

The **mathastext** package

JEAN-FRANÇOIS BURNOL
jfbu (at) free (dot) fr
Package version: 1.3b (2013/12/11)

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`.

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. 19
Installation	1.2, p. 3	Shape of Greek letters.	
What <code>mathastext</code> does	2, p. 4	Unicode engines	2.10, p. 21
Examples	2.1, p. 4	Compatibility issues	2.11, p. 23
Overview	2.2, p. 6	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 default <code>mathsizes</code> 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. 12	Complete list of options	3.4, p. 36
Italic corrections	2.6, p. 12	Change log	4, p. 39
Other extra glue in math mode	2.7, p. 15		

1 Recent changes and installation instructions

1.1 Recent changes

This is the documentation for version 1.3b of 2013/12/11.

Changes for this release:

1. `\Mathastext` and `\MTDeclareVersion` both accept a final optional argument, a math version name whose settings regarding things not otherwise changed by `mathastext` (like most symbols and large symbols) will be inherited by the declared math version (first optional argument of `\Mathastext` or first mandatory argument of `\MTDeclareVersion`). Typical use will be with this final optional argument set to be `[bold]`.
2. in case of option `symbolmisc`: `\defaultprod` and `\defaultsum` have been renamed to, respectively, `\MToriginalprod` and `\MToriginalsum`.

Changes for release 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

X_YTeX. So **mathastext** version 1.3 under Lua_YTeX *requires* the version of **lua_Ylatex** (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

Obtain **mathastext.dtx**, the **README** and optionally **mathastext.ins** and **mathastext.tex** from CTAN: <http://ctan.org/pkg/mathastext>

* with **mathastext.ins**: run **tex** on **mathastext.ins** to generate the package style file **mathastext.sty** as well as **mathastext.tex** and some test files.

* without **mathastext.ins**: run **tex** on **mathastext.dtx** to generate the package style file **mathastext.sty** as well as **mathastext.tex** and some test files. (and also **mathastext.ins**)

Move the style file **mathastext.sty** to a location where TeX can find it.

In a TDS compliant hierarchy this will be

<TDS>:tex/latex/mathastext/mathastext.sty

documentation: run **latex** thrice on **mathastext.tex** then **dvipdfmx**. Or, run **pdflatex** thrice on **mathastext.dtx**. In the former case the documentation is with source code included, in the latter without. The file **mathastext.tex** can be customized to change the font size or set other options therein.

(Note: one cannot use lualatex/xelatex to compile the documentation.)

```
*test files*: mathastexttestmathversions.tex
               mathastexttestunicodemacs.tex
               mathastexttestunicodelinux.tex
               mathastexttestalphabets.tex
```

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:

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

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

This was typeset using the Times font (available in any T_EX distribution). Let us now be a bit more original and have our mathematics with italic letters from the sans serif font Helvetica, while the letters in text use New Century Schoolbook.

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

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

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

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

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

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

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

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

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

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

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

Notice that the Greek letters also changed according to the *math version*: **mathastext** has indeed some (limited) capabilities to this effect, with its **LGRgreek**

option. This document uses the LGR encoded fonts `cmtt`, `cmss`, and `txr`, which are part of standard T_EX distributions.¹

2.2 Overview

2.2.1 Basic use

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

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}
```

The package records which font is set up for text, at the time it is loaded,² 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.³ The exact list of characters concerned by `mathastext` is a subset of the basic ASCII set:

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

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

The 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.⁴

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

²alternatively 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.

³let’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.

⁴`v1.2` adds a customizable tiny space before ‘ to separate it from the previous letter, this is really needed when using upright letters in math mode with the CM derivative glyph. Compare f' with f' .

- 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 lowered⁵ (and with the correct spaces around it as binary operator). The reason for making it optional is that after this $\$R^*\$$ or $\$R^{\ast}\$$ do not work and have to be written $\$R^{\{*\}}\$$ or $\$R^{\{\ast\}}\$$.

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

2.2.2 sans in math

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

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

2.2.3 always load **mathastext** last

As said already none of the “large” math symbols is modified in any way by **mathastext**. Only loading some math font packages such as **fourier**, **kpfonts**, **mathabx**, **mathdesign**, **txfonts**, **newtxmath**, etc... will change them. Think of loading these packages before **mathastext**, else they might undo what **mathastext** did. The more common symbols can be taken from the Symbol font (option **symbol-misc**, 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”.⁶ Of course it is always possible to leave the responsibility to set up Greek letters in math mode to packages loaded previously to **mathastext**.

⁵the amount of lowering can be customized.

⁶a 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.

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.⁷

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.

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 numerous⁸ questions⁹ 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.¹⁰ There is also an option `frenchmath` to make the uppercase letters nevertheless upright, because this is the way of traditional French mathematical typography.¹¹

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`.

⁷the `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.

⁸this means "more than one."

⁹I thank in particular TARIQ PERWEZ and KEVIN KLEMENT for their kind remarks (chronological order).

¹⁰more precisely stated, the value of `\itdefault` is used.

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

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 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 LGR_{greek} 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]{lmttt}{T1}{lmttt}{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**¹²: `lmttt`, `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 `\Mathastextversion{normal}`.

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

1. in the body of the 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

¹²math versions are discussed in the document `fntguide.pdf` from your T_EX distribution.

package command `\MTversion` (or its synonyms `\mathastextversion`, `\Mathastextversion`, `\MTVersion`), with a mandatory argument `{\langle version_name \rangle}`. It

- checks in case the `subdued` option was specified whether the asked-for math version is `\langle normal \rangle` or `\langle bold \rangle`, 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.¹³ But one can use instead `\renewcommand{\rmdefault}{epigrafica}`,¹⁴ 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}`.

¹³may typically give a ‘too many math alphabets’ error message.

¹⁴sometimes 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.

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 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 T_EX in math mode to work as well as they usually do, the characters from the math italic font obviously should have their bounding boxes wide enough for the glyphs not to collide with other symbols. A letter from a text italic font such as *f* extends way out of its declared bounding box; let us compare the bounding boxes¹⁵ for the letter *f* in the math italic font to the one from the text italic font: \overline{f} vs. \overline{f} .

This could make us think that attempting to use in math a text italic font will lead to disaster. Well, surprisingly the situation is not that bad. Sure $\$f(x)\$$ is wider with the standard math italic $\overline{f(x)}$ (21.31474pt) than it is with the text

¹⁵let’s be honest, we are lying here about what exactly the first of these is bounding; this is explained later!

italic font used in math:¹⁶ $\overline{f(x)}$ (19.74986pt) but we should be surprised that our text italic f did not end up even closer to the opening parenthesis. Why is it so?

The explanation is that T_EX uses in such a situation the *italic correction* for the letter f . The italic correction also exists and is used for the math italic font, it was inserted in $\text{\texttt{\$f\$}}$ without us having to ask anything. Its value is 1.17865pt for the math italic f and 1.8919pt for the text italic f .¹⁷ With the italic corrections included our bounding boxes are indeed more alike: \overline{f} vs \overline{f} .

Without the italic corrections¹⁸ it is \overline{f} vs \overline{f} . I said that $\text{\texttt{\$f\$}}$ included the italic correction automatically, but if we tell T_EX to use the text italic in math, and typeset the alphabet, we obtain something exactly identical to typing the letters in text, hence without any italic correction:

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

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

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

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

$abcdefghijklmnopqrstuvwxy$	text italic in math
$abcdefghijklmnopqrstuvwxy$	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 $\text{\texttt{\$f(x)\$}}$. Another clever thing done by T_EX is to use it for the placement of superscripts; the next examples systematically use the text italic in math. We see that f^j is very different from $\overline{f^j}$... where the latter was coded with $\text{\texttt{\$ \hbox{\itshape f}^j \$}}$. The inputs $\text{\texttt{\$ \mathit{\hbox{\itshape f \!/}^j \$}}$ and $\text{\texttt{\$ \mathit{f}^j \$}}$ give almost identical results: $\overline{f^j}$ vs. $\overline{f^j}$. Close examination reveals that the horizontal spacing is exactly identical, however the exponent in the second case is a bit lower. Anyway, the point is that in the second case the italic correction for f was indeed used.

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

¹⁶we used simply $\text{\texttt{\$ \mathit{f(x)\$}}}$.

¹⁷these values are for the Latin Modern fonts of course.

¹⁸here we give correctly the bounding box for the math italic f ... without its italic correction!

affected, not the horizontal placement.

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

If we try something like this: `\f\/_i^i` we obtain f_i^i . Our overlapping game with `\rlap{\f_i^i}\f\/_i^i` gives f_i^i . We discover that the effect of the explicit italic correction has mainly been to translate the subscript horizontally to be positioned exactly below the superscript!¹⁹ We most probably do *not* want this to happen for our indices and exponents in math mode. So perhaps we can rejoice in how astute `TeX` has been in judiciously using the italic correction data, and there seems to be no need into fiddling with this algorithm which seems to work well even when applied to a text italic font. Actually we may even be of the opinion that the text italic version f_i^i is a bit better-looking than the true math italic f_i^i . . .

But wait... `mathastext` was initially developed to easily use in math mode the document text font not in its italic variant, but as is, so, usually, upright. And upright `TeX` fonts may also have italic correction data! And what I just said about the shift of the superscript with respect to the subscript apply equally well to such a font, if `TeX` has been told to use it. Let's try Latin Modern Upright for letters in math: `\f_i^i` now gives²⁰ f_i^i . We see the italic correction in action for the positioning of the superscript! Compare with `\mathrm{\hbox{f}_i^i}`: f_i^i . Overlapping with `\rlap{\mathrm{f}_i^i}\mathrm{\hbox{f}_i^i}` gives f_i^i and shows that the upright `f` has an italic correction which was used to shift the superscript to the right (and it is now in a slightly lower position). Let's now do `\mathrm{\f\/_i^i}`: this gives f_i^i and the subscript is shifted to the right, and is now on the same vertical axis as the superscript. There are also some slight vertical displacements, `\rlap{\mathrm{f}_i^i}\mathrm{\f\/_i^i}` gives f_i^i .

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

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

¹⁹there are also some tiny vertical displacements of the sub- and superscripts.

²⁰we just use `\mathrm{f}_i^i`.

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.²¹

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 (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_YT_EX does not have the italic corrections. LuaL^AT_EX does, but only for the italic font. pdfL^AT_EX has them for both the italic and the upright font.

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.

²¹the 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.

- 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' .²²

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

Also, when the `subdued` option has been used, the normal and bold math versions have by default zero length skips. Note though that there is no `\MTversion{normal}` done implicitly by the package when the document body starts²³ (*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.

²²the derivative glyph from the `txfonts` math symbols adapts itself better to an upright letter, no skip seems to be needed then.

²³when `\MTversion` is used, `mathastext` resets all font defaults, so for example a `\renewcommand\sfddefault` 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.

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

1. the pre-declared or user-declared (using the `\DeclareMathOperator` or equivalent) operator names obey the math alphabet commands,²⁴
2. and, *optionally*, all non alphabetical characters²⁵ treated by `\mathastext`, *i.e.*, if not disabled by options, `! ? , : ; + - = () [] < > { }`, the asterisk `*`, and `. / | \ # $ % &`²⁶ will also obey the math alphabet commands (when not used as delimiters). The important thing is that the spaces added by T_EX before and after are not modified.

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

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

with the traditional default behavior:

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

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

`\MTmathoperatorsdonotobeymathxx`
`\MTnonlettersdonotobeymathxx`
`\MTeasynonlettersdonotobeymathxx`

for deactivation and

`\MTmathoperatorsobeymathxx`
`\MTnonlettersobeymathxx`
`\MTeasynonlettersobeymathxx`

for activation.^{27 28}

²⁴contrarily to the next feature, this one is not likely to create incompatibilities with other packages, so it is activated by default.

²⁵of course some of them are input preceded by a backslash, and the backslash itself is input as `\backslash`.

²⁶`# $ % &` 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 T_EX 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’.

²⁷these commands are to be used outside of math mode. Their scope is limited to the current L^AT_EX environment or group. They use the `\everymath` and `\everydisplay` mechanism so if the document needs to modify these token lists it has to do so in a responsible manner, extending not annihilating their previous contents.

²⁸when in subdued mode, the math alphabets are the default ones, not the ones modified by `\mathastext`

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

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

The brace control sequences `\{` and `\}` have their own (supplementary) switch, which is (even) less activated by default. The reason is that after `\MTexplicitbracesobeymathxx`, they regrettably can not be used anymore as delimiters: `\big\{` or `\big\}` cease to function and must be replaced by `\big\lbrace` and `\big\rbrace`. But we can now enjoy `\{a, a > b\}`, `\{a, a > b\}`, `\{a, a > b\}`, or even `\{a, a > b\}`.²⁹

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

Actually, `mathastext` does try to pick up most of the ‘small variants’ in the text font: `\left<x\right>` gives $\langle x \rangle$ (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.

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.

²⁹this last example uses the `\mathnormalbold` additional alphabet defined by `mathastext`.

Examples: `\mathbf{<a,b>}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. `\mathbf{\left<a,b\right>}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. `\mathbf{\mathopen{<}a,b\mathclose{>}}` gives $\langle \mathbf{a}, \mathbf{b} \rangle$. The \LaTeX standard behavior for `\mathbf{\mathopen{<}a,b\mathclose{>}}` 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 \TeX installation: the Computer Modern, Latin Modern, and the CM-LGC fonts; the Greek Font Society fonts (such as GFS Didot), the `epigrafica` and `kerkis` packages, the `txfontsb` package which extends the `txfonts` package with LGR-encoded Greek letters; the Droid fonts, the DejaVu fonts, the `Comfortaa` font, and the Open Sans font. The LGR encoded CM/LM fonts (in serif, sans-serif and typewriter family) give the nice Greek letters in upright shape from the `cbfonts` package. To get these letters in your `mathastext` math mode, you can do the following:

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

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

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

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

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

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

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

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

2.9.1 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 **LGRgreeks** or **selfGreeks** (notice the additional “s”), each math

version is assumed to have its text font available in LGR (or OT1 encoding) and also the shapes will be local to the math version.

Finally version 1.15c of `mathastext` introduces new preamble-only commands to change the shapes, and even the font, used for Greek letters, in case of package options `LGRgreek`/`selfGreek`. They are `\MTitgreek`, `\MTupgreek`, `\MTitGreek`, `\MTupGreek`: these are used like the options and change only the shapes for the math versions which will be declared *next* in the preamble; and `\MTgreekfont{name_of_font}` will tell the *next* math versions to use that font family. To use this command you need to know the (little) name of a suitable font family available in LGR encoding: for example `lmr`, `txr` (needs `txfonts` 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 \XeTeX or \LuaTeX , but the user is strongly advised to look first at the `mathspec` package, which is a far more powerful package designed for unicode (only for \XeTeX), and to `unicode-math` (to use OpenType math fonts).

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

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 \XeTeX or \LuaTeX it is recommended to use the `fontspec` package. Else, some of the encoding dependent things done by `mathastext` like using the en-dash character to get a minus sign in math mode will not 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 \XeTeX engine and only partially available for the \LuaTeX engine, with the result that the spacings in math mode when using


```

\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]
\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**, **defaulttt**, etc...sub-options.

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

This is compatible with using `\label` and `\ref` in and outside of math mode. But a difficulty arises when some other package has made the character ‘globally active’ everywhere in the document. The action of `mathastext` is made anew at each mathematical inline or displayed formula. If it is detected that a character has been activated then nothing further will be done (so the `mathastext` feature³¹ 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*.³²

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.³³ 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`.

³¹italic correction insertion for the latin letters, receptivity to the math alphabet action for the other characters.

³²only 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 are possible candidates), but `mathastext` has been designed to cope with it, should it happen ...

³³This seems to still be the case with Babel 3.9f and frenchb.lfd 2.6e, as tested on Sep. 2, 2013.

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 T_EX if some precautions are not taken. Examples: 0^+5 or $x\mathrel{?}y$ or R^*5 *must* be input now as $0^{\{+ \}}5$ and, respectively, $x\mathrel{\{? \}}y$ or $R^{\{* \}}5$. 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 T_EX ‘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 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.³⁴ This specific issue will perhaps be fixed by some hypothetic future release of `amsmath`, or by other packages providing patches, but I decided for a preemptive strike. As a result the declaration above will not cause an error when `\crazy` is used with a Unicode engine, but there are now some spacings around the punctuation characters. To avoid this use (also with `LATEX`):

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

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

hbar: the definition of `\hbar` inherited from default `LATEX` will in our context make use of the `h` of the current math font (so for us, it is also the text font, perhaps in italic shape), but the bar accross the `h` will come from the original default math font for letters (usually `cmmmi`), and furthermore its placement on the `h` can be odd-looking. So we redefine `\hbar` to use only the text font (and this

³⁴To 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.

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 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 lowering³⁵ 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’³⁶ 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_YTeX: 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_YTeX (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 L_AT_EX and PdfL_AT_EX, 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_YTeX 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,

³⁵with 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`.

³⁶in a hopefully safe way, for example `$\label{eq*1}` is ok.

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 `amssymb` package). As it does not exist in the default L^AT_EX math font set-up, this alphabet is *not* subjected to the subdued option action.
 - The other math alphabet changing commands defined by the package are `\MathEulerBold`, `\MathEuler` and `\MathPSymbol`.
 - `\mathnormal`, `\mathrm`, `\mathbf`, `\mathit`, `\mathsf` and `\mathtt` are modified to make reference to the document text fonts (this can be disabled by suitable package options).
 - version 1.2 of `mathastext` has extended the scope of the math alphabets to apply to non-alphabetical characters and to operator names. This respects the automatic white spaces added by T_EX around math symbols.

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

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

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

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 not necessary to use them to activate the package basic functionalities, as loading `mathastext` is enough (except with the `subdued` option).

- `\Mathastext` (or `\mathastext`) reinitializes `mathastext`: it sets the fonts used in math mode (in versions `normal` and `bold`) for letters, digits and a few ascii symbols to the *current* defaults of encoding, family, series and shape.³⁷ Both the normal and bold math version are modified by this action of `\Mathastext`.
- **math versions:** `\Mathastext` accepts an optional argument [*⟨name⟩*]. With this (within square brackets) argument, rather than redefining the fonts for math mode, `\Mathastext` declares a new *math version*, and it is this math version which will use the then current text font in math mode.³⁸
- **inheritance:** starting with version 1.3b a second optional argument [*⟨other_version⟩*] will transfer its set-up for things not affected by `mathastext` action, like large symbols, to the declared math version whose name was given as first optional argument. The main use will be with [`bold`] in order for the symbols and large symbols to be type-set as in the bold math version. For example, this document has in its preamble:

```
\usepackage{newcent}% this package makes New Century the roman font
\Mathastext[newcent]% this math version will use New Century
\MTseries{b}          % next \Mathastext will use a bold font
\Mathastext[boldnewcent][bold]% large symbols, etc, will be bold too
```

We can check that it does work:

`\MTversion{newcent}`: *abcde* $\oint \forall \cup \otimes \oplus$

`\MTversion{boldnewcent}`: ***abcde*** $\oint \forall \cup \otimes \oplus$

Naturally, for this one needs an initial math font setup with some nice bold fonts also for large symbols. This is the case with the excellent `txfonts` package of YOUNG RYU. As the present document must use many fonts and declares many math alphabets, we did not load the full package and fonts but only the `largesymbols`:

```
\DeclareSymbolFont{largesymbols}{OMX}{txex}{m}{n}
\SetSymbolFont{largesymbols}{bold}{OMX}{txex}{bx}{n}
\DeclareFontSubstitution{OMX}{txex}{m}{n}
```

- `\Mathastext` may be preceded optionally by one or more of³⁹ `\MTencoding{⟨enc⟩}`, `\MTfamily{⟨fam⟩}`, `\MTseries{⟨ser⟩}`, `\MTshape{⟨sh⟩}`, and `\MTlettershape{⟨sh⟩}`. For example valid values are, respectively, `⟨T1⟩`,

³⁷`\Mathastext` updates 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*.

³⁸The 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.

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

$\langle phv \rangle$, $\langle m \rangle$, $\langle n \rangle$, and $\langle it \rangle$: this is the Helvetica font in T1-encoding, regular (medium) series, upright shape, and the letters will be in italic shape. Once used their effect applies to all succeeding calls to `\Mathastext`, and can only be undone by using them again.

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

`\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).⁴⁰ When the initial optional argument is absent, and `mathastext` was loaded with the `italic` option, then the default letter shape will be `it`,⁴¹ else letters will have the same shape as used for digits and operator-names.

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

- `\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

⁴⁰I do not especially recommend to use this in real life!

⁴¹more 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.

declared afterwards via `\Mathastext` or `\MTDeclareVersion`. Only effective with `LGRgreek` option.

3.3.2 Commands usable everywhere

- `\MTsetmathskips{\langle a-z/A-Z \rangle}{\langle mu glue \rangle}{\langle mu glue \rangle}`: 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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: $vw\ x\ yz\ A\ BC^{vw\ x\ yz}A^{BC}$. The effect obeys the usual $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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{\langle a-z/A-Z \rangle}`: 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`{ $\langle math glue \rangle$ }: 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`{ $\langle math glue \rangle$ }: 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`{ $\langle math glue \rangle$ }: 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.⁴²
- `\MTnormalprime`: the prime $'$ is kept unchanged, it does not take into account the math glue specified by `\MTprimeskip`.
- `\MTlowerast`{ $\langle dimen \rangle$ }: 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.

⁴²if `\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`.

- `\MTmathoperatorsobeymathxx`, `\MTmathoperatorsdonotobeymathxx`, `\MTnonlettersobeymathxx`, `\MTnonlettersdonotobeymathxx`, `\MTexplicitbracesobeymathxx`, `\MTexplicitbracesdonotobeymathxx`, `\MTeasynonlettersobeymathxx`, `\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.⁴³ To cancel its effect either enclose it in a group or environment or re-issue `\MTicinmath` after it.

3.3.3 Body-only commands

- `\MTversion` [`\nametext`] {`\namemath`}, 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 `\namemath`. If there is an optional argument then the text fonts will use `\nametext` and the math mode will use `\namemath`.

⁴³it is provided for the `mathastext` defined math alphabet `\mathnormalbold` which contrarily to `\mathnormal` is not treated apart from the other math alphabets.

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 `\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`, `\MTeasyonlettersdonotobeymathxx`, 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 *mathastextified* font. The underlying internal L^AT_EX structures related to the original commands are not overwritten, so the original commands can be saved under other names before `\usepackage{mathastext}`, to be used in case of necessity (this is what option `subdued` does.)
- `\mathnormalbold`: a bold version of `\mathnormal`. Differs from `\mathbf` when the *italic* option has been used, or when use has been made of `\MTlettershape` to specify a shape for letters distinct from the one for digits and operator names, or similarly when the math version has been declared via `\MTDeclareVersion` with its optional parameter for shape of letters.
- `\inodot`, `\jnodot`: the corresponding glyphs in the chosen font for math mode. By default, will overwrite `\imath` and `\jmath`. With version 1.12 by

default `\i` and `\j` work also in math mode and give then `\inodot`, resp. `\jn-odot`. 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`.⁴⁴ 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 `\angle` \angle `\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 `\prodpsy` and `\sumpsy`: if one really wants in all situations the Symbol glyphs, one can do `\let\prod\prodpsy` and `\let\sum\sumpsy`. Also `\MToriginalprod` and `\MToriginalsum` will refer to the `\prod` and

⁴⁴option `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.

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

3.4 Complete list of options

- **basic**: only `mathastextify` letters and digits.
- **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 (`\cdot`).
- **symbolmisc**: takes quite a few glyphs, including logical arrows, product and sum signs from Symbol. They are listed *supra*. Doing `\renewcommand{\int}{\smallint}` will maximize even more the use of the Symbol font.
- **symboldelimiters**: the characters apart from letters and digits will be taken from the Symbol font.
- **symbol**: combines **symbolgreek**, **symbolre**, and **symbolmisc**.
- **symbolmax**: combines **symbol** and **symboldelimiters**.
- **eulergreek**, **eulerdigits**: to let Greek letters (digits) use the Euler font.
- **LGRgreek**: this 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’ L^AT_EX error.
- `defaultnormal`, `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.
- `fouriervector`: 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

1.3b [2013/12/11]

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

- * added second optional version name argument to `\Mathastext` and to `\MTDeclareVersion`, to transfer settings for things not otherwise changed by `mathastext` from a math version to the one declared. This is mainly for symbols and large symbols to be boldened also when a bold text font is a `mathastextified` in a `mathastext`-declared version.

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

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

1.3a [2013/09/04]

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

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

- * improved documentation.

1.3 [2013/09/02]

- * commands `\MTsetmathskips` and `\MTunsetmathskips` added.

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

- * the new `\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 series as it seems to correct most of the main problems which were introduced with the massive use of mathematically active characters in versions 1.2 and 1.2b.

- * It is indeed a thorny point when one wants to modify only in math mode how an active character acts, without breaking things. The package now does that `/only/` if the activation appears to originate in the Babel system, as it is then possible to modify appropriately the Babel macros `\user@active<char>` and `\normal@char<char>`. The relevant issues are discussed in section 2.10 of the user manual, in the test file `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 commands (apart from `\mathnormal`), so as to not prohibit the formation of ligatures,
 - * the documentation has been extended to explain in detail the issues which are relevant to the new feature of added italic corrections,
 - * version 1.2 had some bad bugs when confronted to active characters. This is corrected and additionally `\MTnonlettersdonotobeymathxx` is made the default, as the user input is too much constrained in its absence.
 - * a less fatal, but still annoying, typo had made the dot in 1.2 of type `\mathpunct` rather than `\mathord`
 - * the inner namespace has been rationalized a bit.
- 1.2 [2012/12/20]
- * a new command sets up the amount of space to be automatically inserted before the derivative glyph (useful when using an upright font).
 - * the scope of the math alphabets has been extended to apply to the non-alphabetical characters, and also to operator names.
 - * the format of the dtx file has changed. The package file is self-extracting from the dtx, and four additional test files are also produced during 'latex mathastext.dtx'.
- 1.15f and 1.15g [2012/10/25]
- * `\$, \#, \&, \%` 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.

This documentation has been compiled without the source code. To produce the documentation with the source code included, run `"tex mathastext.dtx"` to generate `mathastext.tex` (if not already available), then thrice `latex` on `mathastext.tex` and finally `dvipdfmx` on `mathastext.dvi`.