MathTıme™ *Professional II*

PostScript fonts for typesetting mathematics with TEX

The *MathTıme*™ *Professional* Fonts

Introduction

The *MathTımeProfessional* 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

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

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 *MathTımeProfessional* 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, with “Times-italic-like” letters; they also allow other special constructions that improve the results previously obtainable with TeX:

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

**MathTıme* 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-heights will match. The designer of the *MathTimeProfessional* fonts likes 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 *MathTimeProfessional* fonts, like the Computer Modern fonts, retain one important design feature that has generally been discarded in the PostScript world: the characters used for superscripts and for second-order superscripts come from specially designed fonts, so that instead of a formula like C^{xy} with cramped, rather spindly, superscripts that are merely 70% reductions of x, y, we get the more pleasing formula C^{xy}, and instead of

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

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

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

Of course, all the *MathTimeProfessional* 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 *MathTimeProfessional* package involves a multitude of files, which must be placed in the proper directories and subdirectories. Installation of the *MathTimeProfessional* 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 *MathTimeProfessional* Installation document.

The *MathTımePro* “Lite” Fonts

The complete version of *MathTımeProfessional II* contains a large number of fonts, providing bold versions of the standard fonts, numerous additional symbols, and a wide array of additional alphabets. A special “lite” version, containing only the basic fonts (sufficient for most standard mathematical typesetting) is also available, and comes free with PCTEX Publisher.

A simple LATEX option can be used for the lite version, while `plain` TEX users can simply dispense with any of the special ‘`\load...`’ commands that are used when the other fonts are available. We will first describe the features of the lite version, which are simply a subset of the extended features of the complete version.

I. *MathTımeProfessional* and LATEX

A special LATEX macro package `mtpro2`, designed by Walter Schmidt, allows the *MathTımeProfessional II* fonts to be used with LATEX:

```
\usepackage[⟨options⟩]{mtpro2}
```

with one of the many possible options being ‘`lite`’.

The documentation file `mtpro2.pdf` provides all details (as well as a complete technical description of the LATEX implementation). Consequently, the current documentation is devoted almost entirely to the use of the *MathTıme Professional II* fonts with `plain` TEX (with just a few asides concerning LATEX).

II. *MathTımeProfessional* and `plain` TEX

1. To use the *MathTımeProfessional II* fonts with `plain.tex`, put

```
\input mtp2
```

at the beginning of your file, causing TEX to read in the file `mtp2.tex`. If your file also has the line `\input amstex`, this should *precede* the line containing the command `\input mtp2`. [See section **XVI** for differences between usage for the original *MathTimeProfessional* fonts and version *II*.]

2. The macro file `mtp2.tex` causes the *MathTimeProfessional* 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—so the choice of text font has to be made separately. 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 mtp2`.

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}
```

3. Normally, `mtp2.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 `\MTP` command. For example, you could multiply the sizes by approximately 110% by typing

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

See also section **VIII**.

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. In `plain` TEX, the slanted upper-case Greek letters $\varGamma$, $\varDelta$, … are obtained by typing `\mit\Gamma`, etc. But that won't work with the *MathTimeProfessional* 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 '`mtpro2`' package that causes `\Gamma`, … , to produce slanted letters automatically, although upright letters can also be obtained—see **3**.

2. In addition to $\varkappa$, there is $\varbeta$ (`\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 $\vardelta$. Although the `\partial` symbol ∂ looks quite similar, it doesn't slant as much, and the lower part isn't quite as tall; the variant $\vardelta$ has been included on the fonts only because all the various Greek alphabets (regular, bold, etc.) specified for mathematics in the Unicode standard include this variant (perversely called 'partial'). Of course, you could always use `\vardelta` in place of `\partial` if you prefer.

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

3. The *MathTimeProfessional* fonts also include upright lowercase Greek letters $\upalpha$, $\upbeta$, $\upgamma$, $\upvarkappa$, $\upvarbeta$, … , 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 **IX**) might be found useful.

4. `plain` TEX users should note that `mtp2.tex` normally makes `\cal` undefined, since the "calligraphic" letters actually appear on the Computer Modern font that contains the math symbols, while the corresponding *MathTime*

Professional font has other characters in those places. In fact, you will probably prefer to use one of the various “script” fonts provided by the complete *MathTimeProfessional* collection. But if you want to use the Computer Modern calligraphic letters instead of, or in addition to, one of the *MathTime Professional* script fonts, you can put `\useCMcal` in your file, and `\cal` will change fonts in math mode to these calligraphic letters, while `\Cal` will be a control sequence with an argument to produce them. Or you might prefer `\useEulercal` if you want `\cal` and `\Cal` to provide the Euler script font, or `\useLucidacal` if you want the Lucida calligraphic font (this font has lowercase letters, but does not provide different designs for different sizes).

Even more generally, if you had some script font, say `scripty`, you could type something like

```
\useNewcal{scripty}[.95]{scripty}[1.05]{scripty}[1.07]
```

where the `.95`, `1.05`, and `1.07` might be suitable magnification factors to make the heights of `scripty` match with the *MathTimeProfessional* fonts at ordinary sizes, in superscripts, and in second-order superscripts, respectively.

The command `\oldstyle` (and `\oldnos` in $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-TEX) are undefined for similar reasons. Many PostScript text fonts have a companion “expert” font that contains old style digits for that font in the appropriate ASCII positions, but `mtp2.tex` doesn’t attempt to provide commands to use these fonts (which you are unlikely to use in math mode in the first place).

IV. New Symbols

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

1. There is an $\hslash$ (`\hslash`) to accompany $\hbar$ (`\hbar`) [see also **11.** concerning the symbol $\hbar$], and there are

đ	`\dbar`
đ	`\updbar`
$\overset{\sim}{\rightarrow}$	`\simarrow`
&	`\varland` (variant logical and)
⌟	`\contraction`
$:=$	`\coloneq`
$=:$	`\eqcolon`
$\hat{=}$	`\hateq`
$\circ\!\!-\!\!\bullet$	`\circdashbullet`
$\bullet\!\!-\!\!\circ$	`\bulletdashcirc`

2. Some people like to have an italic z with a “swash” tail z. Putting the command `\zswash` in your file will cause the z to appear instead of z in your equations. The command `\zstraight` switches back to the standard z.

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

4. Three new symbols have been added for formulas like

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

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 $\mathbin{\scriptstyle\cup}$ (`\cupprod`) and $\mathbin{\scriptstyle\cap}$ (`\capprod`).

5. There are also corresponding large operators `\bigcupprod` ($\bigcup$ and $\displaystyle\bigcup$) and `\bigcapprod` ($\bigcap$ and $\displaystyle\bigcap$). Moreover, the binary operator $*$ (`\ast`, or simply the `*` key) has the corresponding large operator `\bigast` ($\mathop{\ast}$ and $\displaystyle\mathop{\ast}$). In addition, the new symbol & (`\varland`) has the corresponding large operator `\bigvarland` ($\mathop{\&}$ and $\displaystyle\mathop{\&}$).

$$\bigcup_{i=1}^{k} \alpha_i = \alpha_1 \mathbin{\scriptstyle\cup} \cdots \mathbin{\scriptstyle\cup} \alpha_k \qquad \bigcap_{i=1}^{k} \alpha_i = \alpha_1 \mathbin{\scriptstyle\cap} \cdots \mathbin{\scriptstyle\cap} \alpha_k$$

$$\mathop{\ast}\limits_{i=1}^{k} a_i = a_1 * \cdots * a_k \qquad \mathop{\&}\limits_{i=1}^{k} P_i = P_1 \mathbin{\&} \cdots \mathbin{\&} P_k$$

6. Accompanying `\int` and `\oint`, there are now

$\iint$	$\displaystyle\iint$	`\iint`	$\cwoint$	$\displaystyle\cwoint$	`\cwoint` (clockwise `\oint`)
$\iiint$	$\displaystyle\iiint$	`\iiint`	$\awoint$	$\displaystyle\awoint$	`\awoint` (anticlockwise `\oint`)
$\oiint$	$\displaystyle\oiint$	`\oiint`	$\cwint$	$\displaystyle\cwint$	`\cwint` (clockwise `\int`)
$\oiiint$	$\displaystyle\oiiint$	`\oiiint`	$\barint$	$\displaystyle\barint$	`\barint`
			$\slashint$	$\displaystyle\slashint$	`\slashint`

7. The operators $\sum$, $\displaystyle\sum$; $\prod$, $\displaystyle\prod$; and $\coprod$, $\displaystyle\coprod$, have slanted versions $\slsum$, $\displaystyle\slsum$; $\slprod$, $\displaystyle\slprod$; and $\slcoprod$, $\displaystyle\slcoprod$. The command `\sloperators` makes `\sum`, `\prod`, and `\coprod` produce these slanted versions, and `\upoperators` changes back to the upright versions. Whichever convention you use, you can always use `\slsum`, . . . to get the slanted versions and `\upsum`, . . . to get the upright versions.

8. `\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 $\mathring{A}$.

9. 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 $\dotup{\Gamma}$ and $\ddotup{\Gamma}$ with `\dotup` and `\ddotup`. (If you are using $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX there are also `\Dotup` and `\Ddotup` for double accents.)

10. The math accents $\hat{}$ (`\hat`), $\check{}$ (`\check`), $\tilde{}$ (`\tilde`), and $\bar{}$ (`\bar`) have slightly wider versions

$\what{}$ (`\what`), $\wcheck{}$ (`\wcheck`), $\wtilde{}$ (`\wtilde`), $\wbar{}$ (`\wbar`)

[with corresponding `\Wcheck`, etc., if you are using $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX], and then versions that are slightly wider still:

$\wwhat{}$ (`\wwhat`), $\wwcheck{}$ (`\wwcheck`), $\wwtilde{}$ (`\wwtilde`), $\wwbar{}$ (`\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

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

11. Numerous standard TeX symbols were traditionally constructed from two or more characters. For example, $\hbar$ (`\hbar`) was made from h and the bar accent $\bar{\ }$, and $\hookleftarrow$ was made from the $\leftarrow$ and a $\rhook$ 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. In addition, $\dddot{\ }$ (`\dddot`) and $\ddddot{\ }$ (`\ddddot`) from AMS-TeX have been added (there are also `\dddotup` and `\ddddotup`, and if you are using 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 used when the additional *MathTimeProfessional* fonts are used (see **X.5**).

$\nless$	`\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`
$\not\sim$	`\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. The additional fonts also contain a character `\nsim` ($\nsim$), but this is different from `\notsim`.

[The `\not` symbol is still on the fonts, as are the hook ᶜ (`\lhook`) and the hook ᵓ (`\rhook`), so one can imitate the definitions in `plain.tex` to build a longer hooked arrow $\hookleftarrow$ 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 ǀ (`\mapstochar`), which is just a small vertical line with a tiny bit of horizontal line to the right, to insure that there are no gaps in the construction

```
        \mapstochar\relbar\joinrel\relbar\joinrel
           \relbar\joinrel\rightarrow
```

used to create this $\longmapsto$.]

12. 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 **IX**).

V. Bold Roman Letters

In a math formula, `\bf` gives the bold font that is used in text (in $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-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 Times fonts (as on page 4), then you will get the Times bold font.

But there is an additional `\mbf` command that selects "Times-bold-like" letters that work better in math formulas. 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.

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

Although `\mbf` is generally the best way to get bold letters in math, if you wanted something like **Major**(X), it would be better to use `{\bf Major}` [or `\bold{Major}` in AMS-TeX, or `\mathbf{Major}` with LaTeX] instead of `\mbf{Major}`, because the spacing with `\mbf` isn't meant for text, so you would get $\mathbf{Major}(X)$.

VI. The Big Differences

1. The formula

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

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

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

```
$$\PARENS{\matrix...}$$
```

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. Similarly, instead of

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

with an extensible radical sign, you 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

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

3. There are also two special "delimiters" `\lcbrace` and `\rcbrace` that can't be used with `\left` and `\right`, but can be used after `\LEFTRIGHT`. These give individually designed vertical curly braces up to 4 inches high, which you might prefer to the usual extensible braces. The commands

`\left\{...\right\}` and `\LEFTRIGHT\lcbrace\rcbrace{...}`

give output like

$$\left\{\rule{0pt}{1.2in}\quad\right\} \qquad \text{and} \qquad \left\{\rule{0pt}{1.3in}\quad\right\}$$

To go along with this, the `\cases` construction has an analogous `\ccases` construction that gives curly braces:

$$f(x) = \begin{cases} a & x < -1 \\ b & -1 \le x < 0 \\ c & 0 \le x < 1 \\ d & x \ge 1 \end{cases}$$

(If you are using $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX, then `\ccases` uses the usual $\mathcal{A}\mathcal{M}\mathcal{S}$-TeX syntax, `\ccases...\endcases`, with `\\` used to separate lines instead of `\cr`.)

4. 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.

If, for some reason, you need double `\wide...` accents, you may be disappointed to find that

```
\widehat{\widehat{A+B+C+D+E+F+G }}
```

gives

$$\widehat{\widehat{A+B+C+D+E+F+G}}$$

with the top `\widehat` seemingly too high (its base is at the level of the top of the lower `\widehat`).

So there is also

```
\widehatdown{⟨dimen⟩}{...}
```

to move a `\widehat` down (and similarly for the `\widetilde` and the `\widecheck` accents). For example,

```
\widehatdown{2pt}{\widehat{A+B+C+D+E+F+G}}
```

produces

$$\widehat{\widehat{A+B+C+D+E+F+G}}$$

5. You can also get individually designed curly underbraces and overbraces up to 4 inches wide. Compare

`\underbrace{A_1+\cdots+A_i+\cdots+A_n}` $\underbrace{A_1+\cdots+A_i+\cdots+A_n}$

`\undercbrace{A_1+\cdots+A_i+\cdots+A_n}` $\underbrace{A_1+\cdots+A_i+\cdots+A_n}$

If you are using AMS-TeX, the commands `\undersetcbrace...\to{...}` and `\oversetcbrace...\to{...}` are also available.

6. 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 AMS-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

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

with the root 3 in '`\scriptstyle`' instead (you can experiment yourself with the necessary `\uproot` and `\leftroot` amounts). You might even want to use '`\textstyle3`' for a very tall root. (In AMS-TeX you can type `\ssize` instead of `\scriptstyle` and `\tsize` instead of `\textstyle`.)

(e) [See section **XVI.1** regarding changes from previous versions.] Small discrepancies might arise when you have large constructions like `\LEFTRIGHT(]{`⟨formula⟩`}` because delimiters like `|` and `]` are simply created as extensible characters, growing at a uniform rate, while parentheses and various other delimiters grow faster for larger formulas. So in such cases you may need to replace ⟨formula⟩ with

`\vcorrection{`⟨dimen⟩`}`⟨formula⟩

to artificially increase its vertical size to ⟨dimen⟩, thereby forcing the delimiter] to be larger.

VII. Extra Large Operators

[See section **XVI.2** regarding changes from previous versions.]

In displayed formulas like

$$\sum_{i\notin I} \frac{\displaystyle\int_{-\infty}^{\infty} f(\alpha_i x)\,dx + 1}{\displaystyle\oint\limits_{C} f(\beta_i z)\,dz - 1} \qquad \text{and} \qquad \int_{-\infty}^{\infty} \sqrt{\frac{\sum\limits_{i=1}^{\infty} f(\alpha_i x)}{\gamma + \sum\limits_{i=1}^{\infty} f(\beta_i x)}}\,dx$$

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

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

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

$$\sum_{i=1}^{\infty}$$

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

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

$$\sum_{i=1}^{\infty}$$

As usual, you can also add `\nolimits` after the `\sum` if you don't want the subscript and superscript set as "limits". And, in combinations like `\xl\int` where they are normally not set as limits, you can add `\limits` if you do want them set that way.

The commands `\xl` and `\XL` can be applied to all ‘large operators’, including those in sections **IV.5–7**; 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.

The command `\xl\sum` will produce an extra large $\sum$ or $\sum$ depending on which convention you are using, and `\xl\slsum` and `\xl\upsum` will specifically choose slanted or upright versions, etc.

$\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX has the special construction `\Sb...\endSb` to produce stacked lower limits,

```
\sum \Sb i<j\\k<l\endSb ^\infty
```

$$\sum_{\substack{i<j\\k<l}}^{\infty}$$

and a similar construction `\Sp...\endSp` for stacked upper limits. If you are using $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX, then these commands can also be used after combinations like `\xl\sum`.

VIII. Designing Your Own Formats

[See section **XVI.3** regarding changes from previous versions.]

Section **II.3** explained how to change the sizes of the *MathTimeProfessional* fonts, but you might want to have a more complicated format, where, for example, there might be a command `\tenpoint` for setting most of the text, but a `\ninepoint` command for footnotes or quoted material.

1. Section **II.2** has already discussed roman fonts in formulas (assignments to `\textfont0`, `\scriptfont0`, and `\scriptscriptfont0`). You could also declare fonts like

```
\font\ninetimes=times at 9pt
...
```

and then place

```
\textfont0=\tentimes
...
```

inside the definition of `\tenpoint` and

```
\textfont0=\ninetimes
...
```

inside the definition of `\ninepoint`.

2. The very last line of `mtp2.tex` is

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

This command loads the various fonts needed for using the lite version, with the default choice of sizes for text size, superscripts, and second-order superscripts (see **3.** for details), and assigns the appropriate fonts for letters (and bold letters), symbols, and extension characters. Another `\MTP` command, with different arguments, will simply change all the assigned fonts to new ones. Thus, one could define `\tenpoint` to mean

```
\textfont0=...
\scriptfont0=...
\scriptscriptfont0=...
\MTP{10pt}{7pt}{5.5pt}
```

and define `\ninepoint` to mean something like

```
\textfont0=...
\scriptfont0=...
\scriptscriptfont0=...
\MTP{9pt}{6.5pt}{5pt}
```

When you are using the complete version, and using various '`\load...`' commands, these automatically load fonts at the same sizes as the latest `\MTP` command. So, for example, if you are using the bold math fonts, with the `\loadbm` command (section **IX.2**) it is only necessary to add `\loadbm` after the `\MTP` in the definition of `\tenpoint` and also add `\loadbm` after the `\MTP` in the definition of `\ninepoint`.

By the way, the sizes used after the most recent `\MTP` command are stored in `\tMTPsize`, `\sMTPsize`, `\fMTPsize`, and the command `\loadbm` is actually an abbreviation for

```
\loadbmAt{\tMTPsize}{\sMTPsize}{\fMTPsize}
```

so you could even vary the sizes of additionally loaded fonts (not that we are actually recommending such a procedure). For example, if you wanted your bold math symbols to be extra striking, in your definition of `\tenpoint` you could have

```
\MTP{10pt}{7pt}{5.5pt}
\loadbmAt{11pt}{9pt}{7pt}
```

Similarly, the command `\useCMcal` of section **III.4** is an abbreviation for `\useCMcalAt{\tMTPsize}{\sMTPsize}{\fMTPsize}`.

3. The font used for the italic letters at text size is `mt2mit` at 10 points.* Rather than being given an arbitrary name, like `\tenmi`, this font is named in a systematic, but somewhat strange-looking, way by a control sequence having letters, numerals, and even spaces in its name (accessible only by using the `\csname...\endcsname` construction):

```
\mt2mit at 10pt
```

If your log file contains an `Overfull box` message, or a listing for some other reason, this font will thus show up in this informative form.

Similarly for the fonts '`\mt2mis at 7pt`' and '`\mt2mif at 5.5pt`', which are assigned to `\scriptfont1` and `\scripscriptfont1`.

Likewise '`mt2syt at 10pt`' is assigned to `\textfont2`, etc. Moreover, '`mt2exa at 10pt`' is used for the extension font, while other similarly named fonts are installed for use with the new constructions like `\LEFTRIGHT` and `\xl`.

*The PostScript name of the font `mt2mit` is actually `MTP2MIT`, all capitals.

Additional Fonts in the *MathTimePro* Complete Version

The additional fonts in the complete *MathTimeProfessional II* version provide:

- Bold versions of the basic fonts, as well as 'heavy' (extra-bold) versions of many symbols.

- Times-compatible versions of the various operators on the AMS's `msam` and `msbm` fonts, together with a few additional related symbols. These are also available in a bold version, as well as a heavy one.

- Several different complete "blackboard bold" fonts, with both uppercase and lowercase letters, as well as numerals.

- A script and bold script font, as well as a variant "curly" font.

- A (German) Fraktur font, and a bold version.

- Two "blackboard bold" fonts that have slanted letters and numerals, and all the Greek letters, both slanted and upright.

As explained in the LaTeX documentation, if you have these fonts, then you can use `mtpro2.sty` without the '`lite`' option. The remainder of this document is for `plain` TeX users.

IX. Bold (*Italic*) Letters, Bold and Heavy Symbols

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 $\boldsymbol{n > 2}$.

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—how do you embolden a formula that already contains bold letters, 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 mtp2`). 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.

The `\boldmath` command **CANNOT** be used for only part of a formula; the combination `$A\boldmath B$` 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 $A''+\boldsymbol{A''}+\boldsymbol{A''^{2}}$. 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 $\boldsymbol{\dagger}$ (`\bmdagger`) and $\boldsymbol{\ddagger}$ (`\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` ($\boldsymbol{\upalpha}$), `\bmupbeta` ($\boldsymbol{\upbeta}$), `\bmupgamma` ($\boldsymbol{\upgamma}$), 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`	**(**	`\bmrparens`	**)**
`\bmlbrack`	**[**	`\bmrbrack`	**]**
`\bmvert`	**|**	`\bmVert`	**‖**

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'', `\hmoplus` gives $\boldsymbol{\oplus}$, 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 at times, as discussed below.

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 that the `\boldmath` command can't be used for only part of a formula).

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 $\sqrt{x} + \boldsymbol{\sqrt{\mathnormal{x}}}$ 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 `mtp2.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. 11**, for example, an extra long bold hooked arrow $\hookleftarrow$. Imitating the definition in `plain.tex` is not so easy now, because there is no special name for the bold hook $\rhook$. 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 [and be sure to leave a space after the second `@` if the next character is one of the hex numbers `A`, . . . , `F`]). 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 $\hookleftarrow$, you will have to work a bit harder, because `mtp2.tex` makes no use of heavy math italic fonts.* To compensate for this, $\rhook$ and similar symbols appear on the heavy symbol fonts. The last part of the font table for these fonts looks like

	'0	'1	'2	'3	'4	'5	'6	'7	
'34x	$\leftharpoonup$	$\leftharpoondown$	$\rightharpoonup$	$\rightharpoondown$	$\lhook$	$\rhook$	(	)	"Ex
'35x	.	,	<	/	>	★	∂	♭	
'36x	♮	♯	$\smile$	$\frown$	ℓ	℘	!	?	"Fx
'37x	[	]	†	‡					
	"8	"9	"A	"B	"C	"D	"E	"F	

*Compare the footnote on page 24.

The heavy hook › occurs in position `"E5`, and the hex numbers of the family for the heavy symbol fonts and the heavy extension font are stored in `\mthsy@@` and `\mthex@@`, so the extra long heavy hooked arrow was obtained as

```
\hmlongleftarrow\mathrel{\mkern-3mu}\mathchar"3\mthsy@@ E5
```

X. New Symbols

The `\loadbm` command loads 3 different bold italic fonts, for the 3 different sizes used in formulas, 3 different bold symbol fonts, and a font for bold large operators. The `\loadhm` command also loads more than a triplet of fonts.

All the additional `\load...` commands are different: they each load exactly 3 fonts, for the 3 different sizes used in formulas. There are so many of these that you might get an error message if you try to use too many of them at once. If that happens, see section **XV**. A summary of all the `\load...` commands can be found right before that section.

The first of these new commands,

```
    \loadsya
```

loads fonts containg additional symbols similar to those on the AMS fonts and defines names for all the new symbols.

1. The symbols

✓	`\checkmark`	✠	`\maltese`
®	`\circledR`	¥	`\yen`

are normally used outside of math mode, but they can be used in math mode also.

Three other special letters for math mode only are the Hebrew

ב	`\beth`	ג	`\gimel`
ד	`\daleth`		

There are also four special delimiters (which occur in only one size):

$\ulcorner$	`\ulcorner`	$\urcorner$	`\urcorner`
$\llcorner$	`\llcorner`	$\lrcorner$	`\lrcorner`

The remaining new symbols fall into various groups.

2. Most of the new symbols are binary operators or relations, but first we have a miscellaneous group of **ordinary symbols**.

$\backprime$	`\backprime`	$\varnothing$	`\varnothing`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`
$\square$	`\square`	$\blacksquare$	`\blacksquare`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`
$\Diamond$	`\Diamond`		
$\circledS$	`\circledS`	$\bigstar$	`\bigstar`
$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\nexists$	`\nexists`	$\complement$	`\complement`
$\mho$	`\mho`	$\eth$	`\eth`
$\Finv$	`\Finv`	$\Game$	`\Game`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`

Remember that the plain $\angle$ (`\angle`) already appears on the *MathTimeProfessional* basic fonts. On the other hand, $\Diamond$ (`\Diamond`) doesn't appear on the AMS fonts, although a similar symbol occurs on the LaTeX symbol font.

3. Next come **binary operators**.

$\dotplus$	`\dotplus`	$\smallsetminus$	`\smallsetminus`
$\ltimes$	`\ltimes`	$\rtimes$	`\rtimes`
$\Cap$	`\Cap`, `\doublecap`	$\Cup$	`\Cup`, `\doublecup`
$\leftthreetimes$	`\leftthreetimes`	$\rightthreetimes$	`\rightthreetimes`
$\barwedge$	`\barwedge`	$\veebar$	`\veebar`
$\doublebarwedge$	`\doublebarwedge`		
$\curlywedge$	`\curlywedge`	$\curlyvee$	`\curlyvee`
$\boxplus$	`\boxplus`	$\boxminus$	`\boxminus`
$\boxtimes$	`\boxtimes`	$\boxdot$	`\boxdot`

continued on next page

⊝	`\circleddash`	⊛	`\circledast`
⊚	`\circledcirc`	⋇	`\divideontimes`
▪	`\centerdot`	⊺	`\intercal`

`\smallsetminus` is actually just a synonym for `\setdif` on the *MathTıme Professional* basic fonts.

4. Binary relations.

In the list below, note that ⊏ (`\sqsubset`) and ⊐ (`\sqsupset`) are new symbols, while the more complicated ⊑ (`\sqsubseteq`) and ⊒ (`\sqsupseteq`) already exist on the basic fonts!

Note also that ⌣ (`\smallsmile`) and ⌢ (`\smallfrown`) are different from the symbols ∪ (`\cupprod`) and ∩ (`\capprod`), and that the old ⊨ (`\models`) is different from ⊨ (`\vDash`).

≦	`\leqq`	≧	`\geqq`
⩽	`\leqslant`	⩾	`\geqslant`
⪕	`\eqslantless`	⪖	`\eqslantgtr`
≲	`\lesssim`	≳	`\gtrsim`
⪅	`\lessapprox`	⪆	`\gtrapprox`
≊	`\approxeq`		
⋖	`\lessdot`	⋗	`\gtrdot`
⋘	`\lll`, `\llless`	⋙	`\ggg`, `\gggtr`
≶	`\lessgtr`	≷	`\gtrless`
⋚	`\lesseqgtr`	⋛	`\gtreqless`
⪋	`\lesseqqgtr`	⪌	`\gtreqqless`
⇅	`\coloneq`	⇵	`\hateq`
≑	`\doteqdot`, `\Doteq`	≖	`\eqcirc`
≒	`\fallingdotseq`	≓	`\risingdotseq`

continued on next page

$\circeq$	`\circeq`	$\triangleq$	`\triangleq`
$\backsim$	`\backsim`	$\thicksim$	`\thicksim`
$\backsimeq$	`\backsimeq`	$\thickapprox$	`\thickapprox`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`
$\Subset$	`\Subset`	$\Supset$	`\Supset`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`
$\precsim$	`\precsim`	$\succsim$	`\succsim`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`
$\vartriangleleft$	`\vartriangleleft`	$\vartriangleright$	`\vartriangleright`
$\trianglelefteq$	`\trianglelefteq`	$\trianglerighteq$	`\trianglerighteq`
$\blacktriangleleft$	`\blacktriangleleft`	$\blacktriangleright$	`\blacktriangleright`
$\vDash$	`\vDash`	$\Vdash$	`\Vdash`
$\Vvdash$	`\Vvdash`		
$\smallsmile$	`\smallsmile`	$\smallfrown$	`\smallfrown`
$\shortmid$	`\shortmid`	$\shortparallel$	`\shortparallel`
$\bumpeq$	`\bumpeq`	$\Bumpeq$	`\Bumpeq`
$\therefore$	`\therefore`	$\because$	`\because`
$\between$	`\between`	$\pitchfork$	`\pitchfork`
$\varpropto$	`\varpropto`	$\backepsilon$	`\backepsilon`

5. Negated relations.

Several negated relation symbols already appear, in identical form, on the basic *MathTimeProfessional* fonts, so in addition to their standard AMS names, their names from the basic font package also appear, in brackets. It should be noted, however, that $\not\sim$ (`\notsim`) from the basic fonts is definitely different from $\nsim$ (`\nsim`).

(It might also be mentioned that the final four symbols, `\nsqsubset`, `\nsqsubseteq`, `\nsqsupset`, and `\nsqsupseteq`, don't actually exist on the AMS fonts.)

$\nless$	\nless [\notless]	$\ngtr$	\ngtr [\notgr]
$\nleq$	\nleq [\notleq]	$\ngeq$	\ngeq [\notgeq]
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq
$\lneq$	\lneq	$\gneq$	\gneq
$\lneqq$	\lneqq	$\gneqq$	\gneqq
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq
$\lnsim$	\lnsim	$\gnsim$	\gnsim
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox
$\nprec$	\nprec [\notprec]	$\nsucc$	\nsucc [\notsucc]
$\npreceq$	\npreceq [\notpreceq]	$\nsucceq$	\nsucceq [\notsucceq]
$\precneqq$	\precneqq	$\succneqq$	\succneqq
$\precnsim$	\precnsim	$\succnsim$	\succnsim
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox
$\nsim$	\nsim	$\ncong$	\ncong
$\nshortmid$	\nshortmid	$\nshortparallel$	\nshortparallel
$\nmid$	\nmid	$\nparallel$	\nparallel
$\nvdash$	\nvdash	$\nvDash$	\nvDash
$\nVdash$	\nVdash	$\nVDash$	\nVDash
$\ntriangleleft$	\ntriangleleft	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq [\notsubseteq]	$\nsupseteq$	\nsupseteq [\notsupseteq]
$\nsubseteqq$	\nsubseteqq	$\nsupseteqq$	\nsupseteqq
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq
$\varsubsetneqq$	\varsubsetneqq	$\varsupsetneqq$	\varsupsetneqq
$\not\sqsubset$	\nsqsubset	$\not\sqsupset$	\nsqsupset
$\not\sqsubseteq$	\nsqsubseteq [\notsqsubseteq]	$\not\sqsupseteq$	\nsqsupseteq [\notsqsupseteq]

6. Arrows.

In the list below, it should be noted that $\rightleftharpoons$ (`\rightleftharpoons`) is actually taken from the *MathTimeProfessional* basic fonts, while $\leadsto$ (`\leadsto`) appears in the LaTeX symbol font (it is also more common than $\rightsquigarrow$). Several vertical arrow and harpoon characters are not actually on the AMS fonts; be careful to distinguish `\updownarrows` from `\updownarrow`, etc.

The characters `\rarrowhead`, `\larrowhead`, and `\midshaft` (which are not given names in the AMS fonts) may be used to construct longer dashed arrows. For example

```
\mathrel{\midshaft\midshaft\midshaft\rarrowhead}
```

can be used to produce the arrow in the formula $A \dashrightarrow B$.

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow,`
			`\dasharrow`

$\larrowhead$	`\larrowhead`	$\rarrowhead$	`\rarrowhead`
$\midshaft$	`\midshaft`		
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`
$\undercurvearrowleft$	`\undercurvearrowleft`	$\undercurvearrowright$	`\undercurvearrowright`
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`
$\upuparrows$	`\upuparrows`	$\downdownarrows$	`\downdownarrows`
$\updownarrows$	`\updownarrows`	$\downuparrows$	`\downuparrows`

$\upharpoonleft$	`\upharpoonleft`	$\upharpoonright$	`\upharpoonright,`
			`\restriction`

$\downharpoonleft$	`\downharpoonleft`	$\downharpoonright$	`\downharpoonright`

continued on next page

⥣	`\upupharpoons`	⥥	`\downdownharpoons`
⥮	`\updownharpoons`	⥯	`\downupharpoons`
$\rightsquigarrow$	`\rightsquigarrow`	$\leftrightsquigarrow$	`\leftrightsquigarrow`
$\leadsto$	`\leadsto`	$\multimap$	`\multimap`

7. Negated arrows.

$\nleftarrow$	`\nleftarrow`	$\nrightarrow$	`\nrightarrow`
$\nLeftarrow$	`\nLeftarrow`	$\nRightarrow$	`\nRightarrow`
$\nleftrightarrow$	`\nleftrightarrow`	$\nLeftrightarrow$	`\nLeftrightarrow`

8. You can access the bold versions of the new symbols with `\loadbsya` and the heavy versions with `\loadhsya` (not particularly recommended for these rather complicated symbols).

(a) If you have the lines

```
\loadbm
\loadsya
\loadbsya
```

in your file, then

```
$\boldmath .   .   .  $
```

may be applied even when the formula contains the extra symbols. For example, `$\boldmath x<y\leqq z$` produces the formula $\boldsymbol{x < y \leqq z}$

(b) For individual symbols, instead of introducing a whole slew of new names, like '`\bmboxplus`', the single command `\bma` may be used before any of these additional symbols to produce the bold version. For example, `\bma\boxplus` yields $\boldsymbol{\boxplus}$. Similarly, `\hma\boxdot` yields $\boldsymbol{\boxdot}$.

Just as there aren't heavy versions of letters, the "alphabetical" symbols $\circledR$ and $\circledS$ don't have heavy versions, so `\hma\circledR` or `\hma\circledS` will simply give a "slug" ▮.

Additional Alphabets

In addition to all these symbols, the complete *MathTimeProfessional* collection contains several different alphabets for use in mathematics formulas.

XI. “Blackboard Bold”

[See section **XVI.4** regarding changes from previous versions.]

Two different varieties of “blackboard bold” alphabets are provided.

The first version, *MathTime* **h**oley **r**oman **b**old, is a “bold open” font, formed by hollowing out bold letters:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz0123456789

By contrast, the *MathTime* **b**lackboard **b**old font is the sort of alphabet that one might actually write on a blackboard:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz0123456789

Or you might prefer one of the dark versions, **h**oley **r**oman **b**old **d**ark:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz0123456789

or **b**lackboard **b**old **d**ark:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz0123456789

To use the **h**oley **r**oman **b**old font, first put

```
        \loadhrb
```

in your file.* This will load the appropriate fonts and define a control se-

*If this leads to an error message, see **XV**!

quence `\hrb` with one argument, so that `$\hrb a,\ldots,\hrb Z,\hrb0, \ldots,\hrb9$` gives $\mathbb{a},\ldots,\mathbb{Z},\mathbb{0},\ldots,\mathbb{9}$. You can also type `\hrb\imath` to get $\mathbb{\imath}$ and `\hrb\jmath` for $\mathbb{\jmath}$. Similarly,

```
\loadbb
```

will load the appropriate **b**lackboard **b**old fonts, and define a control sequence `\bb`, while `\loadhrbd` and `\hrbd` are used for the **h**oley **r**oman **b**old **d**ark fonts, and `\loadbbd` and `\bbd` are used for the **b**lackboard **b**old **d**ark fonts.

XII. Script

[See section **XVI.5** regarding changes from previous versions.]

The *MathTimeProfessional* Math Script font contains only letters, although a few letters have alternate versions, indicated in brackets (and there are still dotless i and j):

$$\mathscr{A}\mathscr{B}\mathscr{C}[\mathscr{C}]\mathscr{D}\mathscr{E}\mathscr{F}\mathscr{G}[\mathscr{G}]\mathscr{H}\mathscr{I}\mathscr{J}\mathscr{K}\mathscr{L}[\mathscr{L}]\mathscr{M}\mathscr{N}\mathscr{O}\mathscr{P}\mathscr{Q}[\mathscr{Q}]\mathscr{R}\mathscr{S}[\mathscr{S}]\mathscr{T}\mathscr{U}\mathscr{V}\mathscr{W}\mathscr{X}\mathscr{Y}[\mathscr{Y}]\mathscr{Z}[\mathscr{Z}]$$

$$\mathscr{a}\mathscr{b}\mathscr{c}\mathscr{d}\mathscr{e}\mathscr{f}\mathscr{g}\mathscr{h}\mathscr{i}\mathscr{j}\mathscr{k}\mathscr{l}\mathscr{m}\mathscr{n}\mathscr{o}\mathscr{p}\mathscr{q}\mathscr{r}[\mathscr{r}]\mathscr{s}\mathscr{t}\mathscr{u}\mathscr{v}\mathscr{w}\mathscr{x}\mathscr{y}\mathscr{z}[\mathscr{z}]$$

and the Bold Math Script is similar:

$$\boldsymbol{\mathscr{A}\mathscr{B}\mathscr{C}[\mathscr{C}]\mathscr{D}\mathscr{E}\mathscr{F}\mathscr{G}[\mathscr{G}]\mathscr{H}\mathscr{I}\mathscr{J}\mathscr{K}\mathscr{L}[\mathscr{L}]\mathscr{M}\mathscr{N}\mathscr{O}\mathscr{P}\mathscr{Q}[\mathscr{Q}]\mathscr{R}\mathscr{S}[\mathscr{S}]\mathscr{T}\mathscr{U}\mathscr{V}\mathscr{W}\mathscr{X}\mathscr{Y}[\mathscr{Y}]\mathscr{Z}[\mathscr{Z}]}$$

$$\boldsymbol{\mathscr{a}\mathscr{b}\mathscr{c}\mathscr{d}\mathscr{e}\mathscr{f}\mathscr{g}\mathscr{h}\mathscr{i}\mathscr{j}\mathscr{k}\mathscr{l}\mathscr{m}\mathscr{n}\mathscr{o}\mathscr{p}\mathscr{q}\mathscr{r}[\mathscr{r}]\mathscr{s}\mathscr{t}\mathscr{u}\mathscr{v}\mathscr{w}\mathscr{x}\mathscr{y}\mathscr{z}[\mathscr{z}]}$$

To use the math script font, put

```
\loadmscript
```

in your file, to load the fonts and define a control sequence `\mscript` so that `$\mscript a, \ldots, \mscript Z$` gives $\mathscr{a},\ldots,\mathscr{Z}$. In addition, you can

type	*to get*
`\mscript\altC`	$\mathscr{C}$
`\mscript\altG`	$\mathscr{G}$
`\mscript\altL`	$\mathscr{L}$
`\mscript\altQ`	$\mathscr{Q}$
`\mscript\altS`	$\mathscr{S}$
`\mscript\altY`	$\mathscr{Y}$
`\mscript\altZ`	$\mathscr{Z}$
`\mscript\imath`	$\imath$
`\mscript\jmath`	$\jmath$
`\mscript\altr`	r
`\mscript\altz`	z

Similarly, the command `\loadbmscript` loads the bold script fonts and creates the command `\bmscript`; alternate versions of letters are obtained exactly as with the script fonts.

MathTimeProfessional also provides a (non-slanted) “curly” font

ABCDEFG[G]HIJKLM[M]N[N]OPQ[Q]RSTUVWXY[Y]Z
abcdefghijklmnopqrstuvwxyz

which is similar to a traditional font often used for script. Certain of the symbols, like A, P, and R, may be so familiar in certain contexts that you might prefer them to the other script letters.

The command `\loadmcurly` loads the fonts, and produces a control sequence `\mcurly`. Alternate symbols, including $\imath$ and $\jmath$, are obtained just as with the script font.

XIII. Fraktur

To use the math fraktur font, put

```
\loadmfrak
```

in your file. This will load the Math Fraktur font, and create a control sequence `\mfrak`.

𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜[𝔜]ℨ
𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵[𝔵]𝔶[𝔶]𝔷

Similarly, `\loadbmfrak` loads the Bold Math Fraktur font.

𝕬𝕭𝕮𝕯𝕰𝕱𝕲𝕳𝕴𝕵𝕶𝕷𝕸𝕹𝕺𝕻𝕼𝕽𝕾𝕿𝖀𝖁𝖂𝖃𝖄[𝖄]𝖅
𝖆𝖇𝖈𝖉𝖊𝖋𝖌𝖍𝖎𝖏𝖐𝖑𝖒𝖓𝖔𝖕𝖖𝖗𝖘𝖙𝖚𝖛𝖜𝖝[𝖝]𝖞[𝖞]𝖟

NOTE: The symbols ℜ (`\Re`) and ℑ (`\Im`) from the basic *MathTimeProfessional* fonts are not exactly the same as the ℜ and ℑ from the math fraktur font. If you have loaded the math fraktur fonts, and would prefer to have these fraktur letters for `\Re` and `\Im`, you can just redefine `\Re` to mean `\mfrak R` and `\Im` to mean `\mfrak I`.

XIV. Other Weird Fonts

Two additional fonts have been constructed, though with misgivings, at the request of some users. Since these fonts are not "standard", be cautious about using them: you might end up creating notation that nobody else will be able to use!

Putting

```
\loadmbbi
```

in your file will load a math "blackboard bold italic" font, which has blackboard bold versions of slanted letters and numerals, as well as blackboard bold versions of all the Greek letters, both slanted and upright. The command `\mbbi` is used to obtain them:

```
$$\mbbi A,\ldots,\mbbi Z, \mbbi a,\ldots,\mbbi z
  \mbbi\imath,\mbbi\jmath, \mbbi0,\ldots,\mbbi9$$
```

gives

$$\mathbb{A},\ldots,\mathbb{Z},\mathbb{a},\ldots,\mathbb{z},\mathbb{\imath},\mathbb{\jmath},\mathbb{0},\ldots,\mathbb{9}$$

and

```
$$\mbbi\varGamma,\ldots,\mbbi\varOmega,
  \mbbi\alpha,\ldots,\mbbi\varkappa
  \mbbi\Gamma,\ldots,\mbbi\Omega
  \mbbi\upalpha,\ldots,\mbbi\upvarkappa $$
```

gives

$$\varGamma, \ldots, \varOmega, \alpha, \ldots, \varkappa, \Gamma, \ldots, \Omega, \upalpha, \ldots, \upvarkappa$$

The command `\mbbi` should not be used before other symbols.

There is also a math “holey bold italic” version. Putting

```
\loadmhbi
```

in your file will load a math “holey bold italic” font, which has holey versions of bold italic letters and slanted bold numerals, as well as holey versions of all the bold Greek letters, both slanted and upright. The command `\mhbi` is used to obtain them:

```
$$\mhbi A, \ldots, \mhbi Z, \mhbi a,\ldots, \mhbi z
  \mhbi\imath,\mhbi\jmath, \mhbi0,\ldots,\mhbi9$$
```

gives

$$A \ldots, Z, a, \ldots, z, \imath, \jmath, 0, \ldots, 9$$

and

```
$$\mhbi\varGamma,\ldots,\mhbi\varOmega,
  \mhbi\alpha,\ldots,\mhbi\varkappa
  \mhbi\Gamma,\ldots,\mhbi\Omega,
  \mhbi\upalpha,\ldots,\mhbi\upvarkappa$$
```

gives

$$\varGamma, \ldots, \varOmega, \alpha, \ldots, \varkappa, \Gamma, \ldots, \Omega, \upalpha, \ldots, \upvarkappa$$

Summary of all `\load...` Commands

In addition to

command	*loading*	*samples*
`\loadbm`	**b**old **m**ath	$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\ \boldsymbol{+}\ \boldsymbol{=}$
`\loadhm`	**h**eavy **m**ath	$\boldsymbol{+}\ \boldsymbol{x}\ \boldsymbol{=}$
`\loadsya`	**sy**mbolset **a**	$\leqq\ \Lleftarrow\ \curvearrowright$
`\loadbsya`	**b**old **sy**mbolset **a**	$\boldsymbol{\leqq}\ \boldsymbol{\Lleftarrow}\ \boldsymbol{\curvearrowright}$
`\loadhsya`	**h**eavy **sy**mbolset **a**	$\boldsymbol{\leqq}\ \boldsymbol{\Lleftarrow}\ \boldsymbol{\curvearrowright}$

there are

command	*loading*	*samples*	*using*
`\loadbb`	**b**lackboard **b**old	$\mathbb{A}, \mathbb{B}, \mathbb{C}$	`\bb`
`\loadbbd`	**b**lackboard **b**old **d**ark	$\mathbb{A}, \mathbb{B}, \mathbb{C}$	`\bbd`
`\loadhrb`	**h**oley **r**oman **b**old	$\mathbb{A}, \mathbb{B}, \mathbb{C}$	`\hrb`
`\loadhrbd`	**h**oley **r**oman **b**old **d**ark	$\mathbb{A}, \mathbb{B}, \mathbb{C}$	`\hrbd`
`\loadmscript`	**m**ath **script**	$\mathscr{A}, \mathscr{B}, \mathscr{C}$	`\mscript`
`\loadbmscript`	**b**old **m**ath **script**	$\boldsymbol{\mathscr{A}}, \boldsymbol{\mathscr{B}}, \boldsymbol{\mathscr{C}}$	`\bmscript`
`\loadmcurly`	**m**ath **curly**	$\mathcal{A}, \mathcal{B}, \mathcal{C}$	`\mcurly`
`\loadmfrak`	**m**ath **frak**tur	$\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$	`\mfrak`
`\loadbmfrak`	**b**old **m**ath **frak**tur	$\boldsymbol{\mathfrak{A}}, \boldsymbol{\mathfrak{B}}, \boldsymbol{\mathfrak{C}}$	`\bmfrak`
`\loadmbbi`	**m**ath **b**lackboard **b**old **i**talic	$\mathbb{A}, \mathbb{a}, \mathbb{\Gamma}, \mathbb{\gamma}$	`\mbbi`
`\loadmhbi`	**m**ath **h**oley **b**old **i**talic	$\mathbb{A}, \mathbb{a}, \mathbb{\Gamma}, \mathbb{\gamma}$	`\mhbi`

XV. Solving Family Problems

As mentioned before, a ‘`load`’ command like `\loadsya` loads three different fonts—the `mt2syat` font used for text size, the `mt2syas` font used for superscripts, and the `mt2syaf` font used for second-order superscripts—and these three fonts are made into a new TEX font ‘`family`’. (The `\loadbm` and `\loadhm` commands are different, because they involve more than one family.)

TEX allows 16 such families, the first 4 of which are always:

`\fam0`, the roman letters used for operator names like $\sin$ and $\cos$;

`\fam1`, the italic letters like x, y, and z, the Greek letters, and a few other symbols;

`\fam2`, most of the other standard symbols, like $\pm$ and $\oplus$.

`\fam3`, “large operators”, like $\sum$, various size delimiters, like $\big($, $\Bigg($, $\Bigg]$, $\big]$, etc.

The *MathTimeProfessional* fonts fill up a family with the special “math bold” fonts, and the ordinary text bold letters may take up yet another family. In addition, `plain` TeX normally uses up a family for text italic letters, and one for `typewriter` fonts, although people often delete these before making their format files.

There is consequently a real danger of running out of families if you not only want to`\loadbm` and `\loadhm`, and well as `\loadsya`, `\loadbsya`, and `\loadhsya`, but also want holey roman bold and black board bold letters, together with their “dark” versions, script and fraktur, together with their bold versions, and the curly font, not to mention the weird alphabets, and other symbol collections or alphabets that might be created in the future.*

Fortunately, even if you do happen to need symbols from all these different fonts in a single paper, it’s extremely unlikely that you will need them all within a single section, or in a single formula!

So there is a special device that allows you to get around this problem, should your mania for fonts lead you into trouble.

In this document, several alphabets were exhibited simply by declaring a font for the text size font, rather than using a `\load...` command. Nevertheless, all available families had been used up after loading the script and bold script fonts, so before introducing the curly font, the input file contains

```
        \substitute\loadmcurly\for\loadmscript
```

As a result, all the fonts that were previously gathered into the family used for the `\mscript` command were then replaced by the fonts for the `\mcurly` command, and it was now possible to use `\mcurly` to get the math curly fonts (although the math script fonts could no longer be obtained with `\mscript`).

*For those designing their own styles (section **VIII**), it should be pointed out that a second use of any particular `\load...` command *doesn’t* create a new family, it merely puts different size fonts into it the one that has already been created.

Before section **XIII**, we used

```
        \substitute\loadmfrak\for\loadbmscript
```

in order to use `\mfrak`, and

```
        \substitute\loadbmfrak\for\loadmcurly
```

to use `\bmfrak`.

The

```
        \substitue...\for...
```

constuction isn't allowed with `\loadbm` or `\loadhm`, but any other pair can be used, and you can even "substitute back". For example, in this document

```
        \substitute\loadmscript\for\loadmfrak
```

was just invoked, so that we could type a few math script characters like $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, . . . , $\mathscr{Z}$ (at the expense of no longer being able to get fraktur letters).

Of course, after you do something like

```
        \substitute\loadbb\for\loadsya
```

you should be careful not to try to use symbols like $\boxdot$ (`\boxdot`) and $\boxplus$ (`\boxplus`) that are meant to be used after `\loadsya`. (But you could still use `\bma\boxdot` if you had a `\loadbsya` command before, and haven't substituted for it.)

XVI. Changes From Previous Versions

1. The command `\vcorrection` (section **VI. 6 (e)**) replaces the now obsolete commands `\extcorrect` and `\vertcorrect`.

2. The syntax for the commands `\xl`, `\XL`, etc. (section **VII**) has now been completely changed, so that a combination like '`\xl\sum`' works exactly the same as if it were the name of a large operator: you can add an optional

`\limits` or `\nolimits` afterwards, and then specify the subscript and/or superscript, in either order, with neither being required. The old `\xlnl`, etc., have disappeared.

3. Section **VIII** (Designing Your Own Formats) has been shortened drastically, because the new macros perform almost all the necessary work for you, and the entire file `mtp2.tex` can simply be included in your style file (or `\input` as part of the style file). It is no longer necessary to declare any fonts aside from those used for text and operator names, or know the names and `\skewchar`'s for the fonts. The single command `\MTP` described in section **VIII**, together with any necesssary `\load...` commands, takes care of all details. The commands `\usingMTPsizes` and `\usingMTPextensions` are now obsolete.

4. The command for loading the “blackboard bold” fonts (section **XI**) has changed from `\loadbbb` to `\loadbb`, and `\bbb` has been changed to `\bb`. Similarly, the command `\loadbbbd` is now `\loadbbd`, and `\bbbd` is now `\bbd`.

5. The script and bold script alphabets (section **XII**) have the additional alternate symbol $\mathcal{S}$ (which I admit I added solely in order to use the script font for the $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX logo).