corrected minor typos in documentation; code not touched
A package for using the MathTime and MathTimePLUS fonts

Drahoslav Lím
DrahoslavL@seznam.cz
2002/04/05

Abstract

mt1ip is a package to relatively painlessly get access to everything available in the MathTime and MathTimePLUS commercial math fonts in \LaTeX. It is compatible with the \AMS packages. This package is self-contained and only the Type 1 font programs need to be copied off of the distribution media.

Contents

1 Introduction 3
2 Installation 3
  2.1 Previewing ........ 4
3 Usage & Package Options 4
  3.1 List of options ....... 5
  3.2 Interaction with \AMS macros and other math packages .... 7
  3.3 Math Accents ....... 7
  3.4 Script and calligraphy math fonts ........ 8
  3.5 The symbols †, ‡, §, and ¶ ... 9
  3.6 “New” symbols ....... 9
  3.7 Avoiding the use of bitmap fonts ...... 9
  3.8 Subscript spacing ...... 9
  3.9 Upright Greek lower case letters ....... 10
  3.10 Oldstyle Digits ...... 11
  3.11 Mixing math versions ... 11
  3.12 Local Configuration .... 11
4 Bugs 11
5 Acknowledgment & Misc. 12
6 Implementation 12
  6.1 Option Declarations .. 13
  6.2 Text Encoding ....... 15
  6.3 Text font setup ....... 15
  6.4 Font Sizing ........ 15
  6.5 Fetch the MathTime and MathTimePLUS fonts ... 16
  6.6 Math declarations ... 17
  6.7 Bold & heavy font substitutions ........ 20
  6.8 Mathcodes ........ 21
  6.9 Auxiliary macro ....... 36
  6.10 Assorted Fixes, Odds & Ends ........ 36
  6.11 \boldsymbol and \heavysymbol ....... 39
  6.12 Examples of new math symbol definitions ....... 40
  6.13 Unused glyphs ....... 42
  6.14 muskip Assignments ....... 43
  6.15 Local Configuration File ....... 43
  6.16 Cleanup ....... 43

*This file has version number 0.4a, last revised 1997/01/27. This file has no connection with the mathtime package commissioned by Y&Y.
1 Introduction

The use of Computer Modern fonts for equations has been one of the tell-tale signs indicating a document was typeset by \TeX. The reasons for this are that there are no good freely available substitute fonts, and that even if there were, changing \TeX's math fonts is a tedious and somewhat cryptic exercise. The first problem can be overcome by purchasing math fonts such as MathTime or Lucida. This package is one of the ways overcoming the second, though not the only one. It provides everything except the fonts themselves. (You have shell out money and buy those, SORRY).

It is assumed you are running \TeX 3.x and a recent release of \LaTeX, and that you have the standard ‘resident’ \TeX fonts installed for use as text fonts in \TeX (“PSNFSS”).

2 Installation

This package is useless unless you buy the fonts called MathTime and MathTimePLUS from Y&Y; the address is listed in Sec. 5.

Prerequisites The “\TeX–based” version of PSNFSS must be installed on your system to use this package, because the \TeX fonts refer to the \TeX “raw” fonts. Search on CTAN for the string “psnfss”, consult the \LaTeX Companion, and/or the \TeX distribution on how to do this. You can check if your system already has the relevant files by searching for the \TeX metric file “ptmr8r.tfm” in the \tfm area, and for the file “T1ptm.fd” (or maybe “t1ptm.fd”) in the \TeX input file area; if both are present you are probably OK and everything will work.

This \dtx file should have come with several sets of accompanying files, whose installation should be routine:

- Run \TeX on mt11p.ins; that will generate the package “style” file, mt11p.sty. Copy that file to the \TeX input file area. Run \LaTeX in mti11p.drv to get commented source code (that’s what you’re reading now).
- The .tfm metric files should be copied to the area where \TeX and the \dviview driver program search for font metrics. The .vf definition files should be copied to the area where the \dviview driver program searches for virtual font definitions. Note that the metric files supplied are not identical to the ones distributed with the MathTime fonts.
- The file mti11p.map should be “passed” to the \dviview driver program to tell it that the MathTime fonts are downloadable \PostScript fonts and what the relevant invocation commands are, or perhaps the file should be passed to a \PostScript rasterizer (“PS–to–PK”) program if the \dviview driver must use PK bitmaps. For Rokicki’s dvips driver, the contents of mti11p.map can be appended to the “psfonts.map” file.
- Copy the Type1 font “programs”—files with extension pfa or pfb—from Y&Y’s distribution media to the area where your \dviview driver or rasterizer searches for downloadable fonts. Some fonts may be provided in both pfa and pfb formats; just pick one, most systems can deal with either. (The pfb format is more compact, albeit binary. The \dviview driver can probably convert .pfb to .pfa “on the fly”.) There are two versions of the normal weight symbol font, mtsy and mtsyn.
The difference is minor, but you might as well use the newer mtsyn. See the paragraph called “Symbols” in the documented source code.\footnote{Run \LaTeX{} on the file \texttt{mt11p.drv} to get the commented source code; it follows after Sec. 5.}

The names and types (pfa/pfb) of the downloadable files in \texttt{mt11p.map} should be made consistent with the actual names and types under which the Type 1 font programs are stored on your system. Either rename and/or convert the Type 1 fonts, or simply edit the contents of \texttt{mt11p.map}; beware of case sensitivity.

The \texttt{dvi} driver program will probably not need the \texttt{afm} files from the distribution medea, but a “PS-to–PK” rasterizer might; consult the rasterizer documentation. Note that \texttt{afm} files are ASCII files, convert line endings as appropriate.

At least for this package, no other files from the distribution medea are required.

### 2.1 Previewing

Whether or not you can preview \texttt{dvi} files which use MathTime fonts depends on your setup. If you can preview \texttt{dvi} files containing other PostScript fonts, chances are you can preview with MathTime as well, perhaps after running some utility to generate bitmaps, resolve VF references, etc. Otherwise, it is possible to view the final PostScript output of the \texttt{dvi} driver on practically all platforms using viewers based on the \texttt{GhostScript} rasterizer, or perhaps using \texttt{DisplayPostScript}.

### 3 Usage & Package Options

To switch to Times text and math, it should suffice to say \texttt{\usepackage{mt11p}} in the preamble. The order of packages should not matter, with one exception: \texttt{mt11p} should be loaded after \texttt{AMS} math packages. Any reasonably cleanly written \LaTeX{} input file will then be typeset with MathTime being used for all standard mathematical symbols that would have otherwise been typeset in Computer Modern. There are two notable exceptions, see Sec. 3.2.

Expect a few dozen lines of warnings as this package is being read, and a notice about substituted fonts at the end of the run. I think it is better to live with these NFSS warnings rather than do some hacks to turn them off.

At the end of the run expect a warning about size substitutions if the \texttt{resize} option was in effect (it is by default). They are are due to the Computer Modern fonts usually having a set of sizes available rather than being declared scalable in the \texttt{.fd} file, i.e., a \texttt{\DeclareFontShape} argument of something like `<10><10.95><12><14.4>` rather than `<->`. Whether or not something should be done about this depends on what Computer Modern glyphs were used. It may be the case that, in fact, none were—the warnings come from NFSS having loaded fonts for, e.g., \texttt{\oldstylenums} for math, eventhough you never actually used them. You can change the \texttt{.fd} files if you system is capable of generating bitmaps on-the-fly or if you’re using PostScript versions of the Computer Modern fonts. Consult the \texttt{\TeX{} Companion} or the standard documentation file \texttt{fntguide.tex}.
3.1 List of options

This package has lots of options controlling some detailed aspects of font and math character handling. In general, use the defaults, unless you have some idea of what the implications of a particular option are.

Options in effect by default are indicated by a double dagger (‡).

‡\texttt{T1} reset the \texttt{encodingdefault} (applicable to text fonts and some math operators, symbols and accents) to T1.

\texttt{OT1} reset the \texttt{encodingdefault} to the archaic \texttt{OT1} encoding; upper case Greek letters are never taken from the text font, so the only reasons to do this are human inertia and legacy code.

\texttt{noenc} assume the current \texttt{encodingdefault} is the desired one; see the commented source code if you use something other than \texttt{T1} or \texttt{OT1}.

‡\texttt{opsafe} use symbols and accents from the \texttt{operators} math family only when it is safe to do so.

\texttt{opnone} do not use the \texttt{operators} math family for accents or symbols, except when this can’t be helped; if \texttt{encodingdefault} is not recognized, this option is auto-invoked.

\texttt{opmax} take as much from the \texttt{operators} math family as possible. Doing so can produce problematic output in some circumstances, but it will make the maximum use of printer resident fonts; don’t use this option unless you know why you are doing so. See the paragraph “Symbols from either \texttt{operators} or from elsewhere” in the documented source code.

‡\texttt{opaccents} take math accents from the \texttt{operators} math family, and make them follow math alphabet changes where possible; see Sec. 3.3.

\texttt{symaccents} take math accents from the a fixed (within a math version) font; if \texttt{encodingdefault} is not recognized, this option is auto-invoked; see Sec. 3.3.

‡\texttt{mtcal} make the MathTime script font the math alphabet \texttt{mathcal} and make \texttt{mathscr} and synonym of \texttt{mathcal}; see Sec. 3.4.

\texttt{cmcal} make the MathTime script font the math alphabet \texttt{mathscr} and the Computer Modern symbol font the \texttt{mathcal} alphabet; see Sec. 3.4.

\texttt{nfmtms} use the Computer Modern symbol font as the \texttt{mathcal} alphabet and do not load the MathTime script font at all; see Sec. 3.4.

‡\texttt{dohbar} overwrite any existing definition of the \texttt{\textbackslash hbar} (\bar{h}) symbol with one synthesized from an “\textbackslash h” and a bar accent. The definition here is actually rather decent, unlike the simple one usually used. (At least try it before resorting to the glyph in the \texttt{AMS} fonts...)
nohbar do nothing with the \hbar definition, retain the existing one (if any); e.g., the \AMS fonts have \hbar as a single Computer Modern-like glyph.

‡subs the bold and heavy math versions in principle need some ‘ultra’ and ‘poster’ weight fonts; usually no such fonts are available, so define substitutes.

nosubs assume suitable fonts are available and will be loaded, e.g., using a configuration file mt11p.cfg; do not declare any font substitutions. If should be apparent from the commented source code or from mt11p.sty what family/series/shape NFSS declaration the extra fonts should get. Search for the string “SUBS” in the .dtx or .sty file.

‡do8r use the raw 8r fonts for the four symbols §, ¶, †, ‡. The math italic fonts use the italic 8r raw fonts, so it’s not unreasonable to assume the upright ones are also available; no support for using the 8r encoding need be installed on the system, though we make use of it if it is. 8r is not used for anything other than these four symbols; see Sec. 3.5.

no8r do not use the raw fonts, and take the four symbols §, ¶, †, ‡ from a Computer Modern font; see Sec. 3.5.

nodag don’t touch the definitions of §, ¶, †, ‡ at all; assume suitable definitions have been or will be supplied, e.g., as part of a TS1 setup. See Sec. 3.5.

‡bsy load the \AMS packages needed to define the \boldsymbol macro, and define analogous \heavysymbol and \normalsymbols macros.

nobsy don’t load \AMS packages for \boldsymbol and define \boldsymbol and \heavysymbol to produce messages and behave as no-ops.

‡mathgr get upright Greek letters as the math alphabet \mathgr; see Sec. 3.9.

nomathgr do not load the math alphabet \mathgr; this is only useful if you are running out of math families, in which case you can use \greekshape in an \hbox with the obvious limitations. See ec. 3.9.

‡mathOS get oldstyle digits as a math alphabet; Computer Modern oldstyle digits are always available in text.

nomathOS do not load Computer Modern oldstyle digits as a math alphabet. See the nomathgr option.

‡resize change the subscript and text font sizing to better account for the fact that we are using linearly (not optically) scaled fonts.

noresize don’t touch the math subscript and text font sizing.

‡activesb to improve the spacing in certain subscript situations, the subscript character “_” (an underscore) is made active; this might break some macros; see Sec. 3.8
noactivesb don’t do anything with the underscore.

‡times assume the package (style file) which changes the roman, sans and typewriter
text families to “Times & Helvetica & Courier” is called “times.sty”; this
package is loaded via \RequirePackage.

* any other option will be interpreted as the name of the style file to use
instead of times; e.g., the option “btdutch” will attempt to ‘require’ the
btdutch package instead of the times package.

3.2 Interaction with \AMS macros and other math packages

This package does nothing terribly anti-social and can co-exist with \AMS packages, e.g., amsmath, amsfonts, etc., provided those packages are read before (i.e.,
prior to) reading this package. Some types of macros are hardwired to Com-
puter Modern fonts, e.g., the “\AMS” logo and would have to be redefined. The
fonts msam and msbm will, or course, still be used for the more uncommon math
symbols. See below about math accents.

The default \TeX math setup treats upper case Greek letters as mathematical
“letters” and lower case Greek letters as mathematical “symbols”. Hence \mathbf{\Gamma\gamma}
produces a bold upper case Γ, but does not affect the
lower case γ; similarly for \mathit. This package uses the same conventions as the
AMS macros: names like “\varGamma” are used for italic upper case Greek (Γ),
and math version switching is needed (\boldsymbol) to get bold Greek. The
‘legacy’ input code \mathbf{\Delta\Gamma} (“ΔΓ”) does not work
with the AMS \LaTeX macros nor with this package. See Sec. 3.3 about accents in
math.

It is likely that other math font related packages can co-exist with mt11p,
particularly if they merely add a math alphabet or symbol font, e.g., mathbbol,
stmaryrd or wasysym. The chances of running out of math symbol font families
are slim unless you also load the oldlfont package. Note that the AMS symbol
packages need two math families. Of course it makes no sense to use mt11p to-
gether with another complete math setup package like euler.

One package that will clash with mt11p is yhmath. If you load that package
after mt11p, all largesymbols in all math versions will be taken from the fixed
size font yhcmex10; loading yhmath before mt11p will break yhmath’s wide math
accents. If you only need the extra-wide accents, change all instances of the word
“largesymbols” in the file yhmath.sty to something else, e.g., YHLargesymbols, and
input that altered file. If you want the extra delimiter sizes, you’re on your own,
but see the paragraph “Declaring additional math delimiters” towards the end of
the commented source code.

3.3 Math Accents

MathTime provides an additional accent called \widebar or \overbar. It’s
about 1em long. If the AMS macros \Hat, etc., are defined, we also define the
similar \Overbar and \Widebar.

The Computer Modern layout causes some encoding-related problems with
math accents and math alphabets. NFSS has hooks to fix the problems, but does
not actually do it; the \texttt{AMS} package \texttt{amsmath} does fix them by redefining most math accents.

\texttt{mt11p} incorporates fixes similar to the \texttt{AMS} ones, and overwrites any existing math accent definitions (it pretty much has to), so that any combination of math alphabets and accents (and accents options: \texttt{opaccents} or \texttt{symaccents}) will output the expected glyphs. Some curious combinations will, however, lead to odd-looking accent positioning. In a construct such as \texttt{\textit{\hat{x}}}, the accent positioning will always be “wrong”, because the accent is a \textit{text} accent—the (text) italic font implied by \texttt{\textit{}} doesn’t have the appropriate font metrics because it’s assumed that \texttt{\textit{}} will be used when a group of letters (a ‘word’) is to be treated as a math symbol, e.g., \texttt{x_{\text{initial}}} or \texttt{x_{\text{final}}}. However, “\texttt{x}” is a mathematical symbol by itself. If you want an italic hat on the symbol \texttt{x} you have to say \texttt{\textit{\hat{\normal{x}}}} (assuming it’s a big deal to have an italic \texttt{\hat{...}}). Even then the spacing will not be optimal and will usually require some fiddling with the \texttt{\skew} macro.

### 3.4 Script and calligraphy math fonts

This package makes the distinction between two “calligraphy” math alphabets: \textit{calligraphy} and \textit{script}. This is necessitated by the fact that the Computer Modern fonts only have upper case “calligraphy” letters, and that you perhaps may not like the MathTime \texttt{mtms*} fonts or that you may want to use letters from both fonts.

Irrespective of the options in use, the two math alphabet commands \texttt{\mathcal} and \texttt{\mathscr} will be defined. (The old \texttt{\mit} is not defined.)

The default (\texttt{mtcal}) is to make both alphabets refer to a \texttt{mtms*} font. The variant forms of some letters are available with the commands \texttt{\varA}, \texttt{\varE}, \texttt{\varG}, \texttt{\varvarG}, \texttt{\varI}, \texttt{\varL}, \texttt{\varQ}, \texttt{\varS}, \texttt{\varr}, \texttt{\varz}, \texttt{\iscript} and \texttt{\jscript}. (\texttt{\i} and \texttt{\j} are invalid in math mode and \texttt{\imath} and \texttt{\jmath} cannot be redefined.) Both upper and lower case letters will work in \texttt{\mathcal}.

Another possibility, \texttt{cmcal}, is to have both Computer Modern and MathTime fonts available; \texttt{\mathscr} will be the latter, and \texttt{\mathcal} the former. \texttt{\mathcal} will then not have lower case letters.

A third setting, \texttt{nomtms}, avoids using the MathTime \texttt{mtms*} fonts entirely. \texttt{\mathcal} is then the Computer Modern symbols font, and \texttt{\mathscr} gives a warning before calling \texttt{\mathcal}.

A certain amount of “reasonableness” is assumed with respect to the usage of “calligraphic” alphabets. The \texttt{mtcal} option does not give any warning about the fact that \texttt{\mathcal} and \texttt{\mathscr} are identical. With the \texttt{cmcal} or \texttt{nomtms} options in effect you will get warnings if you try to use a \texttt{\varX} letter in \texttt{\mathcal}, but \texttt{\mathcal} will never try to check its argument to verify that it consists of upper case letters only, even when no font with “calligraphy” lower case letters is available. Consider using \texttt{\mathcal} when the argument is known to contain nothing but upper case letters and use \texttt{\mathscr} when using lower case letters as well; that way you will at least get a warning pertaining to script math alphabets if the options are set to avoid the use of \texttt{mtms*} or if the document is reset without MathTime fonts.

The MathTime manuals describe a \texttt{\script} macro; \texttt{mt11p} defines an equivalent macro. Note that \texttt{\script} should only be followed by white space, an optional ‘\textit{*}’, and then a single letter, with no space between the asterisk and the
letter, e.g., $\textsc{\textasteriskcentered}AB$ is equivalent to $\textsc{\mathcal{\textasteriskcentered}}B$, not $\textsc{\mathcal{\textasteriskcentered}}(AB)$, and $\textsc{\textasteriskcentered} AB$ is an error.

### 3.5 The symbols †, ‡, §, and ¶

The standard \LaTeX\ setup takes these symbols, in both text and math from the Computer Modern symbols font. The MathTime symbols font does not have these glyphs, so something must be done to have them available for footnote markers, etc.

There is a fairly standard PostScript text symbol font setup, which uses the TS1 encoding, but it seems to not (yet) be in general use.

This package offers three relevant options. The option \texttt{nodag} does absolutely nothing with the four symbols, but this can only be used if they’ve already been or are about to be redefined (perhaps in the \texttt{mt1ip.cfg} file), because the default definitions assume the use of Computer Modern fonts and \textit{will} produce incorrect glyphs after switching to MathTime.

The default option \texttt{do8r} takes these four symbols (and \textit{only} these four symbols) from the “raw” \texttt{8r} encoded fonts, upon which the virtual \texttt{T1} and \texttt{OT1} PostScript fonts are built. If .fd files or an \texttt{8r.sty} are available, we use them; if they’re not, we supply just enough code to be able to use the symbols in NFSS. It is not unreasonable to do so, because the \texttt{8r} metric file \texttt{ptmri8r.tfm} must be on the system (the \texttt{mtmi.vf} virtual font uses it), so the upright font \texttt{ptmr8r.tfm} is probably available too. Note that \texttt{8r} is an “unsupported” encoding.

The option \texttt{no8r} overwrites any existing definition and forces the use of Computer Modern for these four symbols (without affecting anything else) and avoids the direct use of \texttt{8r} raw fonts.

### 3.6 “New” symbols

MathTime has a few mathematical symbols that aren’t available in Computer Modern. Here is the list, with alternate names in parentheses: $\overline{}$ (\texttt{\textbar{}}), \texttt{openclubsuit}, \texttt{shadedclubsuit}, \texttt{openspadesuit}, \texttt{shadedspadesuit}, \texttt{triangleleft}, \texttt{triangleright}, \texttt{\textcupprod}, \texttt{\textcapprod}, \texttt{\textvarcirc} (\texttt{\textcomp}).

### 3.7 Avoiding the use of bitmap fonts

Often the point of buying the MathTime fonts is not just to get something other than Computer Modern math, but also to avoid the use of bitmap (Type 3) fonts and use scalable (Type 1) fonts instead. The default option are set so that this condition is satisfied, as long as you don’t use glyphs that are not available in either the resident fonts nor in MathTime, i.e., oldstyle digits, or you force the use of Computer Modern or other MetaFont fonts in some situations (the \texttt{cmcal}, \texttt{nomtms}, or \texttt{no8r} options, \texttt{AMS} packages, etc.) and you do not have substitute outline fonts. See the MathTime licence agreement about “exporting” files containing outline fonts.

### 3.8 Subscript spacing

The sidebearings of math italic letters with descenders are such that a spacing adjustment is required in certain contexts, in particular in the very common case
of \( j \) subscripts. The MathTime distribution suggests using an active subscript character, which checks if the first thing in a subscript is one of the letters \( f \), \( j \) or \( p \), and inserts a small negative kern if it is. The macro \texttt{\textbackslash adjust} can be used if the correction needs to be done by hand; see Appendix B.2 of the MathTime Reference Manual.

Changing category codes is akin to Russian roulette: we can try to make the magazine larger, but it will still hold a bullet somewhere. It is unreasonable to assume that other people’s packages will take steps to work even when the underscore is a macro. \texttt{\textunderscore} can crop up in font names or reference labels, for example.

The macro we use is slightly different from the original one in the MathTime distribution. The \texttt{\textcatcode} of the underscore ends up being 12, i.e., “other” rather than 13 (“active”), which is much more forgiving. We rely on the fact that the underscore’s \texttt{\textmathcode} is “8000 to get subscripts in math mode. This seems to be \texttt{ok} when used in most contexts. In text mode this will produce whatever the current font has in slot ‘\textunderscore’, but you shouldn’t be using a ‘naked’ underscore in text anyway...

Nothing is done with the underscore unless its \texttt{\textcatcode} is “subscript” (8), and its \texttt{\textmathcode} is “8000; if both conditions are not fulfilled, someone has presumably already fiddled with the underscore, so we don’t touch it.

The option \texttt{noactivesb} keeps the underscore as it was, in order to quickly ascertain whether the active underscore is braking some macros. Reports of such cases are welcome.

### 3.9 Upright Greek lower case letters

MathTime contains an upright Greek letters (text) which are used in some disciplines to designate units or particles. Such letters are normally not used as math symbols. We provide three relevant commands: \texttt{\textgreekshape}, which is a font switch like \texttt{\textitshape}, \texttt{\texttextgr}, which takes an argument similarly to \texttt{\textit}, and the math alphabet command \texttt{\textmathgr}.

Alternate forms of some letters can be obtained as ligatures—see Section 8 of the MathTimePLUS reference manual. However, in math (\texttt{\textmathgr}) it is not possible to use the ligatures with the asterisk character (\texttt{*}) suggested in the manual, because the asterisk will not be interpreted as a character from the same font as the Greek letters. The TFM metric files accompanying this package contain extra ligatures which produce the variants: in lieu of \texttt{\textomega*} or \texttt{\textphi*}, write \texttt{\textomega\textnu} or \texttt{\textphi\textnu}, respectively; in place of \texttt{\textepsilon*} write \texttt{\textupsilon\textepsilon\textnu}.

This package does not define any \texttt{\textgr} font switch. \texttt{\textgreekshape} performs the same function in text; use the \texttt{\textmathgr} form in math.

MathTimePLUS comes with two “hybrid” fonts, \texttt{mtmub} and \texttt{mtmu}, which are math letters fonts with upright (instead of sloping) lower case Greek letters. The two fonts contain both upright and italic upper case Greek letter. For reasons listed in the reference manual, there is no font \texttt{mtmu}. This package comes with a \texttt{mtmu} font, created as a VF using \texttt{mtmi} and \texttt{mtgu}, in case you decide to use such a font after all.

See the commented source about declaring upright Greek letters as math symbols and about using \texttt{mtgu}. 

10
3.10 Oldstyle Digits

None of the "standard 35" PostScript fonts has oldstyle digits (1234567890) and neither do any of the MathTime fonts. The standard \TeX math letters font has such digits, and some packages make use of the glyphs. \texttt{mt11p} defines an \texttt{\oldstylenums} macro which is essentially equivalent to the one provided in \texttt{\LaTeX}, but note that the Computer Modern oldstyle digits don’t go well with Times text fonts. A \texttt{mt11p.cfg} file or a subsequent package could redefine the macro to use a more suitable font.

3.11 Mixing math versions

Like both the standard \TeX math setup and the \texttt{\AMS} math setup, we assume that it will be relatively rare to require bold (unbold) symbols in a formula set in the unbold (bold) math version. We load the \texttt{\AMS \amsbsy} package which defines a \texttt{\boldsymbol} macro, that sets its argument using the bold math version, and define corresponding \texttt{\heavysymbol} and \texttt{\normalsymbol} macros.

However, it’s awkward to use the \texttt{\boldsymbol}, etc., macros in situations where alignment is critical (accents) or for large delimiters. For those cases one could declare additional symbol fonts and define things like \texttt{\boldalpha}, \texttt{\boldhat} or \texttt{\Lheavybrace}, and then an input text like

\[
\left\Lheavybrace 2\boldalpha (\text{some big formula}) \right\Rheavybrace
\]

would do what would otherwise take unreadable input code (\texttt{\vphantom} constructs) to produce. This requires some knowledge of the details of \TeX’s math workings, but nothing too esoteric. For a simple example the commented source code. Such a setup could be placed in \texttt{mt11p.cfg}. Consult also the standard \texttt{\LaTeX} documentation file \texttt{fntguide.tex}. Note that the \texttt{\AMS} macros do not provide such a setup, because it’s somewhat rare to actually need it, but it can be done.

3.12 Local Configuration

After setting everything up but before ‘undefining’ internal auxiliary macros and flags, and before changing the subscript sizing this package inputs the file \texttt{mt11p.cfg}, if it can find one. Further definitions (bold or upright Greek letters as symbols, etc.) could be placed in such a file.

4 Bugs

Any reports about bugs, suspected bugs or package clashes are welcome provided you also send just enough \TeX code to reproduce the bug and your \TeX .log file—use the e-mail address given on the title page of this document and at various places in the files. Report of usage are also welcome;-).
5 Acknowledgment & Misc.

This section has been put here, in case you don’t print out the commented source code.
Thanks to the authors and maintainers of \LaTeX, and particularly to the authors of NFSS.
Portions of this package were stolen from the \texttt{mathtime} package by Aloysius Helminck, and from the \TeX files provided with the MathTime distribution.
The “operative” part of this package was written before the release of the \texttt{mathtime} package by Y\&Y.
MathTime is a trademark of Publish or Perish, Inc.
A bunch of other names are trademarks or registered trademarks, as everybody knows...
The contact address for Y\&Y is

\begin{verbatim}
Y\&Y, Inc.
106 Indian Hill
Carlisle, MA 01741
USA
http://www.yandy.com
\end{verbatim}

This software is provided “AS-IS”, with no warranty of any kind; the user assumes all responsibility for its use.
The author has no association with Y\&Y, Inc., other than having purchased the MathTime fonts.

6 Implementation

We start by the usual identifications. We need a few commands not present in early versions of NFSS\_2, so check for the date.

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{mt11p} \[
\filedate: setup \space for \space Times \space + \space MathTime \space + \space MathTimePlus]
\end{verbatim}

Before we do anything real, check the catcodes of a few characters in case some fanatic changed them already. These are the characters I’ve seen activated in various packages.
We hope the the single left quote wasn’t...

\begin{verbatim}
\chardef\mt@quotedbl \catcode'"
\chardef\mt@quotesinglr\catcode'
\chardef\mt@colon \catcode':
\chardef\mt@question \catcode'?
\chardef\mt@semicolon \catcode';
\chardef\mt@exclam \catcode'!
\@makeother" \\
\@makeother' \\
\@makeother': \\
\@makeother'? \\
\@makeother'; \\
\@makeother'!
\end{verbatim}
6.1 Option Declarations

We will need the option setting in conditionals at various points, so use integers `(\chardefs)` as flags.

There are three options relevant to text font encodings: either make the default T1, or make the default OT1, or don’t touch `\encodingdefault`. The upright “roman” (`\rmdefault`) is used for the operators math family, so the choice of text encoding has implications for math. This package can handle (in as much as it needs to) with any of the following: T1, OT1 or a superset thereof, 8r and 8a. See the commented source code if you need more information, Sec. 6.8.

\begin{verbatim}
17 \DeclareOption{T1}{\def\mt@enc{\def\encodingdefault{T1}}}
18 \DeclareOption{OT1}{\def\mt@enc{\def\encodingdefault{OT1}}}
19 \DeclareOption{noenc}{\let\mt@enc\relax}
\end{verbatim}

Next the options for dealing with what math symbols to take from the operator family. Either take everything from elsewhere, or take ‘safe’ things from operators, or take as much as possible from operators, which could be dangerous in some circumstances; see Sec. 6.8.

\begin{verbatim}
20 \DeclareOption{opnone}{\chardef\mt@ops0␣}
21 \DeclareOption{opsafe}{\chardef\mt@ops1␣}
22 \DeclareOption{opmax}{\chardef\mt@ops2␣}
\end{verbatim}

Accents can either come from operators and follow math alphabets (usual case), or be always taken from symbols.

\begin{verbatim}
23 \DeclareOption{opaccents}{\chardef\mt@accs0␣}
24 \DeclareOption{symaccents}{\chardef\mt@accs1␣}
\end{verbatim}

We need options to control the handling of script/calligraphy fonts.

\begin{verbatim}
25 \DeclareOption{mtcal}{\chardef\mt@cal0␣}
26 \DeclareOption{cmcal}{\chardef\mt@cal1␣}
27 \DeclareOption{nomtms}{\chardef\mt@cal2␣}
\end{verbatim}

In order to cooperate with AMS macros, we provide a switch to not define `\hbar`, i.e., retain the present definition (if any). You may want to use this to keep `\hbar` pointing to the glyph in the `msbm` fonts, rather than use the faked one we define here, though then you’ll have a Computer Modern “h” which looks too light among Times, and the one provided herein isn’t too shabby...

\begin{verbatim}
28 \DeclareOption{dohbar}{\chardef\mt@hbar1␣}
29 \DeclareOption{nohbar}{\chardef\mt@hbar0␣}
\end{verbatim}

The bold and heavy math versions will probably need some font substitutions, but if you have the additional fonts, you can skip the substitutions contained herein.

\begin{verbatim}
30 \DeclareOption{subs}{\chardef\mt@sub1␣}
31 \DeclareOption{nosubs}{\chardef\mt@sub0␣}
\end{verbatim}

For historical reasons the `\dag`, `\ddag`, `\S` and `\P` symbols have to be dealt with. The `mtmi` virtual fonts use `ptmi8r` raw fonts, so at least the TFM files should be available, and we include just enough code to use them for these four symbols. Provide an option to explicitly avoid using `8r` for these symbols; in that case we’ll have to use OMS. A subsequent package could redefine them to use the TS1 encoding or something. Also provide an option that does nothing at all with any of the relevant definitions.

\begin{verbatim}
32 \DeclareOption{no8r}{\chardef\mt@eightR0␣}
\end{verbatim}
It may be convenient to get the \texttt{\textbackslash boldsymbol} macro from the \texttt{AMS} style files.

Before exiting this package should reset the subscript sizing, because we’re using linearly scaled fonts. Provide an option to switch that off.

This is mainly to test if the active underscore is causing any problems.

Now a few options to save math families. The default leaves room for four, which may be too little if you also load \texttt{AMS}, etc.

The package to get “Times/Helvetica/Courier” might have some wiered name:

The defaults are

• \texttt{T1} encoding;
• assume the text font package is called \texttt{times.sty};
• use \texttt{operators} for safe glyphs and accents;
• use \texttt{mtms} as the \texttt{\mathcal} font;
• declare substitutions for the bold and heavy math versions;
• overwrite any existing definition of \texttt{\hbar};
• use \texttt{8r} for \texttt{\dag}, etc.;
• fetch the \texttt{\textbackslash boldsymbol} \texttt{AMS} macro;
• get upright Greek and oldstyle digits;
• switch to different subscript sizing; and
• use the active underscore character.

Then read the option list.

\texttt{\textbackslash ExecuteOptions{T1, times, ops\textunderscore safe, op\textunderscore accents, mt\textcal, \%}
\texttt{dohbar, subs, do\texttt{8r}, b\texttt{sey}, resize, mathgr, math\texttt{OS}, actives\texttt{b}}}
6.2 Text Encoding

We have to know that the text \texttt{encodingdefault} is, because accents in math mode may get taken from the \texttt{operators} font. We make \texttt{encodingdefault} either T1 (default, and preferred), or OT1, or do nothing with \texttt{encodingdefault}.

6.3 Text font setup

Here we fetch “Times + Helvetica + Courier” using the \texttt{times} package, or whatever we were told to use, or just do it by hand. On exit \texttt{rmdefault}, \texttt{sfdefault} and \texttt{ttdefault} should be OK.

6.4 Font Sizing

The documentation accompanying MathTime suggests different subscript sizing. We do it the same way the mathtimy package does it, by redefining things like \texttt{@ixpt} to actually be something other than what the name implies. Then we reset all math sizes, from \texttt{tiny} to \texttt{Huge}. This is the sizing used in mathtimy, which appears to be taken from the files in the MathTime distribution. This macro gets called just before we exit. Switching between MathTime sizing and the default Computer Modern sizing on the fly makes little sense, so we don’t try to preserve the original settings. The standard sizes are, in sequence, 5, 6, 7, 8, 9, 10, 10.95, 12, 14.4, 17.28, 20.74 and 24.88 points, i.e., we decrease the ‘spread’ somewhat.
6.5 Fetch the MathTime and MathTimeplus fonts.

We don’t create .fd files, since the fonts are not for general use. We declare them as separate encodings, called Lmt*, because they differ quite a bit from the OM* encodings, which might lead to trouble if, e.g., someone assumed the letters font has oldstyle digits or that the symbols font has calligraphic letters.

There will be a heavy series for some fonts, which we denote by the macro \hvdefault, and an upright italic shape, \uidefault. For the bold and heavy math versions we would need additional bold, ultra (\uldefault) and poster (\psdefault) weight fonts; pretend they exist, and then define substitutes. For the off chance that these are already defined, we “provide” them, rather than define them. Only the macros are used in the rest of this file, i.e., “{ui}” never appears again, so no font designations are hardwired into this file (except those for common Computer Modern fonts).

Letters   Declare the letters fonts. Include \noaccents so math accents don’t break in \mathnormal.

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>\DeclareFontEncoding{LmtL}{&lt;-&gt;mtl}{}</td>
</tr>
<tr>
<td>93</td>
<td>\DeclareFontFamily{LmtL}{mtl}{}</td>
</tr>
<tr>
<td>94</td>
<td>\DeclareFontShape{LmtL}{mtl}{&lt;-&gt;mtmi}{&lt;-&gt;mtmib}{}</td>
</tr>
<tr>
<td>95</td>
<td>\DeclareFontShape{LmtL}{mtl}{&lt;-&gt;mtmuh}{&lt;-&gt;mtmih}{}</td>
</tr>
</tbody>
</table>

Note that the font mtmih is a virtual font that uses ptmbi8r, so even if you have a “heavy” weight Times italic font, you can’t use it as a math letters font unless you know how to create a suitable raw font and change mtmih.vf and mtmih.vf. The same applies to mtmuh.

Declare the letters font with upright lower case Greek letters too; we won’t do anything with them, but we will describe how to set them up, if someone wants to use them as symbols (Sec. 6.12).

Symbols   Now the symbols font:

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>\DeclareFontEncoding{LmtS}{&lt;-&gt;mtl}{}</td>
</tr>
<tr>
<td>101</td>
<td>\DeclareFontFamily{LmtS}{mts}{}</td>
</tr>
</tbody>
</table>

The fonts have no skewkerno
Why do we say `\no@accents` when this is the one font that will have accents if we don’t find any elsewhere?—because they are in non-standard slots. There is no “\mathsy” alphabet, but someone could try to be clever and define one. Since we said `\noaccents@`, we might as well do the following re-declarations (\AMS does them too).

\begin{verbatim}
102 \DeclareFontEncoding{OML}{}{\noaccents@}
103 \DeclareFontEncoding{OMS}{}{\noaccents@}
\end{verbatim}

mtsyn and the older mtsy differ only in the radical1 glyph (C132, ‘207), which is a surd-radical at the baseline (as opposed to below it), otherwise identical to the other radical sign at ‘160; see Sec. 6.13.

\begin{verbatim}
104 \DeclareFontShape{LmtS}{mts}{\mddefault}{\updefault}{<->mtsyn}{}
105 \DeclareFontShape{LmtS}{mts}{\bfdefault}{\updefault}{<->mtsyb}{}
106 \DeclareFontShape{LmtS}{mts}{\hvdefault}{\updefault}{<->mtsyh}{}
\end{verbatim}

**Script Letters** The letters-only mtms* fonts, which we use for `\mathcal` and/or `\mathscr`.

\begin{verbatim}
107 \DeclareFontEncoding{LmtW}{}{\noaccents@}
108 \DeclareFontFamily{LmtW}{mtw}{<skewchar 42 \hyphenchar -1>}
109 \DeclareFontShape{LmtW}{mtw}{\mddefault}{\updefault}{<->mtms}{}
110 \DeclareFontShape{LmtW}{mtw}{\bfdefault}{\updefault}{<->mtmsb}{}
\end{verbatim}

**Math Extention** The largesymbols and math extension fonts mtex* are compatible with OMX; the only difference is an extra radical in slot ‘237, which we don’t use, because it is identical to the one in the symbols font (‘160).

\begin{verbatim}
111 \DeclareFontFamily{OMX}{mx}{<25>}
112 \DeclareFontShape{OMX}{mx}{\mddefault}{\updefault}{<->mtex}{}
113 \DeclareFontShape{OMX}{mx}{\bfdefault}{\updefault}{<->mtexb}{}
114 \DeclareFontShape{OMX}{mx}{\hvdefault}{\updefault}{<->mtexh}{}
\end{verbatim}

**Greek Text** MathTime includes a font with upright Greek letters, upper and lower case, meant for use as letters (as opposed to symbols). The fonts have ligatures with empty slots to get alternate forms—see Section 8 of the Math-TimePLUS reference manual.

\begin{verbatim}
115 \DeclareFontEncoding{LmtG}{}{\noaccents@}
116 \DeclareFontFamily{LmtG}{mtg}{}
117 \DeclareFontShape{LmtG}{mtg}{\mddefault}{\updefault}{<->mtgu}{}
118 \DeclareFontShape{LmtG}{mtg}{\bfdefault}{\updefault}{<->mtgub}{}
\end{verbatim}

Lastly, declare default for the new encodings, so NFSS is happy.

\begin{verbatim}
119 \DeclareFontSubstitution{LmtL}{mtl}{\mddefault}{\itdefault}
120 \DeclareFontSubstitution{LmtS}{mts}{\mddefault}{\updefault}
121 \DeclareFontSubstitution{LmtW}{mtw}{\mddefault}{\updefault}
122 \DeclareFontSubstitution{LmtG}{mtg}{\mddefault}{\updefault}
\end{verbatim}

### 6.6 Math declarations

We assume the usual math setup was loaded into the format and change things around to use MathTime. If for some reason you don’t read a standard math setup into the format, you’ll have to read the setup in before loading this package. Note that it is not necessary to “preload” the standard Computer Modern font metrics, if font memory is a problem.
Math Versions  We use three math versions, normal, bold, and heavy, and keep everything separate, as if there really were three sets of fonts available for everything. Actually, they’re not, but one could use the hooks to supply such fonts if one had them.

Give a notice so people don’t freak when they see all the warnings messages NFSS emits.

Some of the code that follows is redundant.

\PackageWarning{mt11p-Font}{Expect LOTS of NFSS warnings}

\DeclareMathVersion{normal}
\DeclareMathVersion{bold}
\DeclareMathVersion{heavy}
\def\normalmath{\@nomath\normalmath\mathversion{normal}}
\def\boldmath{\@nomath\boldmath\mathversion{bold}}
\def\heavymath{\@nomath\heavymath\mathversion{heavy}}
\def\unboldmath{\normalmath}
\let\unheavymath\unboldmath
\let\unnormalmath\relax

This is a private joke...

Declare Math Font Families  The following does not depend on any options:

\DeclareSymbolFont{operators}{\encodingdefault}{\rmdefault}{\mddefault}{\updefault}
\DeclareSymbolFont{letters}{LmtL}{\mddefault}{\itdefault}
\DeclareSymbolFont{symbols}{LmtS}{\mddefault}{\updefault}
\DeclareSymbolFontAlphabet \mathnormal {letters}
\DeclareSymbolFontAlphabet \mathrm {operators}
\DeclareMathAlphabet \mathbf {\encodingdefault}{\rmdefault}{\bfdefault}{\updefault}
\DeclareMathAlphabet \mathsf {\encodingdefault}{\sfdefault}{\mddefault}{\updefault}
\DeclareMathAlphabet \mathit {\encodingdefault}{\rmdefault}{\mddefault}{\itdefault}
\DeclareMathAlphabet \mathtt {\encodingdefault}{\ttdefault}{\mddefault}{\updefault}

The calligraphy stuff depends on what ‘cal’ option is being used; we either use mtms as the \mathcal font, or we make cmy the \mathcal font and make mtms a “script” font \mathscr, or we make cmy the \mathcal font and don’t use mtms for anything.

\ifcase\mt@cal % case 0
  \DeclareMathAlphabet \mathcal {LmtW}{mtw}{\mddefault}{\updefault}
  \or % case 1
  \DeclareMathAlphabet \mathsc {LmtW}{mtw}{\mddefault}{\updefault}
  \or % case 2
  \DeclareMathAlphabet \mathcal {OMS}{cmsy}{m}{n}
  \else % oops
  \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@cal}{
The \string\mathcal/\string\mathscr setup is hosed.
\fi

We’ll have to deal with more \mathcal-related stuff later.

Note that things defined via \DeclareSymbolFontAlphabet switch math versions automatically with \SetSymbolFont; these include \mathnormal and \mathrm.

Normal Math Version  This is largely redundant.
\SetSymbolFont{operators}{normal}{\encodingdefault}{\mddefault}{\updefault}
\SetSymbolFont{letters}{normal}{LmtL}{mlt}{\mddefault}{\itdefault}
\SetSymbolFont{symbols}{normal}{LmtS}{mts}{\mddefault}{\updefault}
\SetSymbolFont{largesymbols}{normal}{OMX}{mtx}{\mddefault}{\updefault}
\SetMathAlphabet \mathsf {normal}{\encodingdefault}{\sfdefault}{\mddefault}{\updefault}
\SetMathAlphabet \mathtt {normal}{\encodingdefault}{\ttdefault}{\mddefault}{\updefault}
\SetMathAlphabet \mathit {normal}{\encodingdefault}{\rmdefault}{\itdefault}
\SetMathAlphabet \mathbf {normal}{\encodingdefault}{\rmdefault}{\uldefault}{\updefault}
\mathcal and \mathscr have several cases.
\ifcase\mt@cal % case 0
\SetMathAlphabet \mathcal {normal}{LmtW}{mtw}{\mddefault}{\updefault}
or % case 1
\SetMathAlphabet \mathcal {normal}{OMS}{cmsy}{m}{n}
or % case 2
\SetMathAlphabet \mathcal {normal}{OMS}{cmsy}{m}{n}
\fi

Bold Math Version  Same story as above.
\SetSymbolFont{operators}{bold}{\encodingdefault}{\bfdefault}{\updefault}
\SetSymbolFont{letters}{bold}{LmtL}{mlt}{\bfdefault}{\itdefault}
\SetSymbolFont{symbols}{bold}{LmtS}{mts}{\bfdefault}{\updefault}
\SetSymbolFont{largesymbols}{bold}{OMX}{mtx}{\bfdefault}{\updefault}
\SetMathAlphabet \mathsf {bold}{\encodingdefault}{\sfdefault}{\bfdefault}{\updefault}
\SetMathAlphabet \mathtt {bold}{\encodingdefault}{\ttdefault}{\bfdefault}{\updefault}
\SetMathAlphabet \mathit {bold}{\encodingdefault}{\rmdefault}{\bfdefault}{\itdefault}
\SetMathAlphabet \mathbf {bold}{\encodingdefault}{\rmdefault}{\uldefault}{\updefault}
And the cases
\ifcase\mt@cal % case 0
  \SetMathAlphabet \mathcal {bold}{LmtW}{mtw}{\bfdefault}{\updefault} % case 0
\or % case 1
  \SetMathAlphabet \mathcal {bold}{OMS}{cmsy}{b}{n} % case 1
\fi

\SetSymbolFont{operators}{heavy}{\encodingdefault} % case 0
\SetSymbolFont{letters} {heavy}{LmtL}{mtl}{\hvdefault}{\itdefault} % case 0
\SetSymbolFont{symbols} {heavy}{LmtS}{mts}{\hvdefault}{\updefault} % case 0
\SetSymbolFont{largesymbols}{heavy}{OMX}{mtx}{\hvdefault}{\updefault} % case 0
\SetMathAlphabet \mathsf {heavy}{\encodingdefault}{\sfdefault}{\hvdefault}{\updefault} % case 0
\SetMathAlphabet \mathtt {heavy}{\encodingdefault}{\ttdefault}{\hvdefault}{\updefault} % case 0
\SetMathAlphabet \mathit {heavy}{\encodingdefault}{\rmdefault}{\hvdefault}{\itdefault} % case 0
\SetMathAlphabet \mathbf {heavy}{\encodingdefault}{\rmdefault}{\psdefault}{\updefault} % case 0

And the cases for script,
\ifcase\mt@cal % case 0
  \SetMathAlphabet \mathcal {bold}{LmtW}{mtw}{\hvdefault}{\updefault} % case 0
\or % case 1
  \SetMathAlphabet \mathcal {bold}{OMS}{cmsy}{\hvdefault}{\updefault} % case 1
\fi

\ifcase\mt@sub % case 0
  \PackageWarning{mt11p}{No math font substitutions set} % case 0
\or % case 1
  \def\mt@fd@check#1{\edef\reserved@a{\lowercase{
oexpand % case 1
\fi

\section{Bold \& heavy font substitutions}

We pretended there are all sorts of weights available, but, in fact, they’re not. Declare substitutions, unless we were told not to. NFSS will complain if we use an as-of-yet undeclared font as a substitute, so we explicitly read in the \texttt{.fd} files (this happens later for other fonts too). No harm will be done if they have already been read. Read in even \texttt{OMScmsy.fd}, in case the format has zilch preloaded.
\ifcase\mt@sub % case 0
  \PackageWarning{mt11p}{No math font substitutions set} % case 0
\or % case 1
  \def\mt@fd@check#1{\edef\reserved@a{\lowercase{ % case 1
\fi

Check for and read in the files \texttt{encodingdefault.\xydefault.fd} for the three text families. Be nice and check for the all-lower case name too.
\ifcase\mt@sub % case 0
  \PackageWarning{mt11p}{No math font substitutions set} % case 0
\or % case 1
  \def\mt@fd@check#1{\edef\reserved@a{\lowercase{ % case 1
\fi

\end{document}
6.8 Mathcodes

The MathTime fonts are nearly replacements for Computer Modern, but there are a few things to fix up. We also make it possible to take some symbols and accents from either operators or from elsewhere, depending on the options; this will be done later.

letters Font  Upright upper case Greek no longer comes from the operators font, but from the letters font, so upper and lower case Greek letters now behave identically, namely as math symbols. This means that something like $\mathit{\Delta\Gamma}\mathbf{\Xi}$ no longer works—upper case Greek letters are not affected by math alphabet changes. There is no simple fix.

Upright upper case Greek:
284 \DeclareMathSymbol\varGamma \{\mathord\} \{letters\} \{'000\}
285 \DeclareMathSymbol\varDelta \{\mathord\} \{letters\} \{'001\}
286 \DeclareMathSymbol\varTheta \{\mathord\} \{letters\} \{'002\}
287 \DeclareMathSymbol\varLambda \{\mathord\} \{letters\} \{'003\}
288 \DeclareMathSymbol\varXi \{\mathord\} \{letters\} \{'004\}
289 \DeclareMathSymbol\varPi \{\mathord\} \{letters\} \{'005\}
290 \DeclareMathSymbol\varSigma \{\mathord\} \{letters\} \{'006\}
291 \DeclareMathSymbol\varUpsilon \{\mathord\} \{letters\} \{'007\}
292 \DeclareMathSymbol\varPhi \{\mathord\} \{letters\} \{'010\}
293 \DeclareMathSymbol\varPsi \{\mathord\} \{letters\} \{'011\}
294 \DeclareMathSymbol\varOmega \{\mathord\} \{letters\} \{'012\}

Lower case Greek letters are OK, except for
295 \DeclareMathSymbol\varkappa \{\mathord\} \{letters\} \{'176\}
296 \DeclareMathSymbol\omicron \{\mathord\} \{letters\} \{'\o\}', for completeness

\textbf{symbols Font} There are a few new symbols, and the fonts no longer contain calligraphic letters. Easy things (those where no alternatives are available) first; the new names are those used in the MathTime distribution.
297 \DeclareMathSymbol\openclubsuit \{\mathord\} \{symbols\} \{'78\}
298 \DeclareMathSymbol\shadedclubsuit \{\mathord\} \{symbols\} \{'79\}
299 \DeclareMathSymbol\openspadesuit \{\mathord\} \{symbols\} \{'7A\}
300 \DeclareMathSymbol\shadedspadesuit \{\mathord\} \{symbols\} \{'7B\}
301 \DeclareMathSymbol\triangleleft \{\mathbin\} \{symbols\} \{'47\}
302 \DeclareMathSymbol\triangleright \{\mathbin\} \{symbols\} \{'46\}
303 \DeclareMathSymbol\cupprod \{\mathbin\} \{symbols\} \{'59\}
304 \DeclareMathSymbol\capprod \{\mathbin\} \{symbols\} \{'5A\}

The \texttt{\varcirc} symbol is slightly smaller than the ‘open bullet’. mathtimy calls it \texttt{\comp}.
305 \DeclareMathSymbol\varcirc \{\mathbin\} \{symbols\} \{'42\}
306 \let\comp=\varcirc

\texttt{\vec} used to be in \texttt{letters}, now it’s in \texttt{symbols}.
307 \DeclareMathAccent\vec \{\mathord\} \{symbols\} \{'45\}

\texttt{mtsny} has an accent glyph called \texttt{overbar}, which mathtimy calls \texttt{\widebar}.
308 \DeclareMathAccent\overbar \{\mathord\} \{symbols\} \{'53\}
309 \let\widebar=\overbar

The glyph called \texttt{\backslash} in the AFM file of \texttt{mtsny} is the one usually used for \texttt{\setminus}, and the format file has a definition for it (\texttt{symbols}, slot \texttt{"6E}). There is a \texttt{\setminus} in the AFM file, but it looks somewhat like a tilted minus sign. \texttt{\AMS} packages call that glyph \texttt{\smallsetminus}.
310 \DeclareMathSymbol\varsetminus \{\mathbin\} \{symbols\} \{'58\}
311 \let\smallsetminus=\varsetminus
Calligraphic letters and their variants  Irrespective of the options, both \texttt{\mathcal} and \texttt{\mathscr} get defined; warn only when the \texttt{nomentms} option is used.

\begin{verbatim}
312 \ifcase\mt@cal % case 0
313 \let\mathscr\mathcal % don't warn
314 \or % case 1
315 \or % case 2
316 \def\mathscr#1{\mathcal{#1}}%
317 \PackageWarning{mt11p}{no \string\mathscr\space available;
318 \MessageBreak using OMScmsy \string\mathcal}
319 \fi

We don't try to trap cases like \texttt{\mathcal{a}} when the \texttt{\mathcal} alphabet is OMS (and hence has no lowercase letters!). Presumably by writing \texttt{\mathcal{a}} you know that \texttt{\mathcal} is \texttt{LmtW}, i.e., the \texttt{mtcal} option is in effect. Writing \texttt{\mathscr{a}} when \texttt{\mathscr} is essentially a synonym for \texttt{\mathcal} will at least produce a warning.

\texttt{mtms*} has alternate forms of a few letters. We define math commands to fetch them, assuming we have an \texttt{mtms*} font as \texttt{\mathscr}. (The script letters have sidebearings which make them unsuitable for text; use math mode for logos and such.) The auxiliary macro will be defined later.

\begin{verbatim}
320 \def\varA {\mt@al@hexsy\mathscr{30}}
321 \def\varE {\mt@al@hexsy\mathscr{31}}
322 \def\varG {\mt@al@hexsy\mathscr{32}}
323 \def\varvarG {\mt@al@hexsy\mathscr{33}}
324 \def\varI {\mt@al@hexsy\mathscr{34}}
325 \def\varL {\mt@al@hexsy\mathscr{35}}
326 \def\varQ {\mt@al@hexsy\mathscr{36}}
327 \def\varS {\mt@al@hexsy\mathscr{37}}
328 \def\varr {\mt@al@hexsy\mathscr{38}}
329 \def\varz {\mt@al@hexsy\mathscr{39}}
330 \def\iscript {\mt@al@hexsy\mathscr{7B}}% can't use \imath
331 \def\jscript {\mt@al@hexsy\mathscr{7C}}% can't use \jmath

The MathTime distribution defines the macro \texttt{\script}, which is supposed to produce the script letter “A” and its alternate form with “\texttt{\script A}” and “\texttt{\script^*A}”, respectively. The argument of \texttt{\script} had better be a single token (or an asterisk and a single token), else we get a mess; in order to get the expected output that token should be a single letter\textsubscript{11}, one of those that have variant forms.

\begin{verbatim}
332 \DeclareRobustCommand\script{\@ifstar\@varscript\@script}
333 \ifnum\mt@cal<2 % mtms available
334 \def\@script#1{\mathscr{#1}}
335 \def\@varscript#1{%
336 \ifx#1A\varA
337 \else\ifx#1E\varE
338 \else\ifx#1G\varG
339 \else\ifx#1I\varI
340 \else\ifx#1L\varL
341 \else\ifx#1Q\varQ
342 \else\ifx#1S\varS
343 \else\ifx#1r\varr
344 \else\ifx#1z\varz
345 \endverbatim

23
Symbols from either operators or from elsewhere Many symbols and punctuation marks can be taken from either the operators math family, or from letters/symbols. Most of the glyphs are virtually identical in either font. The ‘optimal’ choice depends on several things, e.g., do you wish to use as few glyphs from the MathTime fonts as possible, do you prefer ‘thinner’ symbols, etc. The opmax option will take as many things from operators as possible, but risk, e.g., that the = sign will not line up with double arrows, etc. The opsafe option uses operators glyphs only when absolutely nothing can go wrong; the opnone option avoids operators whenever possible and uses MathTime glyphs instead.

The operators font is uses the encoding \encodingdefault. We must recognize that encoding in order to take any symbols from operators. The only encodings I know of that someone might possibly be using with LATEX2ε are T1, OT1 or a superset thereof, 8r and 8a. It’s easy to hack in another encoding, if required, as would be the case with a superset or OT1. If we don’t recognize the encoding, stick to using symbols and letters.

NFSS has a “\DeclareMathDelimiter” but nothing to assign \delcodes. We try to make a general macro and insist on saying “\symletters” rather than just saying “1”, eventhough it’s pretty much out of the question that \symoperators, \symletters, \symsymbols and \symlargeoperators are something other than 0, 1, 2 and 3, respectively, since \TeX itself checks for extra fontdimens of \fam2 and \fam3 before it agrees to typeset math. (The second and fourth arguments should be hexadecimal numbers without the “\”.)

This code is read once and not stored, so we just write the whole thing out case by case. \colon is defined in some of the AMS packages, so don’t overwrite an existing definition.

Here’s the overly conservative option:

```
\ifcase\mt@ops % case 0
  \DeclareMathSymbol{0}{\mathopen}{letters}{"2E}
  \mt@assign@del{\{letters\}}{\{largesymbols\}}{00}
  \DeclareMathSymbol{0}{\mathclose}{letters}{"2F}
  \mt@assign@del{\{letters\}}{\{largesymbols\}}{01}
  \DeclareMathSymbol{+}{\mathbin}{symbols}{"43}
```
\DeclareMathSymbol{=} \mathrel \{symbols\} \{\texttt{\char44}\} % identical one at \texttt{\char48}
\DeclareMathSymbol{;} \mathpunct \{symbols\} \{\texttt{\char49}\}
\DeclareMathSymbol{[} \mathopen \{symbols\} \{\texttt{\char54}\}
\mt@assign@del{'}\texttt{[}\{symbols\} \{\texttt{\char54}\} \{largesymbols\} \{\texttt{\char02}\}
\DeclareMathSymbol{]} \mathclose \{symbols\} \{\texttt{\char55}\}
\mt@assign@del{'}\texttt{]} \{symbols\} \{\texttt{\char55}\} \{largesymbols\} \{\texttt{\char03}\}
\DeclareMathSymbol{:} \mathrel \{symbols\} \{\texttt{\char56}\}
\ifx\colon\undefined % AMS defines it differently; don't overwrite that
\DeclareMathSymbol{\colon} \mathpunct \{symbols\} \{\texttt{\char56}\}
\fi
\DeclareMathSymbol{!} \mathclose \{symbols\} \{\texttt{\char57}\}
\DeclareMathSymbol{.} \mathord \{letters\} \{\texttt{\char3A}\}
\DeclareMathSymbol{\period} \mathpunct \{letters\} \{\texttt{\char3A}\}
\DeclareMathSymbol{,} \mathpunct \{letters\} \{\texttt{\char3B}\}
% \DeclareMathSymbol{<} \mathrel \{letters\} \{\texttt{\char3C}\} % format defaults
% \DeclareMathSymbol{>} \mathrel \{letters\} \{\texttt{\char3E}\}
% \DeclareMathSymbol{|} \mathord \{symbols\} \{\texttt{\char6A}\}
% \DeclareMathSymbol{/} \mathord \{letters\} \{\texttt{\char3D}\}
% \DeclareMathSymbol{-} \mathbin \{symbols\} \{\texttt{\char00}\}

Now the safe version; the stuff we take from operators never gets used in “synthe-
sized” or extensible symbols.
\or % case 1
\DeclareMathSymbol{=} \mathrel \{operators\} \{\texttt{\char3D}\}
\mt@assign@del{'}\texttt{=} \{operators\} \{\texttt{\char28}\} \{largesymbols\} \{\texttt{\char00}\}
\DeclareMathSymbol{[} \mathopen \{operators\} \{\texttt{\char5B}\}
\mt@assign@del{'}\texttt{[} \{operators\} \{\texttt{\char5B}\} \{largesymbols\} \{\texttt{\char02}\}
\DeclareMathSymbol{]} \mathclose \{operators\} \{\texttt{\char5D}\}
\mt@assign@del{'}\texttt{]} \{operators\} \{\texttt{\char5D}\} \{largesymbols\} \{\texttt{\char03}\}
\DeclareMathSymbol{:} \mathrel \{operators\} \{\texttt{\char3A}\}
\ifx\colon\undefined % AMS defines it differently; don't overwrite that
\DeclareMathSymbol{\colon} \mathpunct \{operators\} \{\texttt{\char3A}\}
\fi
\DeclareMathSymbol{!} \mathclose \{operators\} \{\texttt{\char3B}\}
\DeclareMathSymbol{.} \mathord \{operators\} \{\texttt{\char3C}\}
\DeclareMathSymbol{\period} \mathpunct \{operators\} \{\texttt{\char3C}\}
\DeclareMathSymbol{,} \mathpunct \{operators\} \{\texttt{\char3D}\}
% \DeclareMathSymbol{<} \mathrel \{letters\} \{\texttt{\char3F}\} % format defaults
% \DeclareMathSymbol{>} \mathrel \{letters\} \{\texttt{\char40}\}
% \DeclareMathSymbol{|} \mathord \{symbols\} \{\texttt{\char44}\}
% \DeclareMathSymbol{/} \mathord \{letters\} \{\texttt{\char46}\}
% \DeclareMathSymbol{-} \mathbin \{symbols\} \{\texttt{\char47}\}

Now \texttt{opmax}. This is risky, or more exactly some glyphs will be flawed, because the
minus and equal signs won’t line up with arrows, etc. On the other hand, this will
minimize the use of non-resident fonts, so it may make the final output marginally
smaller if you have partial font downloading or if you include the MathTime fonts
as bitmaps.
\or % case 2
\PackageWarning{mt11p}{opmax enabled}
Now we have to be careful: <, > and | can be taken from \texttt{operators} if \texttt{encodingdefault} is anything \textit{except} OT1. We use an \texttt{endash} instead of a hyphen for the minus sign if we recognize the encoding to get at least something close to a real minus sign. That’s a sloppy kludge, but choosing the \texttt{opmax} option implies you’ve lowered your standards somewhat.

\begin{verbatim}
\def\reserved@a{OT1}\edef\reserved@b{\encodingdefault}
\ifx\reserved@a\reserved@b % unavailable in OT1...
  \DeclareMathSymbol{<}{\mathrel}{letters}{"3C}% format defaults
  \DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
  \DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
  \DeclareMathSymbol{-}{\mathbin}{operators}{"7B}% dangerous
\else
  \def\reserved@b{T1}%
  \ifx\reserved@b\reserved@b
    \DeclareMathSymbol{-}{\mathbin}{operators}{150}% dangerous
  \else
    \fi % 8r
  \fi % T1
  \DeclareMathSymbol{<}{\mathrel}{operators}{'\<}% ascii positions
  \DeclareMathSymbol{>}{\mathrel}{operators}{'\>}
  \DeclareMathSymbol{|}{\mathord}{operators}{'\|}
  \mt@assign@del{'\|}{operators}{7C}% ascii bar
\end{verbatim}
Math Accents  Accents in math can be made to either use a fixed font, or to obey math alphabets. In order to have accents of the latter type, we must recognize the \encodingdefault (which the operators math family is in) and set the corresponding accent codes up for that encoding. If we don’t recognize the encoding we have to use fixed accents; MathTime provides suitable accents in the symbols font.

For some reason the ring accent “å” is always left out, so we define it, but don’t warn if we’re stuck with symbols accents which have no ring accent glyph and just make \ring a no-op.

The code below is written out long hand style, to keep it legible. It does not get stored, it’s read only once, and most of it is skipped, so it would be only a marginal improvement to merge everything.

The opaccents case:

\ifcase\mt@accs % case 0

We do something similar to what AMS does, but rather simple-mindedly: choose between ‘fixed’ (class 0) or ‘variable’ (class 7) based on the value of \mt@acc@class, whose value should be either 0 or 7, depending on whether the (encoding of the) current math alphabet has accents in the expected positions. We could use the AMS hook \accentclass@, but we ought to make our own… Actually, this is slightly naughty: we could have gone through the \DeclareMathAccent command, declare two accents, and have the user-level accent command pick the right one based on something \noaccents@ says, but that’s kind of pointless.

All of this works as long as we recognize the encoding. If we don’t, use fixed accents from symbols and complain. The slot specification must be in hex.
If we don’t recognize the encoding, we have to resort to symbols:

\def\reserved@b{OT1}\textquoteleft‘Knuth roman’\textquoteright
\else
\def\reserved@b{8r}% TeXBase1
\fi\reserved@b
\def\reserved@b{8a}% AdobeStandard
\else
% I give up...
\else % I give up...
\DeclareMathAccent\grave\mathord{\symbols}{"4A}
\DeclareMathAccent\acute\mathord{\symbols}{"4B}
\DeclareMathAccent\check\mathord{\symbols}{"4C}
\DeclareMathAccent\breve\mathord{\symbols}{"4D}
\DeclareMathAccent\bar\mathord{\symbols}{"4E}
\DeclareMathAccent\hat\mathord{\symbols}{"4F}
\DeclareMathAccent\dot\mathord{\symbols}{"50}
\DeclareMathAccent\ddot\mathord{\symbols}{"52}
\def\ring#1{{#1}} % no \ring in the symbols font
\PackageError{mt11p}\
{Unknown \string\encodingdefault\space}
\encodingdefault. \MessageBreak
Reverting to fixed accents
from symbols font}%
{I had to make math accents ignore alphabet changes, \MessageBreak
because I don’t recognize the \‘\encodingdefault’ encoding.}\
\fi % 8a
\fi % Sr
\fi % OT1
\fi % T1

Add the switch to \noaccents@ (and check that that’s defined before hand) and initialize the flag.

\ifx\noaccents@\@undefined\let\noaccents@\@empty\fi % be safe
\g@addto@macro{\noaccents@}{\def\mt@acc@class{0}}
\def\mt@acc@class{7}

The symaccents case is easy; just make everything fixed, and make sure \noaccents@ is defined.

\or % case 1, accents from symbols
\DeclareMathAccent\grave\{mathord\}{symbols}\{"4A\}
\DeclareMathAccent\acute\{mathord\}{symbols}\{"4B\}
\DeclareMathAccent\check\{mathord\}{symbols}\{"4C\}
\DeclareMathAccent\breve\{mathord\}{symbols}\{"4D\}
\DeclareMathAccent\bar\{mathord\}{symbols}\{"4E\}
\DeclareMathAccent\hat\{mathord\}{symbols}\{"4F\}
\DeclareMathAccent\dot\{mathord\}{symbols}\{"50\}
\DeclareMathAccent\tilde\{mathord\}{symbols}\{"51\}
\DeclareMathAccent\ddot\{mathord\}{symbols}\{"52\}
\def\ring#1{{#1}} % no \ring in the symbols font
\ifx\noaccents@\@undefined\let\noaccents@\@empty\fi % be safe
\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@accs}%
{The math accent setup is hosed; expect bad output.}
\fi

Now another messy part: the A\LaTeX\ package amsmath has the macros \Hat, etc., defined with hard-coded accent positions. We redo the definition to work with whatever encoding or accents we’re using. The code below is basically copied from the A\LaTeX\ package. We omit this patch unless all of the following conditions are satisfied: i) the macro \mathaccent@ exists; ii) the macros \Hat and \Check exist; iii) there exists a macro \accentclass@ whose definition is “7”. If all of that is OK we pretty likely have the amsmath package and the following should be safe. Note that \mathaccent@ is a rather large macro that figures out what skew kern is needed to get double accents to line up.

\ifx\mathaccent@\undefined\else % (i)
\ifx\Hat\undefined\else % (iia)
\ifx\Check\undefined\else % (iib)
\def\@tempa{7}\
\ifx\@tempa\accentclass@ % (iii)
\PackageInfo{mt11p}{amsmath detected; redefining double accents}
\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@accs}%
{The math accent setup is hosed; expect bad output.}
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
\fi
Define fancy versions of \hat, etc., called "\Hat", etc., that perform skew calculations. To get a double accent say something like \Hat{\Hat{x}}.

There are two versions of the definition macro: one for fixed accents, another for variable ones.

These two accents are from the symbols fonts:

Go through the encodings we know and define the accents again... The symaccents option affects this too.
If we don’t know the encoding use fixed accents from symbols. We already gave an error message before, so just warn.

\PackageWarning{mt11p}{using fixed accents for amsmath}
Upright Greek Text  The mtgu* fonts are not meant to be used as symbols, so we provide only text-type support: text mode font changing commands, and a math alphabet. Note that this means you can use the fonts in math with correct subscript sizing, etc.: just say something like \mathgr{m} for an upright \(\mu\) (Computer Modern does not have such a glyph); in text \textgr{m} will work. Note also that the ligatures involving nonprinting letters for getting variant forms will work inside of \mathgr, but o*, O* or e* will not, which is why we added some ligatures to the tfm file (Sec. 3.9). The text commands are non-fragile, because the components of the macros are robust. See Sec. 6.12 for making upright Greek letters into math symbols. Trap the nomathgr case.

\begin{verbatim}
671 \def\greekshape\fontencoding{LmtG}\selectfont \% 672 \DeclareTextFontCommand{\textgr}{\greekshape} 673 \ifcase\mt@mgr \% case 0 674  \def\mathgr#1{\PackageError{mt11p}{upright Greek not loaded for math}\% 675  \{Don't use the nomathgr option.\}% 676  \hbox{\greekshape #1}}% as good as anything else 677 \or \% case 1 678  \DeclareMathAlphabet\mathgr{LmtG}{mtg}{\mddefault}{\updefault} 679  \SetMathAlphabet\mathgr{normal}{LmtG}{mtg}{\mddefault}{\updefault} 680  \SetMathAlphabet\mathgr{bold} {LmtG}{mtg}{\bfdefault}{\updefault} 681  \SetMathAlphabet\mathgr{heavy} {LmtG}{mtg}{\hvdefault}{\updefault} 682 \else \% oops 683  \PackageError{mt11p}{PANIC: Unexpected value of \string\mt@mgr}\% 684  \{The \string\mathgr\space setup is hosed.\} 685 \fi 686 \ifnum\mt@sub=1 \% 687  \DeclareFontShape{LmtG}{mtg}{\hvdefault}{\updefault} 688  \{<->ssub*mtg/{\bfdefault}/\updefault}\% heavy mtgu SUBS 689 \fi
\end{verbatim}

The MathTıme distribution defines the \gk font switch macro, roughly equivalent to \greekshape; we don't.

\texttt{\hbar} needs to be defined a bit carefully. The plain\TeX{} definition is just a macron (bar) accent with an ‘h’ backed up into it: \texttt{\char‘26\mkern-9muh}. This is pretty horrible, and doesn’t work in Times, no matter what \texttt{\m kern} you use. The definition below actually looks rather decent in all math versions. We define a private bar accent to be taken from the \texttt{symbols} math family, since we don’t know what the \texttt{\bar} accent is (it might follow math alphabets changes). Force the ‘h’ to be taken from \texttt{letters}.

\begin{verbatim}
690 \ifnum\mt@hbar=1 \% 691  \DeclareMathSymbol\mt@hbar@macron \{mathord\}{symbols}{"4E}\% 692 \DeclareMathSymbol\mt@hbar@h \{mathord\}{letters}{‘h}\% 693  \def\hbar@macron#1{\hbox to\z@{\m@th\raisebox{-0.16\height} 694  {$\csname#1style\endcsname\mkern.65mu\mt@hbar@macron$}\hss}} 695  \def\hbar{\mathord{\mathchoice{\hbar@macron{display}}{\hbar@macron{text}}{\hbar@macron{script}}{\hbar@macron{scriptscript}}} 696  \mt@hbar@h}}
701 \fi
\end{verbatim}
First a comment: the only reason we are dealing with these symbols at all is that they historically came from math (OMS) fonts; it would be easier to use a text symbol font, but such a font is not yet commonly in use, so we hack around with 8r. If you do have such a setup, the nodag option will skip all of the stuff below and not touch these four symbols, but check that the setup you use does not assume \fam2 is OMS.

The default is to fetch the symbols from 8r. We do not use anything else from 8r, in particular we don’t touch \textregistered, \texttrademark, etc. The 8r.sty “emulation” included herein only deals with the four symbols.

The format files defines these to use text or math versions of the character commands by using definitions similar to

\def\P{\ifmmode\mathparagraph\else\textparagraph\fi}

(The actual definition is made robust.) The text versions have declared defaults taken from OMS; the math versions were \textchars pointing to the (OMS) symbols math family. We do the text versions first, since the handling of the math versions depends on them.

We take the text symbols from 8r fonts, unless we’re told not to (or unless we’re skipping the whole thing). The format file makes the default test encoding for these four symbols OMS, so we don’t have to do anything. 8r fonts.

We have to be careful here, because the use of 8r fonts is “non-standard”. So check if we have an 8r.fd file for \rmdefault. If so, check for 8r.sty, read that, and (re)declare the four text commands we need. If there’s no 8r.sty, supply just enough to get to the four symbols we need. We assume the TFM files do exist (they have to, because MathTime virtual fonts use them), but we try loading the tfm file just to make sure.

\ifcase\mt@eightR % case 0
\PackageInfo{mt11p}{Retaining CM for \string\textdagger, etc}%
\or % case 1
\IfFileExists{8r\rmdefault.fd}{% 
{\PackageInfo{mt11p}{MT math using 8r.sty}%
\RequirePackage{8r}%
\DeclareTextSymbol{\textsection}{8r}{167}% section
\DeclareTextSymbol{\textparagraph}{8r}{182}% paragraph
\DeclareTextSymbol{\textdagger}{8r}{134}% dagger
\DeclareTextSymbol{\textdaggerdbl}{8r}{135}% daggerdbl
\DeclareTextSymbolDefault{\textsection}{8r}%
\DeclareTextSymbolDefault{\textparagraph}{8r}%
\DeclareTextSymbolDefault{\textdagger}{8r}%
\DeclareTextSymbolDefault{\textdaggerdbl}{8r}%
}% using 8r.sty
{\PackageWarning{mt11p}{% 
{No file 8r.sty found. Winging it}%
\DeclareFontEncoding{8r}{{no@accents}}%
\DeclareFontSubstitution{8r}{\rmdefault}%;
\else
\fi

3Lines not proceeded by a line number are a part of this description and do not wind up in the sty file!
If there's no .fd file for \texttt{rmdefault} we have to wing it. Just declare the 8r fonts for \texttt{rmdefault} text family. (That's all you would have with Computer Modern fonts anyway, and since we're here, no elaborate setup exists on the system.) Hopefully the TFM files are available and they have the expected names. This will work when someone has the 8r-based fonts installed, but no support for using the raw fonts. To catch the fatal case right now, we just load the 8r font by hand and put a somewhat illuminating comment on the same line in case \TeX\ complains. Perhaps we're skipping this altogether.

or \% case 2

\PackageInfo{mt11p}{not doing anything with \string\textdagger, etc.}

\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@eightR}{\string\dag, \string\ddag, \string\S\space and \string\P\space are probably hosed.}
\fi
If we’re told not to use \(8r\) fonts (\(\mathtt{eightR} = 0\)) and if \(\mathtt{cal} = 0\), we have no choice but to get OMS as a ‘private’ math alphabet and fetch the symbols from there. The auxiliary macro \(\mathtt{alhexsy}\) gets defined later.

\begin{verbatim}
\ifcase\mathtt{eightR} \ifnum\mathtt{cal}=0 \else \else\fi
\else % \mathtt{cal}>0
\fi
\end{verbatim}

(Substitutions have already been declared.) If \(\mathtt{cal} > 0\) we just use \(\mathcal\), which will be OMS.

\begin{verbatim}
\DeclareMathAlphabet\mathtt{sy}{OMS}{cmsy}{m}{n}
\SetMathAlphabet\mathtt{sy}{normal}{OMS}{cmsy}{m}{n}
\SetMathAlphabet\mathtt{sy}{bold} {OMS}{cmsy}{b}{n}
\SetMathAlphabet\mathtt{sy}{heavy} {OMS}{cmsy}{\hvdefault}{\updefault}
\def\mathsection \{\mathtt{alhexsy}\mathtt{sy}\{78\}}
\def\dagger \{\mathtt{alhexsy}\mathtt{sy}\{79\}}
\def\ddagger \{\mathtt{alhexsy}\mathtt{sy}\{7A\}}
\def\mathparagraph\{\mathtt{alhexsy}\mathtt{sy}\{7B\}}
\end{verbatim}

If it’s OK to use the \(8r\) fonts, make them a “private” math alphabet. Here we do need substitution declarations. The \(\text{.fd}\) file has already been read (or the data supplied).

\begin{verbatim}
\DeclareMathAlphabet\mathtt{sy}{\rmdefault}{\mddefault}{\updefault}
\SetMathAlphabet\mathtt{sy}{normal}{\rmdefault}{\mddefault}{\updefault}
\SetMathAlphabet\mathtt{sy}{bold} {\rmdefault}{\bfdefault}{\updefault}
\SetMathAlphabet\mathtt{sy}{heavy} {\rmdefault}{\hvdefault}{\updefault}
\def\mathparagraph\{\mathtt{alhexsy}\mathtt{sy}\{B6\}}
\def\dagger \{\mathtt{alhexsy}\mathtt{sy}\{86\}}
\def\ddagger \{\mathtt{alhexsy}\mathtt{sy}\{87\}}
\def\mathsection \{\mathtt{alhexsy}\mathtt{sy}\{A7\}}
\end{verbatim}

\begin{verbatim}
\DeclareFontShape{8r}{\rmdefault}{\hvdefault}{\updefault}% heavy \rm in 8r SUBS
{<->ssub*\rmdefault/\bfdefault/\updefault}{}
\end{verbatim}

Or skip it...

\begin{verbatim}
\PackageInfo{mt11p}{not doing anything with \string\dagger, etc.}
\end{verbatim}

Actually, the code above is suboptimal in the sense that if \(\text{encodingdefault}\) happens to be \(8r\), we waste a math family. Hopefully, no heavy math user is running \(8r\)...
6.9 Auxiliary macro

In a few instances we need a macro to get a specific character from a particular math alphabet, i.e., something like \texttt{\textbackslash matalph\{\textbackslash char"yy\}}. You can’t do that though, you would get \texttt{\textbackslash mathchar"00yy}. The \TeX{} math class 7 and \texttt{\fam} are used to get letters and digits from math alphabets, so we can just say \texttt{\matalph\{\mathchar"70yy\}}. The macro below is shorthand for the operation “set character (slot) from the font of the math alphabet \texttt{matalph}”.

\begin{verbatim}
\def\mt@al@hexsy#1#2{#1{\mathchar"70#2}}
\end{verbatim}

where \texttt{#1} is the math alphabet macro (e.g., \texttt{\mathcal}) and \texttt{#2} is the hexadecimal number of the slot (no double quote).

6.10 Assorted Fixes, Odds & Ends

Here are a few assorted (sordid;-)) fixes.

**Active Underscore** Make the underscore character active so we can kern to \texttt{j}, \texttt{f} and \texttt{p} in subscripts. This is rather messy and probably breaks on occasion. The code was basically copied from the MathTime distribution.

\begin{verbatim}
\ifcase\mt@sb % case 0
\PackageInfo{mt11p}{not using active underscore}
\or % case 1
\ifnum\the\catcode'_=8 %
\PackageInfo{mt11p}{redefining the underscore \string\catcode}
\PackageInfo{mt11p}{\string\catcode\space of underscore made 12 (other)}
\let\mt@orig@sb_
\catcode'=_=\active
\def_\ifmmode\expandafter\sb@\else\expandafter\mt@orig@sb\fi
\let\sb_
\def\sb@#1\mt@orig@sb{\futurelet\next\sb@@#1}
\def\sb@@{\ifx\next\space@\def\next@. {uturelet\next\sb@@}
\else\def\next@{.\ifx\next j\jadjust
\else\ifx\next f\jadjust
\else\ifx\next p\mkern-\@ne mu\fi\fi\fi}\
\fi\fi\fi}
\atmakeother_\else % \the\catcode\_<"8000 !
\PackageInfo{mt11p}{underscore did not have \string\mathcode "8000;}
\MessageBreak Underscore not made active}
\fi\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sb}\
\fi\else % \the\catcode\_<>8 !
\PackageWarning{mt11p}{underscore was not of category 8;\MessageBreak Underscore not made active}
\fi\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sb}\
\fi\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sb}\
\fi\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@sb}\
\fi\end{verbatim}
\textcircled{} The large-ish circle for constructing ©, ®, etc., is usually taken from OMS. The \texttt{LmTS} fonts have the same glyph in the same slot, so just redefine the default to point to \texttt{LmTS}. A text symbol font might have ©, etc., available as single characters, e.g., the raw \texttt{8r} fonts do, but it’s not our business to deal with all that.

\textcircled{} The large-ish circle for constructing ©, ®, etc., is usually taken from OMS. The \texttt{LmTS} fonts have the same glyph in the same slot, so just redefine the default to point to \texttt{LmTS}. A text symbol font might have ©, etc., available as single characters, e.g., the raw \texttt{8r} fonts do, but it’s not our business to deal with all that.

\textcircled{} The large-ish circle for constructing ©, ®, etc., is usually taken from OMS. The \texttt{LmTS} fonts have the same glyph in the same slot, so just redefine the default to point to \texttt{LmTS}. A text symbol font might have ©, etc., available as single characters, e.g., the raw \texttt{8r} fonts do, but it’s not our business to deal with all that.

\textcircled{} The large-ish circle for constructing ©, ®, etc., is usually taken from OMS. The \texttt{LmTS} fonts have the same glyph in the same slot, so just redefine the default to point to \texttt{LmTS}. A text symbol font might have ©, etc., available as single characters, e.g., the raw \texttt{8r} fonts do, but it’s not our business to deal with all that.

\textcircled{} The large-ish circle for constructing ©, ®, etc., is usually taken from OMS. The \texttt{LmTS} fonts have the same glyph in the same slot, so just redefine the default to point to \texttt{LmTS}. A text symbol font might have ©, etc., available as single characters, e.g., the raw \texttt{8r} fonts do, but it’s not our business to deal with all that.
\encodingdefault is OT1 and point the defaults to MathTime, so we don’t needlessly resort to Computer Modern. Noone uses these definitions directly, but macros might.

\DeclareTextSymbol\textless{LmtL}{60}% afm names
\DeclareTextSymbol\textgreater{LmtL}{62}
\DeclareTextSymbol\textperiodcentered{LmtS}{1}
\DeclareTextSymbol\textasteriskmath{LmtS}{3}% centered
\DeclareTextSymbol\textopenbullet{LmtS}{14}
\DeclareTextSymbol\textbullet{LmtS}{15}
\DeclareTextSymbol\textbraceleft{LmtS}{102}
\DeclareTextSymbol\textbraceright{LmtS}{103}
\DeclareTextSymbol\textbar{LmtS}{106}
\DeclareTextSymbol\textbackslash{LmtS}{110}

Make these the default:
\DeclareTextSymbolDefault\textperiodcentered{LmtS}
\DeclareTextSymbolDefault\textasteriskmath{LmtS}
\DeclareTextSymbolDefault\textopenbullet{LmtS}
\DeclareTextSymbolDefault\textbullet{LmtS}
\DeclareTextSymbolDefault\textbraceleft{LmtS}
\DeclareTextSymbolDefault\textbraceright{LmtS}
\DeclareTextSymbolDefault\textbar{LmtS}
\DeclareTextSymbolDefault\textbackslash{LmtS}
\DeclareTextSymbolDefault\textless{LmtL}
\DeclareTextSymbolDefault\textgreater{LmtL}

Oldstyle Digits (See Sec. 3.10). MathTime has no oldstyle digits, so we grab the ones from OML. I’m not sure it’s useful to have this command in math mode, but we might as well do that too, though not via a math family—we define a private \math@OS alphabet instead. Read the .fd file, in case the format did no preloading. nomathOS disables the math family.

\ifcase\mt@osdg % case 0
\def\math@OS#1{%PackageError{mt11p}{oldstyle digits not loaded for math}%}{Don't use the nomathOS option.}%{\hbox{\oldstylenums{#1}}}% as good as anything else
\or % case 1
\DeclareMathAlphabet\math@OS {OML}{cmm}{m}{it}
\SetMathAlphabet\math@OS {normal}{OML}{cmm}{m}{it}
\SetMathAlphabet\math@OS {bold} {OML}{cmm}{b}{it}
\SetMathAlphabet\math@OS {heavy} {OML}{cmm}{b}{it}
\else % oops
\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@osdg}{}{Oldstyle digits in math are hosed.}%
\fi
\ifnum\mt@sub=1 %
\InputIfFileExists{omlcmm.fd}{\relax}{\InputIfFileExists{OMLcmm.fd}{\relax}{\PackageError{mt11p}{No OMLcmm.fd file}{Your are missing a basic distribution file. %}{Check the LaTeX installation.}}{\PackageError{mt11p}{PANIC: Unexpected value of \string\mt@osdg}{Oldstyle digits in math are hosed.}}}
\fi
\InputIfFileExists{omlcmm.fd}{\relax}{grrr...}{\PackageError{mt11p}{No OMLcmm.fd file}{Your are missing a basic distribution file. %}{Check the LaTeX installation.}}}
\DeclareFontShape{OML}{cmm}{b}{it}{<->ssub*cmm/b/it}{}% heavy cmmi SUBS
\ff
The \texttt{\oldstylenums} macro just picks between using the OML font as a text font, or the \texttt{\math@OS} math alphabet. If the argument contains things other than digits, the result will be amusing...

\begin{verbatim}
def\oldstylenums#1{\ifmmode\math@OS{#1}\else\textfont{OML}{cmm}{\f@series}{it}#1\fi}
\end{verbatim}

\section{\boldsymbol and \heavysymbol}

As mentioned above, we assumed that formulæ do not require simultaneous access to fonts in different math versions, or, more precisely, that a single math list need not contain several weights of, e.g., the symbols fonts. In other words, that it is not worth while to simultaneously have several weights of, e.g., the symbols fonts available as “\texttt{\mathchar}”. Instead we fetch the \texttt{AMS} package \texttt{amsbsy}, which defines a “\texttt{\boldsymbol}” macro and define a similar “\texttt{\heavysymbol}” macro. Don’t do it if we’re told not to, or if we can’t find \texttt{amsbsy.sty} and don’t overwrite existing definitions.

\begin{verbatim}
\ifcase\mt@bsy % case 0
  \providecommand{\boldsymbol}[1]{\PackageError{mt11p}{Get or load the package amsbsy to use \string\boldsymbol}{See above.}#1\relax}
  \providecommand{\heavysymbol}[1]{\PackageError{mt11p}{Get or load the package amsbsy to use \string\heavysymbol}{See above.}#1\relax}
  \providecommand{\normalsymbol}[1]{\PackageError{mt11p}{Get or load the package amsbsy to use \string\normalsymbol}{See above.}#1\relax}
\or % case 1
  \IfFileExists{amsbsy.sty}{\RequirePackage{amsbsy}
    \DeclareRobustCommand{\heavysymbol}[1]{\begingroup
      \let\@nomath\@gobble \mathversion{heavy}\math@atom{##1}{
        \mathchoice
        \{\hbox{$\m@th\displaystyle##1$}\}
        \{\hbox{$\m@th\textstyle##1$}\}
        \{\hbox{$\m@th\scriptstyle##1$}\}
        \{\hbox{$\m@th\scriptscriptstyle##1$}\}}\endgroup}
    \DeclareRobustCommand{\normalsymbol}[1]{\begingroup
      \let\@nomath\@gobble \mathversion{normal}\math@atom{##1}{
        \mathchoice
        \{\hbox{$\m@th\displaystyle##1$}\}
        \{\hbox{$\m@th\textstyle##1$}\}
        \{\hbox{$\m@th\scriptstyle##1$}\}
        \{\hbox{$\m@th\scriptscriptstyle##1$}\}}\endgroup}}}{
\end{verbatim}

39
6.12 Examples of new math symbol definitions

Note that code in this section, and some of the code in the following sections is not included in the package .sty file. It’s meant as an example of things that could be put into a mt11p.cfg file.

Here follow a few examples of defining more math symbols in order to easily and efficiently mix symbols normally found in different math versions. Note that it is rarely worth it to perform such definitions—not even theAMS document classes do it. For more details about NFSS see the standard \LaTeX documentation file fntguide.tex.

**Upright lower case Greek letters as symbols.** The font set includes a letters font with upright lower case Greek letters (l.c.G.). Usually italic or slanted l.c.G. are used as symbols, though recently it’s possible see the POST-SCRIPT Symbols font (upright; no one bothers to skew it) used a lot, but that is a more a sign of weak software. The upright l.c.G. letters are made available in both text and math (Sec. 3.9). It would be possible to make them available as math symbols by using the code below. (The code is not available as an option, because there is no common definition for how the symbols should be named; for illustration we call the symbols \upalpha, \upbeta, etc.)

First define a math symbol font:

\begin{verbatim}
\DeclareSymbolFont{upgreek}{LmtL}{mtl}{\mddefault}{\uidefault}
\end{verbatim}

Don’t redefine \texttt{\mathgr} to point to the \texttt{upgreek} family, because the encodings aren’t even close.

Next set the math versions:

\begin{verbatim}
\SetSymbolFont{upgreek}{normal}{LmtL}{mtl}{\mddefault}{\uidefault}
\SetSymbolFont{upgreek}{bold}{LmtL}{mtl}{\bfdefault}{\uidefault}
\SetSymbolFont{upgreek}{heavy}{LmtL}{mtl}{\hvdefault}{\uidefault}
\end{verbatim}

If this step is omitted, the \texttt{upgreek} symbol font family will be the same in all math versions.

Lastly, make the symbol declarations (choose names that don’t conflict with existing symbols):

\begin{verbatim}
\DeclareMathSymbol{\upalpha}{\mathord}{upgreek}{"0B}
\end{verbatim}
That is the complete set; of course, you need not define them all.

Additional math accents can be defined in an entirely analogous way by using \DeclareMathAccent. NFSS can define two types of accents: \mathord, which are taken from a fixed font, and \mathalpha, which change with math alphabets. A specially declared accent will probably be \mathord; don't use \mathalpha unless you know what you're doing. An example for a math symbol font family called boldsyms which has an accent in slot "52 is

\DeclareMathAccent{boldddot}{\mathord}{boldsyms}{"52}

Declaring additional math delimiters Although any combination of symbols from various math versions can be achieved through the use of \boldsymbol and \heavysymbol, it can be inefficient and awkward to have to do so frequently. Particularly tedious are delimiters, which either require freezing their size (\boldsymbol{\Bigg(}), or cryptic constructs involving \vphantoms. In this case it is also possible to define special math symbols that work as expected. The procedure is similar to the one for declaring upright Greek letters as symbols, an outline follows. \TeX's math workings are described in D. E. Knuth's The \TeXbook and, e.g., in \TeX by topic by V. Eijkhout.

First declare a new symbol font, e.g.,
Next define the delimiter itself:
\begin{verbatim}
\DeclareMathDelimiter{(cmd)(mathtype) (small)(small slot)(large)(large slot)}
\end{verbatim}

where \{\small\} and \{\large\} are two symbol font math families. If you’re defining a new delimiter from some special font, you probably only have one variant (or more precisely one chain of linked delimiters). This command does not allow making a variant null (000) without assuming that \texttt{operators} is math family 0, but it is possible to make sure the small variant “succeeds” in finding a delimiter (e.g., an extensible delimiter will always terminate a search), so that the large variant is irrelevant.

For example, assuming there is a math symbol font family called \texttt{hvlargesymbols} (see above), \texttt{\lbraceHV and \rbraceHV} will produce correctly sized heavy braces:
\begin{verbatim}
\DeclareMathDelimiter{\lbraceHV}{\mathopen}{hvlargesymbols}{"08}{hvlargesymbols}{"08}
\end{verbatim}
\begin{verbatim}
\DeclareMathDelimiter{\rbraceHV}{\mathclose}{hvlargesymbols}{"09}{hvlargesymbols}{"09}
\end{verbatim}

(The small and large variants are identical, but since the small one will always succeed—the last delimiter in the chain is extendible—the large one never comes into play.) \texttt{\delcodes} are not normally changed.

\textbf{Running out of math families} \LaTeX{} with NFSS is limited to 16 math font families in any single math version, and it’s not all too difficult to run into the limit. With the default options, it’s possible to load four more math symbol fonts/alphabets. (A \texttt{\DeclareSymbolFont} and an accompanying \texttt{\DeclareSymbolFontAlphabet} take up one family.) The options \texttt{nomathgr} and \texttt{nomathOS} will each cause one less math family to be used. The \texttt{cmcal} option uses one math family more than \texttt{mtcal} or \texttt{nomtms} (as long as 8r is OK for the four infamous symbols). If you do run out of math families, declare a new math version, or define infrequently used symbols as explicit font/character references in text mode (in an \texttt{\hbox}), possibly wrapped in \texttt{\mathchoice} and an explicit class command, e.g., \texttt{\mathbin}. \texttt{See fntguide.tex}.

\subsection{Unused glyphs}

This package leaves no unique MathTime glyphs inaccessible. A few glyphs are repeated more than once in a font, or are present in more than one font. It makes no difference to \LaTeX{} whether a font is taken from \texttt{letters} or \texttt{symbols}.

It would be possible to take the small square root sign from the \texttt{largesymbols} font, rather than from the \texttt{symbols} font. That \texttt{largesymbols} radical ("9F) is, however, completely identical to the one in \texttt{symbols}, so it’s rather pointless. The command would be
\begin{verbatim}
\DeclareMathRadical{\sqrtsign}{largesymbols}{"9F}{largesymbols}{"70}
\end{verbatim}
The extra root sign in $\texttt{mtsyn}$ (not present in $\texttt{mtsy}$) is identical to the usual one, except for a vertical shift. Since a ‘radical’ sign is normally of class 1 ($\mathop)$, it gets vertically centered anyway. This alternate radical glyph could be used by defining something like

\begin{verbatim}
\DeclareMathSymbol{\varsurd}{\mathop}{symbols}{"84}
\end{verbatim}

### 6.14 muskip Assignments

A little smaller than for Computer Modern. The original setting is

\begin{verbatim}
\thinmuskip=3mu
\medmuskip=4mu plus 2mu minus 4mu
\thickmuskip=5mu plus 5mu
\end{verbatim}

and we’ll use

\begin{verbatim}
\thinmuskip=2mu \relax
\medmuskip=2.5mu plus 1mu minus 1mu \relax
\thickmuskip=4mu plus 1.5mu minus 1mu \relax
\end{verbatim}

### 6.15 Local Configuration File

Provide a hook to load anything important or complicated enough to be inappropriate for the main input file.

\begin{verbatim}
\InputIfFileExists{mt11p.cfg}{\typeout{**************************************^^J%
* Local config file mt11p.cfg loaded */\^J%
**************************************}}%
\end{verbatim}

### 6.16 Cleanup

Do the resizing now:

\begin{verbatim}
\mt@exit
\end{verbatim}

Restore any crazy catcodes.

\begin{verbatim}
\catcode`\mt@quotedbl\undefined
\catcode`\mt@quotesinglr\undefined
\catcode`\mt@colon\undefined
\catcode`\mt@question\undefined
\catcode`\mt@semicolon\undefined
\catcode`\mt@exclam\undefined
\end{verbatim}

Make the various scratch macros, etc., we used \texttt{\undefined}, but keep \texttt{\mt@cal} and \texttt{\mt@ops}, in case we need to decide something based on them. A ‘real’ input file should not do such a thing, but a demo document might.

\begin{verbatim}
\let\mt@quotedbl\undefined
\let\mt@quotesinglr\undefined
\let\mt@colon\undefined
\let\mt@question\undefined
\let\mt@semicolon\undefined
\let\mt@exclam\undefined
\end{verbatim}
...and we're done.