MathTime™ *Professional*

PostScript fonts for typesetting mathematics with TEX

Version 4

The *MathTime*™ *Professional* Fonts

Introduction

The *MathTimeProfessional* fonts* are PostScript fonts designed to allow the highest quality mathematical typesetting with TEX. Normally, TEX produces formulas like $z^3 = x^3 + y^3$ and

$$C = \sum \frac{\partial m}{\partial x} \frac{\partial n}{\partial y} + \frac{\partial \alpha}{\partial \zeta} \frac{\partial \gamma}{\partial \xi}$$

and

$$\left(\sqrt{\frac{M}{1 - \left(\frac{r}{\widetilde{x_1 + \cdots + u_N}} \right)^2} \left(\sum_{\beta=1}^{N} \sum_{i=1}^{n} \frac{\partial u_\beta}{\partial x_i} + 1 \right)} + \sqrt{XY} \right)^3$$

that were designed to be used with the “Computer Modern” family of text fonts. Unfortunately, these formulas don’t match up very well with many of the traditional text fonts that one can now use in TEX, like the “Times” font that is being used here.

The *MathTimeProfessional* fonts were specifically designed to be used with Times, producing formulas like $z^3 = x^3 + y^3$ and

$$C = \sum \frac{\partial m}{\partial x} \frac{\partial n}{\partial y} + \frac{\partial \alpha}{\partial \zeta} \frac{\partial \gamma}{\partial \xi}$$

that match the style of the Times text fonts; they also allow other special constructions that improve the results previously obtainable with TEX:

$$\left(\sqrt{\frac{M}{1 - \left(\frac{r}{\widetilde{x_1 + \cdots + u_N}} \right)^2} \left(\sum_{\beta=1}^{N} \sum_{i=1}^{n} \frac{\partial u_\beta}{\partial x_i} + 1 \right)} + \sqrt{XY} \right)^3$$

**MathTime* is a trademark of Publish or Perish, Inc.

These fonts also turn out to be quite compatible with various other text fonts, though the text font size may have to be adjusted slightly, so that the x-height will match. The designer of the *MathTimeProfessional* fonts like to use them with the Monotype Baskerville text fonts; the Baskerville italic text letters like *m*, *n*, *x*, and *y* are quite different from the letters m, n, x and y that appear in formulas, but this might be regarded as an advantage, giving the mathematics letters greater weight.

The original *MathTime* fonts lacked the special features of the *MathTime Professional* fonts—the large size radical signs, parentheses, and other delimiters that replace the symbols previously made from “extensible” characters, and the wide math accents that grow much wider.

But the most significant defect was the lack of a design feature that the Computer Modern fonts have retained: the characters used for superscripts were simply scaled down versions of those used for ordinary sizes (something so easily achieved with PostScript), leading to formulas like C^{xy} with cramped, rather spindly superscripts, rather than characters from a font specifically designed to be used at a smaller size, producing more pleasing formulas like C^{xy}.

The *MathTimeProfessional* fonts restore this feature (which was always used for hot-metal type, but unceremoniously abandoned with the advent of digital typesetters). So, instead of the *MathTime* formula

$$A'^{\beta_1\ldots\beta_l}_{\alpha_1\ldots\alpha_k} = \sum_{\substack{i_1,\ldots,i_k\\ j_1,\ldots,j_l}} A^{j_1\ldots j_l}_{i_1\ldots i_k} \frac{\partial x^{i_1}}{\partial x'^{\alpha_1}} \cdots \frac{\partial x^{i_k}}{\partial x'^{\alpha_k}} \frac{\partial x'^{\beta_1}}{\partial x^{j_1}} \cdots \frac{\partial x'^{\beta_l}}{\partial x^{j_l}}$$

with its skinny superscripts and downright anorexic second order subscripts, with the *MathTimeProfessional* fonts we can now get the much more readable formula

$$A'^{\beta_1\ldots\beta_l}_{\alpha_1\ldots\alpha_k} = \sum_{\substack{i_1,\ldots,i_k\\ j_1,\ldots,j_l}} A^{j_1\ldots j_l}_{i_1\ldots i_k} \frac{\partial x^{i_1}}{\partial x'^{\alpha_1}} \cdots \frac{\partial x^{i_k}}{\partial x'^{\alpha_k}} \frac{\partial x'^{\beta_1}}{\partial x^{j_1}} \cdots \frac{\partial x'^{\beta_l}}{\partial x^{j_l}}.$$

The italic letters in the *MathTimeProfessional* fonts come from the original (1929) Monotype “Times New Roman” font family, which was in three different sizes; the Greek letters and other symbols from the fonts were then individually designed in the different point sizes to match the style of the letters from these Monotype fonts. (On the other hand, the text of this document is simply printed in the standard "Times" font, the one that comes automatically with all PostScript printers.)

Of course, all these PostScript fonts can themselves be scaled. In fact, in this guide, both the text fonts and the mathematics fonts have been magnified slightly (by about 105%) to make for easier reading.

The *MathTime Professional* package involves a multitude of files, which must be placed in the proper directories and subdirectories. Installation of the *MathTime Professional* fonts may have been done for you automatically (for example, if you are using the fonts with PCTEX). Otherwise you may first need to consult the separate *MathTime Professional* Installation document.

I. *MathTime Professional* and LaTeX

1. The *MathTime Professional* fonts can be used with LaTeX by means of the style file `mtpro.sty`. Be sure to use the latest version 4 of this style file, designed by Walter Schmidt, for version 4 of the *MathTime Professional* fonts.

Although basic information for using the *MathTime Professional* fonts with LaTeX is covered in the current guide, you may want to consult `mtpro.pdf` for more detailed information, especially if you are going to be using the fonts together with other packages. Complete documentation, including implementation details, can be obtained from the source file `mtpro.dtx`.

Your LaTeX file will begin with something like

```
\documentclass{article}
. . .
\usepackage[subscriptcorrection,nofontinfo]{mtpro}
```

The '`nofontinfo`' option will dispense with lengthy messages in the log file about the various new fonts that get loaded. The '`subscriptcorrection`' option provides better placements of certain subscripts, for example, one obtains C_f, C_j, X_A rather than C_f, C_j, X_A.

(a) When the '`subscriptcorrection`' option is being used, the underscore character `_` operates specially in math mode, so it could conceivably conflict with additional packages. The '`mtpro`' package also provides the `\enablesubscriptcorrection` command, as well as the companion command `\disablesubscriptcorrection`, to turn the option on and off within the document.

If you are using a LaTeX style that prints in 10 point type, say, then in formulas the normal size letters of the *MathTime Professional* fonts will be 10 point type, the superscripts will be 7 point type, and the second order superscripts will be 5.5 point type.

2. Loading the *MathTime Professional* fonts will cause these fonts to be used in mathematics formulas, but it will not affect text—if you are using Computer Modern, for example, then your text font will remain Computer Modern.

The LaTeX '`times`' package is used to get the Times fonts in text, so you will probably want to start your file with something like

```
\documentclass{article}
\usepackage{times}
\usepackage[...]{mtpro}
```

The `\usepackage{times}` should precede `\usepackage{mtpro}`, so that Times will also be used for operator names like 'sin' and 'cos', and more generally whenever you use roman type within a math formula (a few subtleties about roman text in math formulas are discussed in section **IX**).

II. *MathTime Professional* and `plain` TeX

1. To use the *MathTime Professional* fonts with `plain` TeX, just add the line

```
\input mtp
```

at the beginning of your file, causing TeX to read in the file `mtp.tex`. Be sure to use version 4 (or later) of `mtp.tex` for version 4 of the *MathTime Professional* fonts. If your file also has the line `\input amstex`, this should *precede* the line containing the command `\input mtp`.

2. For `plain` TeX, you might be using a special macro package that chooses the Times fonts for text. In this case, just load that macro package before you `\input mtp`.

If you are not relying on some such macro package, then you will have to declare and load the text fonts yourself. Since you not only want your roman font, ‘`times`’ say, to be used in text but also for operator names like ‘sin’ and ‘cos’, you probably want to do something like

```
\font\tentimes=times at 10pt
\font\seventimes=times at 7pt
\font\fivetimes=times at 5.5pt
\textfont0=\tentimes
\scriptfont0=\seventimes
\scriptscriptfont0=\fivetimes
\def\rm{\fam0 \tentimes}
```

Section **IX** covers some subtleties about roman fonts in math formulas.

3. Normally, `mtp.tex` will use 10 point type for the normal size letters in math formulas, 7 point type for superscripts, and 5.5 point type for second order superscripts. If you’ve used something like the `\mag` command or the `\magnification` command to magnify type size, then the *MathTimeProfessional* fonts will be magnified accordingly. However, if you’ve decided to change font sizes “manually”, say by typing

```
\font\RM= Times at 11pt
\RM
\baselineskip = 13pt
```

then you can choose the appropriate size for the *MathTimeProfessional* fonts by using the `\MTPsizes` command. For example, you could multiply the sizes by approximately 110% by typing

```
\MTPsizes{11pt}{7.7pt}{6pt}
```

III. Changes for the *MathTimeProfessional* Fonts

Most of the time, when you are typing a mathematics formula in TEX you can remain oblivious of the fact that the formulas are eventually going to be typeset in the *MathTimeProfessional* fonts. However, there are a few things that you have to be careful about.

1. (a) In `plain` TeX, the slanted upper-case Greek letters $\mathit{\Gamma}$, $\mathit{\Delta}$, ... are obtained by typing `\mit\Gamma`, etc. But that won't work with the *MathTime Professional* fonts, which actually make `\mit` undefined. Instead, you should just type `\varGamma`, `\varDelta`, etc. These new commands shouldn't seem very strange, since they are analogous to `\varepsilon` and similar commands. As a matter of fact, the *MathTimeProfessional* fonts have another variant Greek letter, $\varkappa$, which is typed as `\varkappa`.

If you are using LaTeX, there is a '`slantedGreek`' option in the '`mtpro`' package that causes `\Gamma`, ... , to produce slanted letters automatically, although upright letters can also be obtained—see **(c)**.

(b) In addition to $\varkappa$, there is ϐ (`\varbeta`), an old form of β that you might find useful if you are trying to imitate certain old books. Similarly, you can type `\vardelta` to obtain an old style ∂. This is hardly distinguishable from the `\partial` symbol ∂ (the circular portion of `\vardelta` is taller, to match the height of letters like x and o in math formulas), and is included only because all the various Greek alphabets (regular, bold, etc.) specified for mathematics in the Unicode standard include this variant (perversely called 'partial').

There is also the seldom-used old Greek letter Ϝ (`\digamma`).

(c) The *MathTimeProfessional* fonts also include upright lowercase Greek letters α, β, γ, ϰ, ϐ, ... , which you can get with `\upalpha`, `\upbeta`, `\upgamma`, `\upvarkappa`, `\upvarbeta`, ... Though upright lowercase Greek letters are not customarily used in mathematics, the bold variants (see section **VIII**) might be found useful.

2. The Computer Modern "old style" digits 0, ... , 9 are actually on the Computer Modern mathematics fonts (strangely enough!), so they are not available when using the *MathTimeProfessional* fonts.

> The fact that the old style digits for the Computer Modern Roman font `cmr10` reside on a separate font is not really all that exceptional; many PostScript fonts likewise have a companion "expert" font that contains old style digits in the appropriate ASCII positions, though they usually contain "caps and small caps" in the letter positions, unlike `cmmi10`, which instead contains the italic letters for math formulas. Section **X.8** explains how one can utilize the old style numbers for Computer Modern Roman or any PostScript font with a suitable companion font.

IV. New Symbols

Aside from additional letters, like $\varkappa$, already mentioned, there are numerous other symbols and refinements provided by the *MathTimeProfessional* fonts.

1. The yen symbol ¥ (`\yen`) is normally used outside of math mode, but can also be used in math mode, as a new letter (e.g., $¥^{¥^{¥}}$). There are also $\hslash$ (`\hslash`) and $\hbar$ (`\hbar`) [see also **8.** concerning $\hbar$].

2. In addition to `\circ`, there is a slightly smaller circle, `\comp`, which looks better for the "composition of functions", like $f \circ g$ (`f\comp g`), as compared to $f \circ g$ (`f\circ g`). But use `\circ` for things like 22° (`22^\circ`).

3. Three new symbols have been added for formulas like

$$\alpha \cup \beta \in H^*(A \cup B) \implies \alpha \cap \beta \in H_*(A \smallsetminus B)$$

which might otherwise have been rendered as

$$\alpha \smile \beta \in H^*(A \cup B) \implies \alpha \frown \beta \in H_*(A \setminus B)$$

The use of $\setminus$ (`\setminus`) for the difference of sets can easily cause confusion in algebraic contexts, where double cosets $G\backslash H$ are also used (and where $A - B$ might also be confusing). So the *MathTimeProfessional* fonts provide the alternate symbol $\smallsetminus$ (`\setdif`).

Likewise, mathematicians may feel that $\smile$ and $\frown$ (`\smile` and `\frown`) are too shallow to be used for the "cup-product" and "cap-product", though these must also be distinguished from set-union ($\cup$) and set-intersection ($\cap$). So the *MathTimeProfessional* fonts also supply the symbols $\cup$ (`\cupprod`) and $\cap$ (`\capprod`). There are also corresponding 'large operators' `\bigcupprod` ($\bigcup$ and $\bigcup$) and `\bigcapprod` ($\bigcap$ and $\bigcap$).

4. Accompaning `\int` and `\oint`, there are now

$\iint$	$\displaystyle\iint$	`\iint`	$\ointclockwise$	$\displaystyle\ointclockwise$	`\cwoint` (clockwise `\oint`)
$\iiint$	$\displaystyle\iiint$	`\iiint`	$\ointctrclockwise$	$\displaystyle\ointctrclockwise$	`\awoint` (anticlockwise `\oint`)
$\oiint$	$\displaystyle\oiint$	`\oiint`	$\intclockwise$	$\displaystyle\intclockwise$	`\cwint` (clockwise `\int`)
$\oiiint$	$\displaystyle\oiiint$	`\oiiint`			

5. `\mathring x` (alternatively, `\oacc x`) produces $\mathring{x}$, frequently used for the initial values of x (and sometimes for quaternions). `\mathring` (or `\oacc`) is a math accent, like `\hat`, etc., so it will be positioned properly in constructions like $\mathring{A}$.

6. In situations like $\dot{\Gamma}$ and $\ddot{\Gamma}$ the `\dot` and `\ddot` accents might look better if they were moved up a bit. You can get $\dot{\Gamma}$ and $\ddot{\Gamma}$ with `\dotup` and `\ddotup`. (If you are using $\mathcal{AMS}$-TeX there are also `\Dotup` and `\Ddotup` for double accents.)

7. The math accents ˆ (`\hat`), ˇ (`\check`), ˜ (`\tilde`), and ¯ (`\bar`) have slightly wider versions

ˆ (`\what`), ˇ (`\wcheck`), ˜ (`\wtilde`), ¯ (`\wbar`)

[with corresponding `\Wcheck`, etc., if you are using $\mathcal{AMS}$-TeX], and then versions that are slightly wider still:

ˆ (`\wwhat`), ˇ (`\wwcheck`), ˜ (`\wwtilde`), ¯ (`\wwbar`).

These commands may be used to fine tune accents over upper-case letters. For example, instead of

$\hat{M}$ (`\hat M`) or $\widehat{M}$ (`\widehat M`)

you might prefer

$\hat{M}$ (`\what M`) or $\widehat{M}$ (`\wwhat M`).

8. Numerous standard TeX symbols were traditionally constructed from two or more characters. For example, $\hbar$ (`\hbar`) was made from h and the bar accent ¯, and $\hookleftarrow$ was made from the $\leftarrow$ and a ↄ symbol. But such built-up symbols often do not work well in superscripts or when different fonts are being used, and the *MathTimeProfessional* fonts now contain individually designed characters that are used instead. Two additional symbols, $\dddot{}$ (`\dddot`) and $\ddddot{}$ (`\ddddot`) from $\mathcal{AMS}$-TeX have also been added (there are also `\dddotup` and `\ddddotup`, and if you are using $\mathcal{AMS}$-TeX there are also `\Dddot`, etc., for double accents).

In addition, numerous symbols can be “negated” by prefixing them with `\not`, like $\not<$ (`\not<`), $\not\subset$ (`\not\subset`), and $\not\approx$ (`\not\approx`). All these symbols now exist as individually designed symbols, each with its own name; alternative names, in brackets, are for compatibility with the names provided in Font Supplement A.

$\nless$	`\notless [\nless]`	$\ngtr$	`\notgr [\ngtr]`
$\nleq$	`\notleq [\nleq]`	$\ngeq$	`\notgeq [\nqeg]`
$\nprec$	`\notprec [\nprec]`	$\nsucc$	`\notsucc [\nsucc]`
$\npreceq$	`\notpreceq [\npreceq]`	$\nsucceq$	`\notsucceq [\nsucceq]`
$\not\subset$	`\notsubset [\nsubset]`	$\not\supset$	`\notsupset [\nsupset]`
$\nsubseteq$	`\notsubseteq [\nsubseteq]`	$\nsupseteq$	`\notsupseteq [\nsupseteq]`
$\not\sqsubseteq$	`\notsqsubseteq [\nsqsubseteq]`	$\not\sqsupseteq$	`\notsqsupseteq [\nsqsupseteq]`
$\ncong$	`\notcong [\ncong]`		
$\neq$	`\neq`	$\not\equiv$	`\notequiv`
$\nsim$	`\notsim`	$\not\simeq$	`\notsimeq`
$\not\approx$	`\notapprox`	$\not\asymp$	`\notasymp`

The name '`\notequal`' wasn't introduced because the name `\neq` already exists; as before, `\ne` may be used as a synonym. Font Supplement A also contains a character `\nsim` ($\nsim$), but this is different from `\notsim`.

[The `\not` symbol is still on the fonts, as are the hook $\lhook$ (`\lhook`) and the hook $\rhook$ (`\rhook`), so one can imitate the definitions in `plain.tex` to build a longer hooked arrow $\xhookrightarrow{\quad}$ and extra long arrows like $\longleftarrow$ and $\Longleftarrow$ (the extending character $=$ (`\Relbar`) for the double arrows actually uses an additional symbol, rather than the $=$ sign used by the Computer Modern fonts). Similarly, an extra long $\longmapsto$ can be constructed using the character $\mapstochar$ (`\mapstochar`).]

9. Finally, there are the special symbols

♧ (`\openclubsuit`) ♣ (`\shadedclubsuit`)
♤ (`\openspadesuit`) ♠ (`\shadedspadesuit`)

These are mainly provided as a complement to the rather grotesque ♣ and ♠ from the bold mathematics fonts (see **VIII**).

V. Bold Roman Letters

In a math formula, `\bf` gives the bold font that is used in text (in $\mathcal{AMS}$-TEX one would use `\bold`, which is a control sequence with an argument, rather than a font change, and LaTeX has the similar command `\mathbf`). So if you are using Computer Modern fonts for text, you will get the bold Computer Modern fonts; if you are using the '`times`' package (as on page 4), then you will get the Times bold font.

But there is an additional `\mbf` command that selects letters from the bold fonts of the Monotype "Times New Roman" font family. This command works both in `plain` TeX and with LaTeX, and like the `\mathbf` command from LaTeX, it is a control sequence with an argument.

`$\mbf{f=g^h}+ j$` $\mathbf{f} = \mathbf{g}^{\mathbf{h}} + j$

The bold letters will match up better with the italic letters in math formulas; in addition, the spacing around these bold letters has been changed so that subscripts and superscripts will work better with them. But if for some reason you want something like $\mathbf{Major}(X)$, it is better to use `{\bf Major}` (or `\bold{Major}` in $\mathcal{AMS}$-TeX, or `\mathbf{Major}` with LaTeX) instead of `\bmf{Major}`, because the spacing with `\mbf` isn't meant for text, so you would get $\mathbf{M\,a\,j\,o\,r}(X)$.

VI. The Big Differences

1. The formula

$$\left(\begin{matrix} A_{11} & \dots & A_{1n} \\ A_{21} & \dots & A_{2n} \\ & \ddots & \\ A_{n1} & \dots & A_{nn} \end{matrix}\right)$$

shows the *MathTimeProfessional* "extensible" parentheses that one obtains with a `\left(...\right)` construction. But the *MathTimeProfessional* fonts also have individually designed parentheses of the appropriate size. To obtain these, it is only necessary to use the `\PARENS{...}` construction instead:

$$\begin{pmatrix} A_{11} & \dots & A_{1n} \\ A_{21} & \dots & A_{2n} \\ & \ddots & \\ A_{n1} & \dots & A_{nn} \end{pmatrix}$$

These individually designed parentheses go up to 4 inches high! (In the unlikely event that your formula is even larger, you will be stuck with the 4 inch high versions, and will have to go back and change to `\left(...\right)` by hand.)

Quite a few other delimiters also have individually designed characters up to 4 inches high. In fact, `\PARENS{...}` is basically just an abbreviation for

```
\LEFTRIGHT(){...}
```

where the `\LEFTRIGHT` construction can be followed by any two delimiters (including the period for an empty delimiter). For example,

```
\LEFTRIGHT\langle\rangle{...}
```

will give individually designed “angle brackets” $\langle \quad \rangle$ up to 4 inches high. (Of course, you will be out of luck if you need even larger angle brackets, since extensible angle brackets don’t exist.) As with `\left` and `\right`, you can also type `<` and `>` instead of `\langle` and `\rangle` in this context, and you can also use the delimiters `/` and `\backslash`, as well as the usual delimiters `|`, `\|`, `[`, `]`, etc.

2. Delimiters aren’t the only things that grow much bigger with the *MathTime Professional* fonts. The `\widehat` and `\widetilde` accents automatically grow up to 4 inches wide,

$$\widehat{A+B+C+D+E+F+G}+\widetilde{H+I+J+K+L+M+N}$$

and a `\widecheck` has also been provided. Once again, if your formulas require even wider accents, then you will end up stuck with the 4 inch wide ones.

3. Similarly, instead of the formula

$$\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

with an extensible radical sign, we can use `\SQRT{...}` to get radical signs up to 4 inches high:

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Similarly, `\ROOT 3\OF{...}` [or `\SQRT[3]{...}` in LaTeX], can be used to get

$$(*) \qquad \sqrt[3]{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

4. There are a few cautions, and a few subtleties, that need to be addressed, so you might have to come back to this section if things don't look quite right, or something strange occurs when you are using one of the commands in this section.

(a) Although `\left...\right` can be used with any formula, it is normally applied only around constructions like `\matrix` that are "vertically centered" (their baseline is close to the vertical midpoint of the construction). The `\LEFTRIGHT` command always vertically centers its argument automatically.

(b) It is possible to nest `\PARENS` (or `\LEFTRIGHT`) in the standard way,

```
\PARENS{  ...  \PARENS{...}  ...  }
```

but that should hardly ever be needed, and such nesting slows TeX down exponentially; it might even cause a '`!TeX capacity exceeded`' error message.

(c) It should also be mentioned that `\PARENS` (or `\LEFTRIGHT`) ends up setting its argument more than once, so you need to be careful if you are using `\box`'s. For example, if you've stored a formula in `\box\eqnbox`, then you should be sure to type `\PARENS{\copy\eqnbox}`, rather than `\PARENS{\box\eqnbox}`.

The same precaution applies to the new `\wide...` commands and to `\SQRT` (and `\ROOT...\OF{...}`).

(d) The placement of the ‘root’ 3 in formula (∗) on page 12 can be modified by typing

`\uproot{`⟨number⟩`}` and/or `\leftroot{`⟨number⟩`}`

before the `\ROOT` to move it up and/or to the left by ⟨number⟩ units. (In LaTeX, where `\SQRT[...]` is used, these commands should be typed as `\UPROOT` and `\LEFTROOT`.) The units by which the ‘root’ is moved are quite small, allowing for delicate positioning; formula (∗) was actually typed as

```
\leftroot{-8}\uproot{-1}\ROOT3\OF {...}
```

(modifications of this sort simply have to be done by experimentation).

Notice, by the way, that the use of `\uproot` and `\leftroot` differs from that originally found in $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX (the new syntax makes everything a lot easier).

The root $_3$ is normally set in ‘`\scriptscriptstyle`’, the size for second order superscripts, but you could also type something like

```
\ROOT\scriptstyle3\OF{...}
```

to get

$$\sqrt[3]{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{$*$}$$

with the root $_3$ in ‘`\scriptstyle`’ instead (you can experiment yourself with the necessary `\leftroot` and `\rightroot` amounts). You might even want to use ‘`\textstyle3`’ for a very tall root. (In $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX you can type `\ssize` instead of `\scriptstyle` and `\tsize` instead of `\textstyle`.)

(e) Small discrepancies might arise when large constructions are inserted into combinations like

`\LEFTRIGHT(|{...}` or `\LEFTRIGHT(]{...}`

because delimiters like `|` and `]` are simply created as extensible characters, rather than individually designed ones, so they may not grow in sync with parentheses and various other delimiters.

Thus, you might get formulas like like

$$\left(\dots \right] \text{ and } \left(\dots \right] \quad \text{rather than} \quad \left(\dots \right] \text{ and } \left(\dots \right]$$

when `\LEFTRIGHT(]` is used to surround constructions which have a total height $+$ depth of `84pt` and `85pt`, respectively.*

In such cases, you can use the `\extcorrect` command to correct the extensible character. For example, in the current situation we obtained the desired results by typing

```
\extcorrect{80pt}\LEFTRIGHT[){...}\extcorrect{90pt}...
```

The proper correction, which has to be determined by experimentation, will be close to the total height $+$ depth of the formula enclosed by the delimiters; values slightly smaller that this total should be used to decrease the height of the `]`, while slightly larger values will be required to increase the height. Each `\LEFTRIGHT` construction resets the correction to 0.

There's also a `\vertcorrect` (which overrides any `\extcorrect`) that can be used to make adjustments in a `\PARENS`, or in a `\LEFTRIGHT` construction that uses only the nonextensible delimiters.

*The exact results will depend on the values that your style uses for the special TeX parameters '`\delimiterfactor`' and '`\delimitershortfall`'; the values `1000` and `0pt` are being used in this example.

VII. Even Bigger Differences

In displayed formulas like

$$\sum_{i\notin I}\frac{\displaystyle\int_{-\infty}^{\infty} f(\alpha_i x)\,dx+1}{\displaystyle\oint_C f(\beta_i z)\,dz-1} \qquad \int_{-\infty}^{\infty}\frac{\displaystyle\sum_{i=1}^{\infty}\frac{\delta_i\cdot e^{f\left(\sum_{i=1}^{\infty}\alpha_i x\right)}}{1-\delta_i}}{\displaystyle\beta_i+\sum_{i=1}^{\infty}\frac{\delta_i\cdot e^{f\left(\sum_{i=1}^{\infty}\alpha_i x\right)}}{1-\delta_i}}\,dx$$

you might feel the need for larger $\sum$ and $\int$ signs.

Normally printers don't provide these, but with the *MathTimeProfessional* fonts you can get an extra large $\sum$ with the `\xl` command

`\xl{i=1}{\infty}\sum` $$\sum_{i=1}^{\infty}$$

and an even bigger version with the `\XL` command

`\XL{i=1}{\infty}\sum` $$\sum_{i=1}^{\infty}$$

Notice that the 'limits' are now given as the first two arguments of `\xl` and `\XL`, rather than as a subscript and superscript to `\sum`. The two arguments **must** appear, even if one, or both, are empty:

`\XL{i\in I}{}\bigcup` $$\bigcup_{i\in I}$$ `\XL M{}\int` $$\int_M$$

You can use `\xlnl` and `\XLNL` if you want 'no limits', i.e., if you'd like these limits set at subscript and superscripts:

`\XLNL{i=1}\infty\bigcap` $$\bigcap\nolimits_{i=1}^{\infty}$$

and similarly

`\XLNL{-\infty}\infty\int ...` $$\int_{-\infty}^{\infty} \frac{\displaystyle\sum_{i=1}^{\infty} \frac{\delta_i \cdot e^{f\left(\sum_{i=1}^{\infty} \alpha_i x\right)}}{1-\delta_i}}{\displaystyle\beta_i + \sum_{i=1}^{\infty} \frac{\delta_i \cdot e^{f\left(\sum_{i=1}^{\infty} \alpha_i x\right)}}{1-\delta_i}}\, dx$$

`\xl` and `\XL` can be applied to all 'large operators', including those in section **IV. 3**; in most cases `\XL` produces a symbol half an inch high (36pt). There are also `\XXL` versions that are a full inch high! And, heaven forbid, you can even get `\XXXL` versions that are two inches high, thereby assuring yourself (and probably me) the lasting enmity of journal editors everywhere.

VIII. Bold (*Italic*) Letters and Symbols

The *MathTimeProfessional* fonts also provide bold versions of the basic math fonts, as well as 'heavy' (extra-bold) versions of many symbols. To use these fonts with LaTeX, simply use the standard `bm` package from LaTeX together with the `mtpro` package; see `mtpro.pdf` for further details. The remainder of this section is for `plain` TeX users.

1. One possible use of the bold math fonts is for section headings like

4. The Pythagorean's discovery that $\boldsymbol{\sqrt{2}}$ is irrational.

or

6. Fermat's conjecture about $\boldsymbol{x^n + y^n = z^n}$ for $\boldsymbol{n > 1}$.

You can even get bold displayed formulas, like

$$\boldsymbol{\frac{1}{\sqrt{\pi}}\left(\int_{-\infty}^{\infty} e^{-x^2}\,dx\right) = 1}$$

though such displays are completely non-standard, and probably of almost no use (simply putting a box around a formula accentuates it more effectively). Even the use of bold math in section headings is problematical—what do you do with a formula that contains a vector product like $\mathbf{x} \times \mathbf{y}$?

Most of the time, the bold math fonts are used simply to augment the existing supply of letters and symbols, providing bold letters like $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, in contrast to the usual math symbols x, y, z, and bold symbols $\boldsymbol{+}$, $\boldsymbol{<}$, $\boldsymbol{>}$, in contrast to the usual symbols $+$, $<$, $>$.

The use of bold letters like $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ is actually rather rare, since normally mathematicians simply use ordinary upright bold letters $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ (with the *MathTime Professional* fonts these are best obtained as `\mbf x`, `\mbf y`, and `\mbf z`). On the other hand, bold Greek letters, like $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ are somewhat more common; if these bold Greek letters play the same role as the ordinary bold letters $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, you might prefer upright versions $\boldsymbol{\upalpha}$, $\boldsymbol{\upbeta}$, $\boldsymbol{\upgamma}$, which are also provided.

Bold *symbols* are used quite frequently, and in this case it is usually the *contrast* with the standard symbols that is important, so instead of bold symbols like $\boldsymbol{+}$, $\boldsymbol{<}$, $\boldsymbol{>}$, you might prefer “heavy” symbols like $\boldsymbol{+}$, $\boldsymbol{<}$, $\boldsymbol{>}$, which are also provided.

2. To use the bold italic letters and symbols, put `\loadbm` in your file (after the `\input mtp`). To get a bold version of a whole formula (even a displayed one), just type `\boldmath` at the beginning:

```
$\boldmath x^n+y^n=z^n$

$$\boldmath{1\over\sqrt\pi}\left(...\right)=1$$
```

When you use `\boldmath` before a formula that involves `\PARENS{...}`, that combination is basically replaced by `\left(...\right)`, because the bold delimiters don’t come in all the extra large sizes that *MathTimeProfessional* makes available for ordinary delimiters.

Similarly, `\SQRT` is essentially replaced by `\sqrt`, and wide accents like `\widehat` are available in only a few sizes, and so forth.

Note, by the way, that you can't use `\boldmath` for only part of a formula; `$A\boldmath B$` gives $\boldsymbol{AB}$, while `$A{\boldmath B}$` gives AB, so it doesn't work either.* Nevertheless, it is easy to get $A\boldsymbol{B}$, because there is also a way of getting individual bold symbols.

3. Individual bold symbols are obtained as follows:

(a) The bold letters $\boldsymbol{a}, \dots, \boldsymbol{z}, \boldsymbol{A}, \dots, \boldsymbol{Z}$ are simply obtained as `\bm a`, ... `\bm z`, `\bm A`, ... , `\bm Z`. Similarly, bold numerals **0**, ... , **9** can be obtained as `\bm0`, ... , `\bm9`.

(b) The command `\bm` may also be used with virtually any other key that you would use in a math formula:

`\bm+`	$\boldsymbol{+}$	`\bm-`	$\boldsymbol{-}$	`\bm=`	$\boldsymbol{=}$	`\bm/`	$\boldsymbol{/}$
`\bm<`	$\boldsymbol{<}$	`\bm>`	$\boldsymbol{>}$	`\bm(`	$\boldsymbol{(}$	`\bm)`	$\boldsymbol{)}$
`\bm[`	$\boldsymbol{[}$	`\bm]`	$\boldsymbol{]}$	`\bm.`	$\boldsymbol{.}$	`\bm,`	$\boldsymbol{,}$
`\bm!`	$\boldsymbol{!}$	`\bm?`	$\boldsymbol{?}$	`\bm;`	$\boldsymbol{;}$	`\bm:`	$\boldsymbol{:}$
`\bm|`	$\boldsymbol{\vert}$	`\bm*`	$\boldsymbol{*}$				

You can also type `\bm'` and this works even better than one might dare to hope. For example, `$A''+A\bm'' +A\bm''^{\bm2}$` gives $A''+A\boldsymbol{''}+A\boldsymbol{''^2}$. Just in case, there's also `\bmprime` to get the bold prime symbol explicitly.

(c) For all other symbols, named by control sequences, there are corresponding bold symbols whose names are obtained by putting `bm` in front of them. For example, `\bmalpha` gives $\boldsymbol{\alpha}$, `\bmell` gives $\boldsymbol{\ell}$, `\bmvarkappa` gives $\boldsymbol{\varkappa}$, `\bmwp` gives $\boldsymbol{\wp}$, `\bmleq` or `\bmle` gives $\boldsymbol{\leq}$, and the operators † (`\dagger`) and ‡ (`\ddagger`) have bold versions **†** (`\bmdagger`) and **‡** (`\bmddagger`). There's even `\bmcolon`, the bold analogue of `\colon`, which is just a : with special spacing.

For the upright bold lowercase Greek letters there are `\bmupalpha` (**α**), `\bmupbeta` (**β**), `\bmupgamma` (**γ**), etc.

---

*TeXnical reason: the `\boldmath` command simply specifies a new set of math fonts, and a TeX formula uses whatever fonts have been specified at the time that the closing `$` sign is read. (LaTeX only allows `\boldmath` to be used *outside* of a math formula, so that you would type something like `{\bf\boldmath...that $\sqrt2$ is irrational}`, thus skirting the problem completely.)

All the “built-up” characters, like $\hbar$ (`\hbar`) and $\not\subset$ (`\notsubset`), which are now individually designed characters on the *MathTimeProfessional* fonts, have corresponding individually designed bold versions, like $\boldsymbol{\hbar}$ (`\bmhbar`) and $\boldsymbol{\not\subset}$ (`\bmnotsubset`). This includes $\boldsymbol{\hookleftarrow}$ (`\bmhookleftarrow`) as well as all the other sorts of arrows. (However, the alternative names, like `\nsubset`, do not have corresponding `\bm...` versions.)

All the accents have bold versions, like `\bmhat` and `\bmwhat`, including bold versions $\boldsymbol{\dddot{}}$ (`\bmdddot`) and $\boldsymbol{\ddddot{}}$ (`\bmddddot`) of the new accents `\dddot` and `\ddddot`. And there are also “wide” bold accents, like `\bmwidehat`, but they are only available in a few sizes.

You can also type things like

```
$$\bmsum_{i=1}^\infty\bmint_a^{\bm b}$$
```

to get

$$\boldsymbol{\sum_{i=1}^{\infty}\int_a^{b}}$$

(but there are no bold versions of the `\XL` symbols).

(d) For the bold braces $\boldsymbol{\{}$ and $\boldsymbol{\}}$ you can type `\bmlbrace` and `\bmrbrace`, and you can even type things like

```
$$\left\bmlbrace...\right\bmrbrace$$
```

You *can’t* type things like `\left\bm(...\right\bm)`, but there are also

`\bmlparens`	$\boldsymbol{(}$	`\bmrparens`	$\boldsymbol{)}$
`\bmlbrack`	$\boldsymbol{[}$	`\bmrbrack`	$\boldsymbol{]}$
`\bmvert`	$\boldsymbol{\vert}$	`\bmVert`	$\boldsymbol{\Vert}$

and these control sequences can all be used as delimiters with `\left` and `\right`, so that you can type `\left\bmlparens...\right\bmrparens`. Similarly, constructions like `\bigl\bmlparens` and `\biggl\bmlparens` are all allowed.

All the other standard delimiters, like `\lfloor`, `\uparrow`, ... , also have corresponding bold delimiters, `\bmlfloor`, `\bmuparrow`, etc.

4. Put `\loadhm` in your file in order to obtain heavy versions of the fonts. All the symbols have heavy versions, obtained analogously to the bold symbols. For example, `\hm+` gives $\boldsymbol{+}$, `\hm'` can be used to get heavy primes, as in A'', and `\hmlbrace` and `\hmrbrace` give $\boldsymbol{\{}$ and $\boldsymbol{\}}$.

But there are no heavy letters or Greek letters: `\hm A` isn't allowed and `\hmalpha` is undefined (the symbols `\hmell` ($\boldsymbol{\ell}$) and `\hmwp` ($\boldsymbol{\wp}$) are something of an exception to this rule). There is a `\heavymath` command, but it produces strange substitutes for letters and Greek letters.* Nevertheless, it can be useful for certain special effects, discussed in the next section.

5. Various special effects are possible if you know a bit about TeX `\box`'es and such.

(a) Although virtually all symbols have bold versions, some "constructions", like `\sqrt`, `\underbrace`, and `\overrightarrow` don't, so there's no easy way to get something like $\sqrt{x} + \boldsymbol{\sqrt{x}}$ (recall the caution in section **2**).

I can't imagine offhand why you would need such an expression, but if you do, then you could type

```
\newbox\sqbox
\setbox\sqbox=\hbox{$\boldmath\sqrt x$}
```

and obtain the formula as

```
$\sqrt x + \copy\sqbox$.
```

(`\copy\sqbox` is safer than `\box\sqbox` in case the symbol ends up in some construction that sets its argument twice.)

Similarly, you could get the formula $\sqrt{x} + \boldsymbol{\sqrt{}}x$ by first storing x in a new box `\xbox`, and then putting `$\boldmath\sqrt{\copy\xbox}$` into the box `\sqbox`.

---

*The behavior is somewhat different in `plain` TeX and in LaTeX, because LaTeX loads in a separate family of "heavy math italic" fonts, which have "slugs" (black rectangles like ▮) in most places, while `mtph.tex` saves a family by ignoring these fonts, instead substituting the usual text font. So `{\heavymath$\alpha$}` produces a slug in LaTeX, while the corresponding `$\heavymath\alpha$` for `plain` TeX produces 'ff' (since the position for `\alpha` in the math italic font happens to be the position for the 'ff' ligature on the text font).

And you can get $\sqrt{x} + \boldsymbol{\sqrt{x}}$ by typing

```
\newbox\xbox
\setbox\xbox=\hbox{$x$}
\newbox\sqbox
\setbox\sqbox=\hbox{\heavymath\sqrt{\copy\xbox}}
$\sqrt x+\copy\sqbox$
```

(b) Even more detailed knowledge of TeX would be required if you wanted special symbols like those in section **IV.7**, for example, an extra long bold hooked arrow $\boldsymbol{\longleftarrow\!\rhook}$. Imitating the definition in `plain.tex` is not so easy now, because there is no special name for the bold hook $\boldsymbol{\rhook}$. In `plain.tex`, the $\rhook$ character, `\rhook`, is defined by

```
\mathchardef\rhook="312D
```

You can imitate this command to define a `\bmrhook` character (or simply use an appropriate `\mathchar"3...`), except that the '`1`', which indicates the family where the math italic fonts reside, has to be replaced by (the hex number for) the family where the new bold math italic fonts reside. This number is stored in the control sequence `\mtbmi@@`, so you can use `\mathchar"3\mtbmi@@2D` for this bold hook (note that `@` must be made a letter while doing this). Similarly, `\mtbsy@@` and `\mtbex@@` are the hex numbers of the family for the bold symbol and extension fonts.

(c) If you need an extra long **heavy** hooked arrow $\boldsymbol{\longleftarrow\!\rhook}$, you will have to work a bit harder, because `plain` TeX makes no use of heavy math italic fonts. To compensate for this, there is a $\boldsymbol{\rhook}$ symbol on the heavy symbol fonts, but you will need to look at a font table to see the position of this, and similar, symbols. The hex numbers of the family for the heavy symbol fonts and the heavy extension font are stored in `\mthsy@@` and `\mthex@@`.

IX. Roman Fonts in Formulas

1. If you've used the LaTeX '`times`' package, or are using `plain` TeX with the Times fonts as described in section **II.2**, then the displayed formula

```
\sum_{{\mathrm for\ all\ }x>0}\sin x^{\cos x^{\tan x}}
```

(or with `\rm` instead of `\mathrm` in `plain` TEX) will be typeset as

$$\sum_{{\mathrm{for\ all\ }}x>0}\sin x^{\cos x^{\tan x}}$$

where the words ‘for all’ and the operator ‘cos’ are printed in ‘`times at 7pt`’ (i.e., a 70% reduction of Times), while the operator ‘tan’ is printed using a 50% reduction of Times.

As with the math fonts, it would be preferable to use specially designed roman fonts for these 7 point and 5 point letters. So the ideal arrangement would be to use the Monotype Times roman fonts corresponding to the italic fonts used for math formulas, to get

$$\sum_{{\mathrm{for\ all\ }}x>0}\sin x^{\cos x^{\tan x}}$$

These Monotype fonts are

file name	PostScript Name	
`tim`	TimesNRMT	(Times New Roman)
`sv`	TimesNRSMT	(Times New Roman Seven)
`tst`	TimesSmaTexMT	(Times Small Text)

2. If you have these fonts on your system, then the LATEX ‘`timesmt`’ package may be used instead of the usual ‘`times`’ package.

3. If you are using `plain` TEX with `\input mtp` at the beginning of your file, then the `\MonotypeMR` command will load the fonts properly. Or you can load fonts directly, as in section **II.2**:

```
\font\tentimes=tim
\font\seventimes=sv at 7pt
\font\fivetimes=tst at 5.5pt
\textfont0=\tentimes
...
```

If you need to change point sizes, as in section **II.3**, be sure to use `\MonotypeMR` *before* using `\MTPsizes`.

X. Designing Your Own Formats

This section contains technical information that will be needed by people who are designing their own formats, where, for example, there might be a command `\tenpoint` for setting most of the text, but a `\ninepoint` command for footnotes or quoted material.

In addition to declaring fonts, and assigning them to various families, your style file should include the first part of the macro file `mtp.tex`, while the second part should be ignored (detailed comments appear in `mtp.tex`). If you are using the bold and heavy fonts, then you will also need the first part of the macro file `mtpb.tex` and `mtph.tex`.

1. Sections **II.2** and **VIII.3** have already discussed roman fonts in formulas (assignments to `\textfont0`, `\scriptfont0`, and `\scriptscriptfont0`).

2. The *MathTimeProfessional* fonts containing the italic letters for math are called

`mtmit`	*MathTime* math italic text	(or ten point)
`mtmis`	*MathTime* math italic small	(or seven point)
`mtmif`	*MathTime* math italic fine	(or five point)

(The actual PostScript names for the fonts are `MTMIT`, `MTMIS`, and `MTMIF`.)

The macro file `mtp.tex` loads `mtmit` at 10 points, `mtmis` at 7 points, and `mtmif` at 5.5 points, giving them unconventional names like ‘`\mtmit at 10pt`’ (which can only be created using `\csname...\endcsname`); then it sets `\textfont1` to be ‘`\mtmit at 10pt`’, `\textfont2` to be ‘`\mtmis at 7pt`’, and `\textfont3` to be ‘`\mtmif at 5.5pt`’.

Naturally, one could simply do something more conventional like

```
\font\tenmtmi=mtmit at 10pt
\font\sevenmtmi=mtmis at 7pt
\font\fivetmi=mtmif at 5.5pt
```

```
\font\ninemtmi=mtmit at 9pt
...
```

and then let `\tenpoint` specify

```
\textfont1=\tenmtmi
\scriptfont1=\sevenmtmi
...
```

while `\ninepoint` will specify

```
\textfont1=\ninemtmi
...
```

(This all assumes that the second part of `mtp.tex` is not operative.)

It is also important to identify the `\skewchar` of the `mtmi...` fonts, which is `45` (unlike the `cmmi...` fonts, which have a `\skewchar` of `'177`). So one really needs something like

```
\font\tenmtmi=mtmit at 10pt
\skewchar\tenmtmi=45
...
```

3. Similarly, the fonts corresponding to the `cmsy...` fonts are `mtsyt`, `mtsys`, and `mtsyf`, all with a `\skewchar` of `48`.

So after declaring

```
\font\tenmtsy=mtsy at 10pt
\skewchar\tenmtsy=48
...
```

`\tenpoint` might specify

```
\textfont2=\tenmtsy
...
```

IN ADDITION, you should add a line like

```
\usingMTPsizes{10pt}{7pt}{5.5pt}
```

to provide proper redefinitions of certain constructions from `plain` TeX (and $\mathcal{AMS}$-TeX, if it is being used). This line should come *after* the assignments for `\textfont2`, `\scriptfont2`, and `\scriptscriptfont2` in the definition of `\tenpoint` (and there should be a similar line in the definition of `\ninepoint`).

4. The *MathTimeProfessional* extension fonts are more complicated. Instead of a single font, like `cmex10`, there are four fonts, `mtexa`, `mtexe`, `mtexf`, and `mtexg`, as well as two additional fonts `mtxl` and `mtxxxl` [sic]. The font `mtexe` should be loaded at twice the size of `mtexa`, the font `mtexf` should be loaded at four times that size, and the font `mtexg` should be loaded at eight times that size; the font `mtxl` should be loaded at the same size as `mtexa`, and `mtxxxl` at twice that size. Thus, you might declare

```
\font\tenmtexa=mtexa at 10pt
\font\tenmtexe=mtexe at 20pt
\font\tenmtexf=mtexf at 40pt
\font\tenmtexg=mtexg at 80pt
\font\tenmtxl=mtxl at 10pt
\font\tenmtxxxl=mtxxxl at 20pt
```

In this case, `\tenpoint` would merely specify

```
\textfont3=\tenmtexa
\scriptfont3=\tenmtexa
\scriptscriptfont3=\tenmtexa
```

As usual, the same extension font, `\tenmtexa`, is used as the `\textfont`, `\scriptfont`, and `\scriptscriptfont` for `\fam3`; the fonts `\tenmtexe`, `\tenmtef`, and `\tenmtexg`, as well as `\tenmtxl` amnd `\tenmtxxxl`, are *not* assigned to any family at all.

IN ADDITION, however, you need to indicate that the fonts `\tenmtexa`, . . . ,`\tenmtexg`, `\tenmtxl` and `\tenmtxxxl` form an appropriate sextet by adding the line

```
\usingMTPextensions
  \tenmtexa \tenmtexe \tenmtexf \tenmtexg \tenmtxl \tenmtxxxl
```

This provides the proper information to the macros that define the commands `\LEFTRIGHT`, `\SQRT`, `\XL`, etc., as well as a few other details. Notice that, unlike `\usingMTPsizes`, the arguments for `\usingMTPextensions` are the actual names that you use for the four extension fonts, not their point sizes (braces aren't necessary around these arguments, since they are each single tokens).

Naturally, `\usingMTPextensions` might have different arguments within the definition of `\ninepoint`, etc.

5. The bold Times fonts that are produced by `\mbf` in math formulas are the *MathTime* 'mathbold' fonts `mtmbt`, `mtmbs`, and `mtmbf`, with a `\skewchar` of `32`, and they form a new family `\mbffam`. So after declaring

```
\font\tenmtmb=mtmbt at 10pt
\skewchar\tenmtmb=32
...
```

`\tenpoint` might specify

```
\textfont\mbffam=\tenmtmb
...
```

6. The fonts containing the bold italic letters for math are called `mtbmit`, `mtbmis`, and `mtbmif` (the actual PostScript names are `MTBMIT`, `MTBMIS`, and `MTBMIF`), all with a `\skewchar` of `45`.

In order for the definitions in `mtpb.tex` to work with your fonts, it is assumed that they are in family `\fam\mtbmi@`. So after assigning font names like

```
\font\tenmtbmit=mtbmit
```

you could either say

```
\newfam\mtbmi@
\textfont\mtbmi@=\tenmtbmit
```

or something like

```
\newfam\mtbfam
\textfont\mtbfam=\tenmtbmit
```

and then

```
\def\mtbmi@{\mtbfam}
```

In either case, it is also necessary to add the line

```
\edef\mtbmi@@{\hexnumber@\mtbmi@}
```

This line [note carefully the `\edef`] makes `\mtbmi@@` stand for the hex number of the family `\mtbmi@` (the command `\hexnumber@` is defined near the beginning of `mtpb.tex`).

Similarly, there are the bold symbol fonts `mtbsyt`, `mtbsys`, and `mtbsyf`, with a `\skewchar` of `48`, and they should be in the family `\mtbsy@`, with an appropriate line to define `\mtbsy@@`.

There is only one bold extension font `mtbexa`, and it should be assigned to (all three fonts of) the family `\mtbex@`, with an appropriate line to define `\mtbex@@`.

Although there actually are fonts `mthmit`, `mthmis`, and `mthmif`, it is unnecessary to use them, or to reserve a family for them. It is only necessary to provide a family for the heavy symbol fonts `mthsyt`, `mthsys`, and `mthsyf`, with a `\skewchar` of `48`. They should go in the family `\mthsy@`, with a line defining `\mthsy@@`. Similarly, the extension font `mthexa` should be assigned to the family `\mthex@`, with an appropriate line to define `\mthex@@`.

7. The Computer Modern "calligraphic" letters actually reside on the various `cmsy...` fonts, which aren't going to be used for math formulas. In order to make use of these letters, first add the line

```
\usecal
```

This will define a family `\Calfam` where the fonts for the uppercase calligraphic letters will reside and define `\cal` to mean `\fam\Calfam` (and properly define the command `\Cal` if $\mathcal{AMS}$-TEX is being used). Thus, after declaring

```
\font\tencal=cmsy10
\skewchar\tencal=48
```

```
\font\sevencal=cmsy7 at 7pt
\skewchar\sevencal=48
\font\fivecal=cmsy5 at 5.5pt
\skewchar\fivecal=48
```

`\tenpoint` can specify

```
\textfont\Calfam=\tencal
\scriptfont\Calfam=\sevencal
\scriptscriptfont\Calfam=\fivecal
```

8. Finally, `mtp.tex` provides a command `\useoldnos` that enables you to produce old style numbers. Since these old style numbers can contain commas and periods (for decimal points), both the text font and its companion font with the old style digits might be needed.

For your text font, you might be using a Computer Modern Roman font, or you might be using a PostScript font that has an expert font with old style digits in the appropriate ASCII positions.

If you are using `plain` TEX, so that you are using the Computer Modern Roman font `cmr10`, which happens to be named `\tenrm`, the companion font with the old style digits is `cmmi10`, which happens to be named `\teni`.

The command

```
\useoldno\tenrm\teni
```

will then define `\oldnos` so that `\oldnos{3,141.59}` produces 3,141.59. Notice that this `\oldnos` command is not a font change, but a control sequence with an argument; it is meant to be used only in text, not in math mode.

If you are using a Computer Modern roman font, but with some other style than `plain` TEX, then you will need to know explicitly the name being used for this roman font. And, of course, you will also need to know the name for the corresponding font with the old style digits; you might even have to declare this corresponding font yourself, since it might not be used in this style.

Similarly, if you are using a PostScript font ‘`Rhymes`’ for your text font and it has an expert font '`Rimes`' with old style digits in the appropriate ASCII positions, then you have probably declared something like

```
\font\tenRhymes=Rhymes at 10pt
\font\tenRimes=Rimes at 10pt
```

and you just have to add the line

```
\useoldnos\tenRhymes\tenRimes
```

to define an `\oldnos` command.

(Actually,

```
\useoldnos\tenRimes\tenRimes
```

will work just as well, using the period and comma from `\tenRimes` instead of from `\tenRhymes`.)