

The MathInst Package: New Math Fonts for T_EX

Alan Hoenig

June 3, 1997

If you've ever wanted to typeset proper mathematics using Caslon or Garamond or Baskerville or whatever types, *MathInst* may be of use to you. It allows you to create proper math fonts for use by T_EX and L^AT_EX. You need to tell it the family of Roman types you want to use and the raw math fonts—MathTime, Euler, or Lucida New Math—you can supply, and this script does the rest. All of *MathInst*'s fonts are properly scaled for compatibility with the Roman fonts.

If you supply the name of a typewriter and sans serif font as well, *MathInst* will not only create fonts that respond to the NFSS commands `\mathsf` and `\mathtt`, it will also scale these fonts properly to match the Roman math fonts.

If you supply the name of a calligraphic, fraktur, blackboard bold, and Greek bold font, *MathInst* will create matching fonts and commands for their use. In particular, if a Greek bold is present, *MathInst* will create bold math fonts.

MathInst also creates two style files—one for use by NFSS (L^AT_EX) and one for plain T_EX. There are also two matching test files, so you can not only test the fonts, but see how to access these fonts in a T_EX document.

At the end, *MathInst* places all your new fonts in their proper places, whether you use a traditional or a TDS T_EX system.

For many additional details, please read the remainder of this document. All the *MathInst* material can be found in the `fonts/utilities/mathinst` area of CTAN. Please forward comments and bug reports to me (Alan Hoenig) at `ajhjj@cunyvm.cuny.edu`.

1 Background; purpose; description

Although there are hundreds of outline fonts that T_EX can use to typeset non-technical material, it remains difficult to typeset technical documents using typefaces other than Computer Modern. Difficulties remain even when special math fonts are made available (such as Hermann Zapf's Euler fonts or the MathTime fonts), because specialized knowledge is needed to integrate the math fonts with the text fonts. It is the purpose of the *MathInst* software to perform this integration for an author or typesetter. Providing the author is willing to make some reasonable compromises, hundreds of new math families are now available. The *MathInst* software itself knows about three families of raw math fonts—MathTime, Lucida New Math, and Euler.

Here's what will be the result of a proper *MathInst* font generation:

1. math fonts properly integrating raw math fonts with a given text font family;

2. a macro file for users of plain \TeX which properly integrates the new fonts;
3. a style file for \LaTeX users which integrates the new fonts into the NFSS;
4. test files for both plain \TeX and \LaTeX users which demonstrate the use of the new fonts;
5. several log files which record aspects of the installation; and
6. a script for placing all fonts and other files in their proper locations.

In addition, the software checks to see if all target directories have been created. If not, a script has been prepared which will create the missing directories for you. This might be useful, for example, in preparing the many directories required by the TDS (\TeX Directory Structure) standard.

The *MathInst* software will work for “traditional” \TeX systems (where all tfm’s go in one place, all vf’s in another, and so on) or for the newer TDS systems.

Standard math font families consist of text fonts plus special math fonts. If you supply certain additional fonts, *MathInst* will integrate these other fonts in the math superfamilies. These optional additions include:

1. a typewriter font family;
2. a sans serif font family;
3. a fraktur font;
4. a Blackboard Bold font;
5. a calligraphic font; and
6. a Greek bold font (for proper bold math fonts)

Commands to typeset with these fonts are part of the styles that *MathInst* creates, the test contain examples illustrating their use. *MathInst* takes great care to scale all fonts so the final fonts have an x-height identical to the original text fonts. This way, we obtain the highest degree of visual compatibility between otherwise disparate fonts.

1.0.1 What you will need

Here’s what you’ll need to run the software.

- The modules themselves have been written in Perl (Perl5, actually), so you’ll need a Perl executable placed somewhere on your computer’s search path. Perl is freely available for virtually every computer platform; you’ll need to check out software archives to get hold of this language.
- You’ll also need a current version of Alan Jeffrey’s *fontinst* software, available from CTAN in the area `fonts/utilities/fontinst/inputs`. (Retrieve all these files and place them in a `texinputs` directory.)

- You'll need a copy of Damian Cugley's `pdcfset` font selection scheme for plain \TeX , which you can get in the `macros/plain/contrib/pdcmac` section of CTAN. (Retrieve the two files `pdccode.tex` and `pdcfset.dtx` from that directory and place them in a work directory. Run the `pdcfset.dtx` through plain \TeX ; this generates the documentation file `pdcfset.dvi` for `pdcfset` and the package itself `pdcfset.tex` which should be placed in a directory \TeX knows how to read from.)
- You'll need to install your text fonts **strictly** according to the `fontname2.1` standards. Information in the `info/fontname` section of CTAN explains this standard in detail. Two freely available tools assist in this kind of installation:
 1. `psnfss` by Sebastian Rahtz, and
 2. `VFinst` by me; available in the `fonts/utilities/vfinst` area of CTAN.

MathInst itself is found on CTAN in the

`fonts/utilities/mathinst`
area.

1.1 Platforms

The version supplied here runs under Unix. An untested set of scripts for DOS (or Windows) appears in the DOS subdirectory.

1.2 Legal jargon and caveats

This software is distributed according to the terms of the GNU 'copyleft' agreement. In addition, please note that if you make any alterations to any of the files in this package, you must change the names of the altered files. You are allowed to freely use this material and to pass it along (in the unaltered state!) provided you will not be using it to make money; otherwise, you must get in touch with me to secure permission.

While the author hopes that it will prove useful, it is distributed with no guarantee or warranty whatsoever. Users should take GREAT CARE to carefully back up their data before using.

2 Installing MathInst

Basic installation of this material involves placing files in proper directories. Thereafter, a few parameters need to be adjusted for your system.

Create a separate directory with a name like `mathinst` (say) somewhere on your system and place all the *MathInst* software there.

Create a working directory, with a name like `work` directly below this directory. We'll call this directory the *MathInst* working directory.

<code>\$psfonts_</code>	Where the outline font files are.
<code>\$afm_</code>	Where the outline font metric files are.
<code>\$vf_</code>	Where the T _E X virtual font <code>.vf</code> files are.
<code>\$tfm_</code>	Where the T _E X font metric <code>.tfm</code> files are.
<code>\$map_</code>	Where the dvips map file <code>psfonts.map</code> sits.
<code>\$inputs_</code>	Where is a T _E X input directory?
<code>\$texmf_</code>	What is the root TDS directory?
<code>\$installer_</code>	What installed the text fonts (vfirst or psnfss)?

Table 1: Some variables and their meanings.

Customizing the installation

Once the *MathInst* modules are in place, you'll need to make adjustments to some or all of the `par` parameter files that pertain to an installation. *MathInst will not run properly—and may not run at all—if these parameters are not set properly.* The parameter files are `mt.par`, `lu.par`, `eu.par`, or `ma.par` depending on which math fonts—MathTime, Lucida, Euler, or Mathematica—you'll be using. Once these variables have been set, *MathInst* has been completely installed.

These `par` files all follow the same format. Several variables at the beginning record the location of important files, and all these statements follow Perl syntax—variable names that begin with `$`, double quotes around strings, and statements ending with semicolons. Use the existing statements in the distribution files as models. For the author's system, these variables have the following values.

```

$psfonts_      = "/psfonts";
$afm_          = "$psfonts_/afm";
$vf_           = "/usr/lib/tex/fonts/vf";
$tfm_          = "/usr/lib/tex/fonts/tfm";
$map_          = "/usr/lib/tex/ps";
$inputs_       = "/usr/lib/tex/inputs";
$texmf_        = "/usr/local";
$installer_     = "vfirst";

```

Table 1 lists these variables and their meanings. The variable `$texmf_` must **only** be defined if your T_EX system conforms to the TDS standard (in which case, *MathInst* ignores `$afm_`, `$vf_`, and `$tfm_` and `$inputs_` holds the path where the `fd` files are). In a TDS system, *MathInst* also needs a value for `$installer_`.

Following these variables are a list of variables recording the names of the math fonts as they are on your system. Adjust them as necessary.

For the moment, leave the next few variables:

- `$ssize_` and `$sssize_`;
- `$encoding_`;
- a collection of variables with names like `$tt_`, `$sansserif_`, and so on; and

- a set of variables, all of which end in `...SF_`

commented out. In a Perl script, the comment character `#` works in the same way as the percent character `%` works in a `TeX` file. The default value for the `$encoding_` variable is `OT1`; you may change it now to `ot1`

```
$encoding_ = "ot1";
```

if you know for sure you are using all lowercase file names.

2.1 Installing raw math fonts

The *raw* math fonts consist of a series of outline font files plus the associated `afm` files. It's easy to install these fonts. Here are the steps appropriate for traditional systems.

1. Place the math font files with the other scalable fonts.
2. Place the `afm` files with your other `afm` files.
3. Make sure a proper entry exists for each math font in the `map` file for your `dvi` postprocessor.

Only the last point requires additional comments. For example, for the `dvips psfonts.map` file, we need entries like

```
% MathTime fonts...
mtsy      MTSY                      </psfonts/mtsy.pfa
mtex      MTEX                      </psfonts/mtex.pfa
rmtmi     RMTMI                     </psfonts/rmtmi.pfa
% Lucida New Math fonts...
lbma      LucidaNewMath-Arrows      </psfonts/lbma.pfb
lbme      LucidaNewMath-Extension   </psfonts/lbme.pfb
lbms      LucidaNewMath-Symbol      </psfonts/lbms.pfb
lbmi      LucidaNewMath-Italic      </psfonts/lbmi.pfb
lbmo      LucidaNewMath-AltItalic   </psfonts/lbmo.pfb
% Mathematica Math fonts...
Math1     Math1                     </psfonts/Math1.pfb
Math2     Math2                     </psfonts/Math2.pfb
Math3     Math3                     </psfonts/Math3.pfb
Math4     Math4                     </psfonts/Math4.pfb
Math5     Math5                     </psfonts/Math5.pfb
```

Note that these fonts are proprietary; please respect the licenses under which these fonts are sold or distributed.

Alone of the special math fonts, the Euler fonts are in the public domain. These fonts were implemented by METAFONT, and proper installation consists in placing the `tfm` and `pk` with their mates on your system. However, we will often be scaling these slightly to match various text fonts, and rather than regenerate many new bitmap fonts, it may be easier to use the scalable versions of these fonts, also available for free

(courtesy of Basil Malyshev; they may be found in the fonts/postscript/bakoma section of CTAN). In this case, these fonts will also need entries in your map file. These entries should look something like

```
%% Euler fonts...
euex10 euex10 </psfonts/euex10.pfb
eufb10 eufb10 </psfonts/eufb10.pfb
eufb5 eufb5 </psfonts/eufb5.pfb
eufb7 eufb7 </psfonts/eufb7.pfb
eufm10 eufm10 </psfonts/eufm10.pfb
eufm5 eufm5 </psfonts/eufm5.pfb
eufm7 eufm7 </psfonts/eufm7.pfb
eurb10 eurb10 </psfonts/eurb10.pfb
eurb5 eurb5 </psfonts/eurb5.pfb
eurb7 eurb7 </psfonts/eurb7.pfb
eurm10 eurm10 </psfonts/eurm10.pfb
eurm5 eurm5 </psfonts/eurm5.pfb
eurm7 eurm7 </psfonts/eurm7.pfb
eusb10 eusb10 </psfonts/eusb10.pfb
eusb5 eusb5 </psfonts/eusb5.pfb
eusb7 eusb7 </psfonts/eusb7.pfb
eusm10 eusm10 </psfonts/eusm10.pfb
eusm5 eusm5 </psfonts/eusm5.pfb
eusm7 eusm7 </psfonts/eusm7.pfb
```

Note that the Euler fonts come in a variety of weights and sizes; m and b represent medium and bold weights, and f, r, and s the fraktur, roman, and symbol fonts.

The installation for TDS systems is similar, except that the directories containing the font files have more specialized names. The outline files go in directories like

$$\langle texmf \rangle / \text{fonts/type1} / \langle supplier \rangle / \langle typeface \rangle$$

and the .afm files go in

$$\langle texmf \rangle / \text{fonts/afm} / \langle supplier \rangle / \langle typeface \rangle$$

(Don't forget—the entries in psfonts.map have to reflect these more specialized names.)

Here, $\langle supplier \rangle$ refers to the fontname abbreviation for the origin of the fonts, while $\langle typeface \rangle$ is a fontname name for the font family. Also, $\langle texmf \rangle$ is the root texmf directory. For example, the Euler fonts are provided courtesy of the American Mathematical Society (ams), so scalable versions of these fonts belong in places named

$$\begin{aligned} &\langle texmf \rangle / \text{fonts/type1/ams/euler} \\ &\langle texmf \rangle / \text{fonts/afm/ams/euler} \end{aligned}$$

MathTime: Additional Comments

If the digits for a roman family are too idiosyncratic for use with mathematics, you may wish instead to use Times Roman digits in math modes. If this is the case, there

is an additional parameter you can set in file `mt.par`. Set the Perl variable `$times_` to point to a raw Times Roman font.

MathTime fonts are available from Y&Y, Inc.

Euler fonts: Additional Comments

The Euler fonts need to be fleshed out using some characters from Computer Modern fonts. It may be convenient to use scalable versions of these fonts, and these are now freely available from two CTAN sources—the BaKoMa fonts discussed above, and the Blue Sky fonts, recently placed in the CTAN archive. However, scalable Euler fonts are only found in the BaKoMa suite.

One of the most pleasant surprises in this font work was how handsome the Euler fonts work in combination with text fonts. The Euler fonts are the only math fonts (other than Computer Modern) which are available for free.

Lucida New Math: Additional Comments

There are no bold versions of these fonts, so it is never appropriate to include a Greek bold font.

Lucida New Math fonts are available from Y&Y, Inc.

2.2 Installing the text fonts

The *MathInst* utilities expect the text font family (and the sans serif fonts too) to be installed using the Berry font naming scheme. At the moment, *MathInst* expects this to be done via the *VFinst* utility (by the current author) or by *psnfss* (by Sebastian Rahtz). The text fonts *must* have been installed using the original \TeX encoding (that is, OT1 or ot1).

2.3 Running MathInst

1. Once the *MathInst* software has been properly installed, you execute a module by switching to the working directory issuing a command like

```
../mathinst mt mbv
```

(it would be `..\mathinst mt mbv` in DOS). The double dots ‘`..`’ tell your computer to look for the *mathinst* script in the parent of the current working directory and to use the *mt* (MathTime) math fonts and the text font family *mbv* as arguments. At the moment, there are three allowable math abbreviations:

- *mt* (MathTime);
- *eu* (Euler); and
- *lu* (Lucida).

The three-character font family must be the family name under which the text fonts have been installed. *MathInst* produces the remainder of the scripts that you'll need to execute.

2. Execute `mkdirs.bat` to create any missing directories.
3. Execute `makepl.bat` to create certain property list files that fontinst will need.
4. Run the file `makemath.tex` through \TeX . This is a plain \TeX file.
5. Execute `makevf.bat` to create the binary font files that \TeX needs.
6. Test the fonts, and make any changes (as described below). After making any adjustments, return to the ' \TeX ' step (step 4).
7. When you are satisfied with the fonts, execute `putfonts.bat` to place all fonts, style files, and font descriptor files where they belong. Any files that remain in the working directory after this step can be safely deleted.

All these steps take little time to execute.

2.4 Log files

MathInst creates three log files with the extension `.mlg`. The files `makepl.mlg` and `makevf.mlg` contain information about the process used to create property files and font files. The name of the third log file begins with a 'z', followed by the 2-character math abbreviation, followed by the 3-character font family name. Thus, if I create math fonts from Caslon (`pac`) combined with Euler (`eu`), the third log file is called `zeupac.mlg`. This file records many of the decisions made by *MathInst* as well as many of its findings as it prepares scripts for your fonts, and this information may sometimes be of interest.

2.5 Making adjustments

The fonts that the above process produces are at best good approximations to high quality fonts. Fine details pertaining to character spacing have to be made by hand. *MathInst* tries to make this as easy as possible. For each run, it creates three "fudge" files:

`\family\rhax.mtx`
`\family\ihax.mtx`
`\family\iskew.mtx`

So, if you run *MathInst* on Monotype Baskerville fonts (family name `mbv`), you get three files `mbvrhax.mtx`, `mbvihax.mtx`, and `mbviskew.mtx`. Once these files are present in the working directory or its parent *MathInst* directory, *MathInst* never overwrites them, since you may well have recorded changes in these files.

The two `...hax` files allow you to make changes to the math glyphs themselves in the math roman and math italic fonts. This is by means of a command `\shiftglyph`, whose usage is described in these files. This command allows you to adjust the amount

of space on either side of a character, and the related amount by which subscripts nestle against the character. The `...skew` file makes it possible to adjust the positioning of accents on a math italic character, by means of a command called `\skewkern`.

In all cases, determining the numerical amounts to adjust various characters is a matter of trial and error.

3 Your new math virtual fonts

MathInst uses the two-character designations

```
mt MathTime
lu Lucida New Math
eu Euler
```

to refer to the various math fonts. The new math font families use the *z naming convention*, whereby the font family name for the new math fonts uses the two character math designations together with the text font family designation. Suppose we combine MathTime (mt) with Baskerville, whose family designation is mbv in the Berry scheme. The new fonts will be described in macro files `zmtmbv.tex` and `zmtmbv.sty` (for users of plain $\text{T}_\text{E}\text{X}$ or of $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$). Baskerville plus Euler or Baskerville plus Lucida would form the *z*-names `zeumbv` and `zlumbv`.

The fonts themselves follow the Berry scheme, but you don't need to keep track of this, since the *MathInst* style files load the fonts for you automatically and establish their correspondence either with familiar nicknames (`\it`, `\bf`) or with the NFSS. But if you find yourself poking about your font directories, here's a quick key to the many new fonts you'll see.

Additional font variants `m`, `e`, or `l` indicate a math font using MathTime, Euler, or Lucida New Math fonts, while variants `m`, `y`, or `v` denote the math italic, symbol, or extension fonts. For Baskerville, the four math fonts using MathTime fonts at text size have names

- `mbvrm7t`, the math Roman font;
- `mbvrm7m`, the math italic font;
- `mbvrm7y`, the symbol font; and
- `mbvrm7v`, the extension font.

A math Roman font connecting Baskerville and Euler or Baskerville and Lucida would be called `mbvre7t` or `mbvrl7t`.

MathInst does its best to 'fake out' new fonts at the script and scriptscript sizes. Computer Modern fonts provide the rôle model—those fonts are subtly redesigned at different sizes. Smaller fonts sizes also display greater letterspacing (additional space between adjacent characters) at smaller sizes. We can't redesign a scalable font, but we can create virtual fonts which differ only in the letterspacing for use in various contexts. *MathInst* creates installation files which take letterspacing into account. Therefore, you will see as many as three fonts for each of the above varieties listed above. In addition to

mbvrm7t, the math Roman font for text sizes, you'll see mbvrm7t7 and mbvrm7t5, the same fonts fine-tuned for use in script and scriptscript contexts. (The suffixes '7' and '5' recall the design sizes of seven and five points that Computer Modern typesetting uses for these sizes.) There is only a single extension font for any family.

3.1 Using the new fonts

Whether you use L^AT_EX or plain T_EX, you'll only need to remember the *z*-name for your new fonts. One purpose of the *MathInst* test files is to provide living examples showing just how the new fonts are invoked.

In a L^AT_EX document, all the work is done by the package file whose first name is precisely the *z*-name for the math fonts. As with all package files, its extension is *sty*. If you add a line like

```
\usepackage{zmtmbv}
```

following `\documentclass{...}` then all the usual L^AT_EX font switching mechanisms will now apply to the zmtmbv fonts rather than to the default Computer Modern fonts.

There is no font selection scheme for plain T_EX quite all-encompassing as is the L^AT_EX NFSS, but Damian Cugley's *pdcsel* comes gratifyingly close, and does one or two things better than NFSS. Like the L^AT_EX package, the plain macro file has the font *z*-name as its first name, but the extension is *tex*. To use these macros, simply place a statement like

```
\input zmtmbv
```

very near the beginning of your file. Thereafter, all the usual plain font commands like `\it`, `\rm`, `\tt`, `$`, `$$`, and so on refer to the new math fonts. The *pdcsel* package itself provides more flexibility than plain users are used to, and it would be well worth any reader's time to gain familiarity with this package.

For additional information regarding the installation and use of *MathInst* (including an extensive display of sample output), please see the book *T_EX Unbound: L^AT_EX and T_EX Strategies for Fonts, Graphics and More* by Alan Hoenig (1997, Oxford University Press).

4 Communicating with the author

Please feel free to get in touch with me (Alan Hoenig) if you have any comments or bug reports.

Alan Hoenig
ajhjj@cunyvm.cuny.edu
John Jay College
Department of Mathematics
445 West 59th Street
New York, NY 10019 USA

May 1997