let\digamma\mst@digamma
1686 }
1687 %%
1688 \DeclareMathSymbol{\mst@Alpha}{\mathalpha}{mtlgrfontupper}{65}
1689 \DeclareMathSymbol{\mst@Beta}{\mathalpha}{mtlgrfontupper}{66}
1690 \DeclareMathSymbol{\mst@Gamma}{\mathalpha}{mtlgrfontupper}{71}
1691 \DeclareMathSymbol{\mst@Delta}{\mathalpha}{mtlgrfontupper}{68}

```

```

1692 \DeclareMathSymbol{\mst@Epsilon}{\mathalpha}{mtlgrfontupper}{69}
1693 \DeclareMathSymbol{\mst@Zeta}{\mathalpha}{mtlgrfontupper}{90}
1694 \DeclareMathSymbol{\mst@Eta}{\mathalpha}{mtlgrfontupper}{72}
1695 \DeclareMathSymbol{\mst@Theta}{\mathalpha}{mtlgrfontupper}{74}
1696 \DeclareMathSymbol{\mst@Iota}{\mathalpha}{mtlgrfontupper}{73}
1697 \DeclareMathSymbol{\mst@Kappa}{\mathalpha}{mtlgrfontupper}{75}
1698 \DeclareMathSymbol{\mst@Lambda}{\mathalpha}{mtlgrfontupper}{76}
1699 \DeclareMathSymbol{\mst@Mu}{\mathalpha}{mtlgrfontupper}{77}
1700 \DeclareMathSymbol{\mst@Nu}{\mathalpha}{mtlgrfontupper}{78}
1701 \DeclareMathSymbol{\mst@Xi}{\mathalpha}{mtlgrfontupper}{88}
1702 \DeclareMathSymbol{\mst@Omicron}{\mathalpha}{mtlgrfontupper}{79}
1703 \DeclareMathSymbol{\mst@Pi}{\mathalpha}{mtlgrfontupper}{80}
1704 \DeclareMathSymbol{\mst@Rho}{\mathalpha}{mtlgrfontupper}{82}
1705 \DeclareMathSymbol{\mst@Sigma}{\mathalpha}{mtlgrfontupper}{83}
1706 \DeclareMathSymbol{\mst@Tau}{\mathalpha}{mtlgrfontupper}{84}
1707 \DeclareMathSymbol{\mst@Upsilon}{\mathalpha}{mtlgrfontupper}{85}
1708 \DeclareMathSymbol{\mst@Phi}{\mathalpha}{mtlgrfontupper}{70}
1709 \DeclareMathSymbol{\mst@Chi}{\mathalpha}{mtlgrfontupper}{81}
1710 \DeclareMathSymbol{\mst@Psi}{\mathalpha}{mtlgrfontupper}{89}
1711 \DeclareMathSymbol{\mst@Omega}{\mathalpha}{mtlgrfontupper}{87}
1712 %
1713 \DeclareMathSymbol{\mst@alpha}{\mathalpha}{mtlgrfontlower}{97}
1714 \DeclareMathSymbol{\mst@beta}{\mathalpha}{mtlgrfontlower}{98}
1715 \DeclareMathSymbol{\mst@gamma}{\mathalpha}{mtlgrfontlower}{103}
1716 \DeclareMathSymbol{\mst@delta}{\mathalpha}{mtlgrfontlower}{100}
1717 \DeclareMathSymbol{\mst@epsilon}{\mathalpha}{mtlgrfontlower}{101}
1718 \DeclareMathSymbol{\mst@zeta}{\mathalpha}{mtlgrfontlower}{122}
1719 \DeclareMathSymbol{\mst@eta}{\mathalpha}{mtlgrfontlower}{104}
1720 \DeclareMathSymbol{\mst@theta}{\mathalpha}{mtlgrfontlower}{106}
1721 \DeclareMathSymbol{\mst@iota}{\mathalpha}{mtlgrfontlower}{105}
1722 \DeclareMathSymbol{\mst@kappa}{\mathalpha}{mtlgrfontlower}{107}
1723 \DeclareMathSymbol{\mst@lambda}{\mathalpha}{mtlgrfontlower}{108}
1724 \DeclareMathSymbol{\mst@mu}{\mathalpha}{mtlgrfontlower}{109}
1725 \DeclareMathSymbol{\mst@nu}{\mathalpha}{mtlgrfontlower}{110}
1726 \DeclareMathSymbol{\mst@xi}{\mathalpha}{mtlgrfontlower}{120}
1727 \DeclareMathSymbol{\mst@omicron}{\mathalpha}{mtlgrfontlower}{111}
1728 \DeclareMathSymbol{\mst@pi}{\mathalpha}{mtlgrfontlower}{112}
1729 \DeclareMathSymbol{\mst@rho}{\mathalpha}{mtlgrfontlower}{114}
1730 \DeclareMathSymbol{\mst@varsigma}{\mathalpha}{mtlgrfontlower}{99}
1731 \DeclareMathSymbol{\mst@sigma}{\mathalpha}{mtlgrfontlower}{115}
1732 \DeclareMathSymbol{\mst@tau}{\mathalpha}{mtlgrfontlower}{116}
1733 \DeclareMathSymbol{\mst@upsilon}{\mathalpha}{mtlgrfontlower}{117}
1734 \DeclareMathSymbol{\mst@phi}{\mathalpha}{mtlgrfontlower}{102}
1735 \DeclareMathSymbol{\mst@chi}{\mathalpha}{mtlgrfontlower}{113}
1736 \DeclareMathSymbol{\mst@psi}{\mathalpha}{mtlgrfontlower}{121}
1737 \DeclareMathSymbol{\mst@omega}{\mathalpha}{mtlgrfontlower}{119}
1738 \DeclareMathSymbol{\mst@digamma}{\mathalpha}{mtlgrfontlower}{147}
1739 \DeclareMathSymbol{\mst@Digamma}{\mathalpha}{mtlgrfontlower}{195}
1740 % cf http://milde.users.sourceforge.net/LGR/lgrxenc.def.html

```

```

1741 % et greek.ldf du package babel
1742 \ifmst@subdued\else\mst@setuplgrgreek\fi
1743 \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.

```

1744 \edef\mst@tmp@enc{\encodingdefault}
1745 \mst@goaheadtrue
1746 \ifx\mst@tmp@enc\mst@eui % xetex and unicode font
1747   \XeTeXmathchardef\inodot="7 \symmtletterfont "0131 \relax
1748   \XeTeXmathchardef\jnodot="7 \symmtletterfont "0237 \relax
1749 \else
1750 \ifx\mst@tmp@enc\mst@euii % luatex and unicode font
1751   \luatexUmathchardef\inodot="7 \symmtletterfont "0131 \relax
1752   \luatexUmathchardef\jnodot="7 \symmtletterfont "0237 \relax
1753 \else
1754 \ifx\mst@tmp@enc\mst@ti % T1
1755   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{25}
1756   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{26}
1757 \else
1758 \ifx\mst@tmp@enc\mst@oti % OT1
1759   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
1760   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
1761 \else
1762 \ifx\mst@tmp@enc\mst@lyi % LY1
1763   \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}{16}
1764   \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}{17}
1765 \else
1766   \ifmst@XeTeX\mst@goaheadfalse\else\ifmst@LuaTeX\mst@goaheadfalse\else
1767     \DeclareMathSymbol{\inodot}{\mathalpha}{mtletterfont}
1768     {\expandafter\the\expandafter\csname\mst@tmp@enc\string\i\endcsname}
1769     \DeclareMathSymbol{\jnodot}{\mathalpha}{mtletterfont}
1770     {\expandafter\the\expandafter\csname\mst@tmp@enc\string\j\endcsname}
1771   \fi\fi
1772 \fi\fi\fi\fi\fi
1773 \ifmst@defaultimath\else\typeout{** \string\i\space and \string\j\space}
1774   \ifmst@goahead
1775     \renewcommand*{\imath}{\inodot}
1776     \renewcommand*{\jmath}{\jnodot}
1777     \let\mst@oldi\i \let\mst@oldj\j
1778     \DeclareRobustCommand*{\i}{\ifmmode\inodot\else\mst@oldi\fi}
1779     \DeclareRobustCommand*{\j}{\ifmmode\jnodot\else\mst@oldj\fi}
1780   \fi
1781 \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

```

1782 \edef\mst@tmp@enc{\encodingdefault} %% rather one too many than sorry
1783 \ifmst@mathaccents\typeout{** math accents}
1784 \ifx\mst@ti\mst@tmp@enc
1785 \DeclareMathAccent{\acute}{\mathalpha}{mtoperatorfont}{1}
1786 \DeclareMathAccent{\grave}{\mathalpha}{mtoperatorfont}{0}
1787 \DeclareMathAccent{\ddot}{\mathalpha}{mtoperatorfont}{4}
1788 \DeclareMathAccent{\tilde}{\mathalpha}{mtoperatorfont}{3}
1789 \DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{9}
1790 \DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{8}
1791 \DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{7}
1792 \DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{2}
1793 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{10}
1794 \DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{6}
1795 \else
1796 \DeclareMathAccent{\acute}{\mathalpha}{mtoperatorfont}{19}
1797 \DeclareMathAccent{\grave}{\mathalpha}{mtoperatorfont}{18}
1798 \DeclareMathAccent{\ddot}{\mathalpha}{mtoperatorfont}{127}
1799 \DeclareMathAccent{\tilde}{\mathalpha}{mtoperatorfont}{126}
1800 \DeclareMathAccent{\bar}{\mathalpha}{mtoperatorfont}{22}
1801 \DeclareMathAccent{\breve}{\mathalpha}{mtoperatorfont}{21}
1802 \DeclareMathAccent{\check}{\mathalpha}{mtoperatorfont}{20}
1803 \DeclareMathAccent{\hat}{\mathalpha}{mtoperatorfont}{94}
1804 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{95}
1805 \DeclareMathAccent{\mathring}{\mathalpha}{mtoperatorfont}{23}
1806 \ifx\mst@lyi\mst@tmp@enc % LY1 encoding
1807 \DeclareMathAccent{\dot}{\mathalpha}{mtoperatorfont}{5}
1808 \else
1809 \ifx\mst@oti\mst@tmp@enc\else
1810 \typeout{** mathastext: math accents have been assumed to be^J%
1811 ** as in OT1 encoding.}
1812 \fi
1813 \fi
1814 \fi\fi

```

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

```

1815 \ifmst@defaultsizes\else
1816 \providecommand\@xxxpt{29.86}
1817 \providecommand\@xxxvpt{35.83}
1818 \ifmst@twelve
1819 \def\Huge{\@setfontsize\Huge\@xxxpt{36}}
1820 \def\HUGE{\@setfontsize\HUGE\@xxxvpt{43}}
1821 \typeout{** \protect\Huge\space and \protect\HUGE\space have been (re)-defined.}
1822 \else
1823 \def\HUGE{\@setfontsize\HUGE\@xxxpt{36}}
1824 \typeout{** \protect\HUGE\space has been (re)-defined.}

```

```

1825 \fi
      I choose rather big subscripts.
1826 \def\defaultscriptratio{.8333}
1827 \def\defaultscriptscriptratio{.7}
1828 \DeclareMathSizes{9}{9}{7}{5}
1829 \DeclareMathSizes{\@xpt}{\@xpt}{8}{6}
1830 \DeclareMathSizes{\@xipt}{\@xipt}{9}{7}
1831 \DeclareMathSizes{\@xipt}{\@xipt}{10}{8}
1832 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xipt}{10}
1833 \DeclareMathSizes{\@xvipt}{\@xvipt}{\@xivpt}{\@xipt}
1834 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xvipt}{\@xivpt}
1835 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xvipt}
1836 \DeclareMathSizes{\@xxxpt}{\@xxxpt}{\@xxvpt}{\@xxpt}
1837 \DeclareMathSizes{\@xxxvpt}{\@xxxvpt}{\@xxxpt}{\@xxvpt}
1838 \typeout{** mathastext has declared larger sizes for subscripts.^~J%
1839 ** To keep LaTeX defaults, use option 'defaultmathsizes\string'.}
1840 \fi

1841 \endinput

```