The macro package lucimatx*

Walter Schmidt

2005/11/28

Abstract

This document describes the macro package lucimatx, which serves for using the Lucida® Bright and Lucida® New Math fonts[1] with LaTeX. It constitutes a successor to the packages lucidabr and lucbmath. The present document serves also as a font sample, since it was typeset using the Lucida fonts and shows at least one member of each single font family in the collection.

Contents

*This document refers to version 1.0.1 of the package.

[1] Lucida ® is a trademark of Bigelow & Holmes Inc. registered in the U. S. Patent & Trademark Office and other jurisdictions.

The fonts are non-free and must be acquired from a licensed distributor.

1 Introduction

Lucida is a 'super-family' of harmonized typefaces that includes a variety of styles: serifed and sans-serif, roman and italic, normal and bold weights, scripts, typewriter and blackletter. The related math fonts ('Lucida New Math') contain the character set needed for use with TEX, including the 'AMS symbols', fraktur and doublestroke alphabets and complete Greek alphabets in both upright and slanted form. Even though they were designed for use with the Lucida text fonts, they may blend harmoniously with other typefaces, too.

To make the use of the Lucida text and math fonts with LATEX most easy, the macro package lucimatx is provided. Loading the package

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

makes LATEX use the Lucida Bright and Lucida New Math fonts in place of the default Computer Modern text and math fonts. In detail:

- ▷ Computer Modern Roman is replaced with Lucida Bright,
- ▷ CM Sans is replaced with Lucida Sans,
- ▷ CM Typewriter is replaced with Lucida Sans Typewriter,
- ▷ the CM Math fonts are replaced with Lucida New Math.

The following sections describe the Lucida fonts, the particular features of the package, and the additional options that control its behavior.

2 Package option syntax

All package options are set using a ⟨*key*⟩ = ⟨*value*⟩ scheme; valid keys and values will be explained in the course of this document. So-called 'boolean' options accept `true` and `false` as values. Specifying a boolean key without a value is equivalent to setting it to `true`.

For a summary of all package options see section 8.

3 Font size and baseline distance

The Lucida text and math fonts are larger than many other typefaces (including CM) when used at the same nominal size. This may lead to problems with the default vertical spacing of existing document classes and with the blending with other fonts possibly used in the document. The package lucimatx provides two alternative means to control the effective font size, if you cannot (or do not want to) modify the document class acordingly:

First, there is an option key `scale`.[2] Its value indicates a factor by which the Lucida fonts will be scaled; for instance, `scale=0.9` causes the Lucida fonts to be used at 90% of their nominal size (which is a reasonable value under normal circumstances).

[2] Notice that the key is named `scale`, rather than `scaled` as in certain other packages.

Second, for the sake of compatibility with the package lucidabr, lucimatx provides also an alternative way to control the effective font size: When the package is loaded with the boolean option `lucidascale` or `lucidasmallscale` set to `true`, a sort of 'nonlinear' scaling is enabled, i.e., different sizes are scaled differently:

	< 5 pt	...	10 pt	...	≥ 22.5 pt
`lucidascale`:	1.04	...	0.95	...	0.89
`lucidasmallscale`:	0.98	...	0.90	...	0.85

Notice that linear and nonlinear scaling cannot be combined, i.e., you can select either the option `scale=...` or `lucidascale` or `lucidasmallscale`.

Because of the relatively large x-height of the Lucida fonts, the ratio of baseline distance to font size must be considerably larger than with Computer Modern. The standard document classes (and most others, too) are, however, optimized for the CM fonts, so the leading must be enlarged, even when the Lucida fonts are already scaled down. Since the optimum setting is a question of document design and line width, adjusting the baseline distance is left to the user. If necessary, all you have to do is to issue the command

`\linespread{`⟨*factor*⟩`}`

in the preamble; as a result, the baseline distance is enlarged by the given factor. Typical values for ⟨*factor*⟩ are 1.04 ... 1.05, provided that the fonts are simultaneously scaled down to 90%.

4 The Lucida text fonts

Table 1 lists all font families in the Lucida distribution[3] and the available font encodings, series and shapes. For more detailed information see the file `doc/fonts/lucida/lucida.txt` in your TEX file system hierarchy.

4.1 Choice of text fonts through package options

When the package lucimatx is used, the roman font family of the document (`\rmdefault`) can be chosen through the package option `romanfamily`. Possible values are:

`romanfamily=bright`: Lucida Bright (default setting)

`romanfamily=bright-osf`: Lucida Bright with oldstyle figures

`romanfamily=fax`: Lucida Fax

`romanfamily=casual`: Lucida Casual

Under normal circumstances, the commands `\itshape` and `\textit` select the italic shape that belongs to the current font family. The package can optionally replace this with the Lucida Calligraphic or Lucida Handwriting font. This facility is controlled using the package option `italics`. Possible values are:

`italics=handwriting`: use *Lucida Handwriting*

[3] as of October 2005

encodings	family	series	shape(s)
Lucida Bright			
T1, TS1, LY1	hlh	m, b	n, it, sl, sc
Lucida Bright, oldstyle figures			
T1, TS1	hlhj	m	n, it, sc
T1, TS1	hlhj	b	n, sc
Lucida Sans			
T1, TS1, LY1	hls	m, b, ub	n, it
Lucida Sans Typewriter			
T1, TS1, LY1	hlst	m, b	n, sl
Lucida Typewriter			
T1, TS1, LY1	hlct	m, b	n, sl
Lucida Fax			
T1, TS1, LY1	hlx	m, b	n, it
Lucida Calligraphy			
T1, TS1, LY1	hlce	m	it
Lucida Casual			
T1, TS1, LY1	hlcn	m	n, it
Lucida Handwriting			
T1, TS1, LY1	hlcw	m	it
Lucida Blackletter			
T1, TS1, LY1	hlcf	m	n

Table 1: NFSS classification of the Lucida text fonts

`italics=calligraphic`: use *Lucida Calligraphic*

`italics=default`: do not replace the italics (default setting)

Notice, however, that this affects the roman font family only. The italics of the sans-serif and typewriter families will remain unchanged.

By default, the package `lucimatx` selects `Lucida Sans Typewriter` as the typewriter font family (`\ttdefault`). To use the serifed variant `Lucida Typewriter` instead, specify the boolean package option `seriftt=true`.

4.2 Encoding

The Lucida text fonts are currently available with T1/TS1 (European) and LY1 (TeX'n ANSI) encoding, whereas they are no longer made available with the obsolete OT1 encoding (which is still the default setting in LaTeX).

LY1 encoding is supported primarily with respect to Y&Y-TeX systems[4] and legacy documents, and it implies a few restrictions: The font family `hlhj` is not

[4] Using the Lucida fonts with T1/TS1 encoding requires virtual font support, which is not available on Y&Y-TeX.

available with LY1 encoding, the command `\oldstylenums` does not work properly, and Euro symbols (`\texteuro`) are not available in all font families. These restrictions do not apply with T1/TS1 encoding, which is to be preferred, if possible.

When the lucimatx package is loaded with OT1 encoding still being in effect, it will change the font encoding to T1 and load the package textcomp additionally. To use LY1 instead, or to declare additional encodings, you must load the fontenc package *before* lucimatx; in this case textcomp is *not* loaded automatically.

4.3 Additional usage notes

The Lucida text fonts do not include all symbols of the TS1 ('text companion') font encoding. Only those symbols are available, that belong to the ISO-Adobe 1 character set, plus the Ω (`\textohm`) and the € `\texteuro`.

The font series `b` is actually assigned *demibold* fonts. To access Lucida Sans Bold, which is assigned to the series `ub` (ultra bold), use a declaration such as `\fontseries{ub}\sffamily`.

Oldstyle numerals are available in the font family `hlhj`, where they constitute the default numerals, and in the family `hlh`, where they can be accessed through `\oldstylenums`. The latter way requires the use of a recent version of the textcomp package, as shipped with LaTeX from the 2003/12/01 release on.

The blackletter font is primarily intended to be used in math. When used as a text font with T1 or LY1 encoding, no 'long s' is made available, while there are many accented or special characters that actually do not make much sense with this typeface. Consequently, no package option or 'user level' command is provided to select Lucida Blackletter for text, but you can access it through the declaration `\fontfamily{hlcf}\selectfont`.

5 Using Lucida New Math with 'foreign' text fonts

Instead of changing both text and math fonts of the document to Lucida, it is possible to make the package change only the math fonts; to do so, specify the boolean option `onlymath=true`. Thus, the Lucida New Math fonts can be used with 'foreign' text fonts. Notice that the default font families used for the text must be selected *before* loading lucimatx; for instance, to use Lucida New Math with Charter as the default roman typeface:

```
\usepackage{charter}
\usepackage[onlymath=true,scale=.9,...]{lucimatx}
```

The text font encoding, too, must be chosen *before* the loading of lucimatx, unless you are staying with OT1.

The options described in the above section 4.1 have no effect, when `onlymath` is selected.

6 Math typesetting

6.1 Choice of math italics

Lucida New Math includes two italic math alphabets with different shapes of the lower-case Latin letters:

Lucida New Math Italic: $a, b \ldots\ m, n, o, p, q, r \ldots\ z$
Lucida New Math Alternate Italic: $a, b \ldots\ m, n, o, p, q, r \ldots\ z$

Despite its name, the 'Alternate' variant is the one that is used by default for variables in formulas. To use the 'standard' one instead, load lucimatx with the package option `stdmathitalics=true`.

6.2 Upright vs. italic letters in math mode

With TEX or LATEX, uppercase Greek letters in math mode are usually typeset as upright, even though they are usually meant to designate variables. This violates the International Standards ISO31-0:1992 to ISO31-13:1992. Loading the package with the option `math-style=iso` will result in slanted uppercase Greek letters `\Gamma`, `\Delta` etc.

Upright lowercase and uppercase Greek letters are available with command names such as `\upalpha`, `\upbeta`, `\upGamma`, `\upDelta`, etc. They are always upright, regardless of the value of the `math-style` option.

The package supports also the French tradition of typesetting all Greek variables (both uppercase and lowercase) upright, and ditto the uppercase roman variables. This behavior is selected by specifying the option `math-style=french`.

The following table is to clarify the differences between TEX's default behavior and the options `math-style=iso` and `french`:

default: $M \in \Gamma \Leftrightarrow OM = x\rho$
ISO : $M \in \mathit{\Gamma} \Leftrightarrow OM = x\rho$
French : $\mathrm{M} \in \Gamma \Leftrightarrow \mathrm{OM} = x\rho$

6.3 Using bold math fonts

All Lucida math fonts, except for the extensible operator symbols, are available in a normal and a bold version. The appropriate way to use the bold math fonts depends on the particular purpose:

- The declaration `\boldmath` will embolden all formulas within its scope. Use it, for instance, to emphasize complete formulas or to make sure that mathematical expressions within bold section titles also appear in bold type. The declaration `\boldmath` can, however, *not* be issued when you are already in math mode. Thus it is not a suitable means to embolden single letters, e.g., if you want to designate vectors with bold type.

- The 'canonical' way to embolden single letters or symbols is to use the macro package bm. It defines the command `\bm`, which can embolden almost anything—provided that the required font exists. The package bm belongs to the tools collection, which is part of every LATEX system. Make sure to load the package only *after* lucimatx.

6.4 Math alphabets

- ▷ The standard math alphabets `\mathrm`, `\mathsf` and `\mathtt` are mapped to the roman, sans-serif and typewriter text fonts, as expected.
- ▷ Correspondingly, `\mathbf` uses the bold variant of the roman text font.
- ▷ `\mbf` is similar, but uses the upright LucidaNewMath-Demibold font. with the spacing and the letter shapes adapted to math typesetting. Thus `\mbf` is appropriate to typeset single variables, while `\mathbf` can be used, e.g., to emphasize an operator name.
- ▷ `\mathcal` uses the calligraphic alphabet of the Lucida New Math Symbol font; notice that only upper-case Latin letters are available: $\mathcal{A}, \mathcal{B}, \ldots \mathcal{Z}$
- ▷ `\mathbb` uses the doublestroke alphabet in the Lucida New Math fonts; again, only upper-case letters are available: $\mathbb{A}, \mathbb{B}, \ldots \mathbb{Z}$
- ▷ `\mathfrak` points to the Lucida Blackletter font: $\mathfrak{a}, \mathfrak{b}, \ldots \mathfrak{z}, \mathfrak{A}, \mathfrak{B}, \ldots \mathfrak{Z}$

The 'Blackboard bold' and fraktur alphabets are provided by the lucimatx package; do *not* load the packages amsfonts or amssymb for this purpose!

The symbols `\Re` and `\Im` are, by default, not exactly the same as the corresponding letters from these `\mathfrak` alphabets. If you would prefer to have `\Re` and `\Im` use the `\mathfrak` alphabet, just redefine these macros appropriately:

```
\renewcommand{\Re}{\mathfrak{R}}
\renewcommand{\Im}{\mathfrak{I}}
```

6.5 Digits in math mode

Digits in formulas are normally taken from the default roman text font family of the document, i.e., `\rmdefault`. This may not always meet your expectations, for instance, if this font family has oldstyle digits. The option `stdmathdigits=true` forces the macro package to use always the Lucida lining figures in math mode, regardless of the text fonts.

6.6 AMS symbols

The Lucida New Math fonts contain the so-called 'AMS symbols' described, for instance, in [1], section 8.9. With Computer Modern the symbols are made available by loading the package amssymb; with Lucida, in contrast, no additional package is needed beside lucimatx. Do *not* load amssymb or amsfonts; all features usually provided by these packages are already built into lucimatx.

Notice that the mathematical symbols `\digamma` and `\yen` are not available with the Lucida fonts.

6.7 Additional symbols not available with standard LaTeX

6.7.1 Integrals

A surface integral symbol $\iint$ is available under the name `\surfint`.

6.7.2 Negated relation symbols

Lucida New Math includes a number of ready-made negated relation symbols, which are normally built from pieces. For instance, with lucimatx you should write \notequiv instead of \not\equiv. For most of these symbols an alternative name is indicated, which follows the naming scheme of the amssymb package:

≢	\notequiv, \nequiv	≉	\notapprox, \napprox
⊄	\notsubset, \nsubset	⊅	\notsupset, \nsupset
⊈	\notsubseteq, \nsubseteq	⊉	\notsupseteq, \nsupseteq
≁	\notsim, \nsim	≄	\notsimeq, \nsimeq
⋤	\notsqsubseteq, \nsqsubseteq	⋥	\notsqsupseteq, \nsqsupseteq
≭	\notasymp, \nasymp	≇	\notcong, \ncong
∌	\notni		

6.7.3 Miscellaneous symbols

Further extra symbols, too, are not defined with standard LaTeX. These are relations, except for the lightning and the dashed arrows, which are of type 'ordinary', and the open face brackets, which are delimiters:

≝	\defineeq, \defineequal	≙	\hateq
≔	\coloneq	≕	\eqcolon
⊷	\image	⊶	\original
⊷	\longimage	⊶	\longoriginal
⇅	\updownarrows	⇵	\downuparrows
⇜	\leadsfrom	↯	\lightning
⇡	\dashuparrow	⇣	\dashdownarrow
⟦	\ldbrack	⟧	\rdbrack

6.8 Compatibility with amsmath

In contrast to its predecessors lucidabr and lucbmath, the package lucimatx should be fully compatible with amsmath, and it should not matter which package is loaded first.[5] Also, it should work together with the chemarr package, which constitutes an extension to amsmath.

The only exception affects the commands \leftroot and \uproot provided by amsmath. They have no effect, when the Lucida math fonts are used, because lucimatx has a method of its own to optimize the placement of root indices without interaction of the user.

7 Diagnostic output

lucimatx applies a large number of changes to the default font settings of the LaTeX program, which would normally result in lots of warnings on the terminal (console). To avoid this, the package makes any font warnings appear in the log file only. Optionally you can suppress font warnings in the log file, too, by

[5] This holds for amsmath version 2.13. Later versions may behave differently; in case you experience any problems, please let us know!

issuing the package option `nofontinfo=true`. Finer control over the diagnostic output can be achieved by using the package tracefnt, which is to be loaded after lucimatx.

8 Option summary

The following table lists all options (keys) of the lucimatx package. Values that correspond to the default behavior of the package are marked by an asterisk and need normally not to be specified.

key	values	see section…
`italics`	`default`*, `handwriting`, `calligraphic`	4.1
`lucidascale`	`true`, `false`*	3
`lucidasmallscale`	`true`, `false`*	3
`onlymath`	`true`, `false`*	5
`math-style`	`tex`*, `iso`, `french`	6.2
`nofontinfo`	`true`, `false`*	7
`romanfamily`	`bright`*, `bright-osf`, `fax`, `casual`	4.1
`scale`	⟨*factor*⟩	3
`seriftt`	`true`, `false`*	4.1
`stdmathdigits`	`true`, `false`*	6.5
`stdmathitalics`	`true`, `false`*	6.1

9 Migrating from lucidabr v4.2 to lucimatx

Almost all documented features of the packages lucidabr and lucbmath (version 4.2) are available with lucimatx, too. Exceptions affect mainly the facilities to choose between italic and upright letters in math mode, i.e., the former package options `mathitalics3`, `slantedgreek`, `uprightgreek` and `vargreek`: A completely different behavior has been implemented now, which is—to our opinion—much more useful and complies with de-facto standards implemented already in other packages; see section 6.2.

Most other options are still available through the key-value syntax; a few are no longer needed:

`expert`, `noexpert`: These options don't exist any more, since the Lucida fonts are always distributed as a whole now.

`nolucidascale`, `lucidascale`, `lucidasmallscale`: The latter two options exist still (see section 3), but the **default** behavior is now **not to apply any scaling.**

`mathitalic1`, `mathitalic2`: The former option corresponds to the default behavior, the latter corresponds to `stdmathitalics=true`.

`OT1`, `T1`, `LY1`: These options are no longer needed; lucimatx works with any of OT1, T1 or LY1 and will auto-detect the encoding that is in use when package is loaded.

`seriftt`: See section 4.1.

`fax`, `casual`: Use the corresponding values of the option `romanfamily` instead.

`handwriting`, `calligraphic`: Use the corresponding values of the option `italics` instead.

`noamssymbols`: The option is no longer provided, because contemporary TEX systems have sufficient memory.

`errorshow`, `warningshow`, `nofontinfo`: lucimatx provides only the latter option. Finer control over font-related infos and warnings can be achieved by loading the package tracefnt after lucimatx.

`altbullet`: The option is no longer provided. Instead of patching the font definitions, redefine `\labelitemi` appropriately, if you don't like the shape of the default item label `\textbullet`.

10 The implementation of the macro package lucimatx

The package code is, to some extent, based on lucidabr v4.2, written by Sebastian Rahtz and David Carlisle, distributed under the LATEX Project Public License (LPPL).

10.1 Options

We require the keyval package, so that options can be specified using a key-value syntax:

```
⟨*package⟩
\RequirePackage{keyval}
```

Additional tools to declare and evaluate options, partially 'stolen' from hyperref. (Thank you, Heiko!)

```
\def\Lbt@boolkey{\@dblarg\Lbt@@boolkey}
\def\Lbt@true{true}
\def\Lbt@false{false}
\def\Lbt@@boolkey[#1]#2#3{%
  \lowercase{\def\Lbt@tempa{#3}}%
  \ifx\Lbt@tempa\@empty
    \let\Lbt@tempa\Lbt@true
  \fi
  \ifx\Lbt@tempa\Lbt@true
  \else
    \ifx\Lbt@tempa\Lbt@false
    \else
      \let\Lbt@tempa\relax
    \fi
  \fi
  \ifx\Lbt@tempa\relax
    \Lbt@WarnOptionValue{#3}{#1}{'true' or 'false'}%
  \else
    \csname Lbt@#2\Lbt@tempa\endcsname
  \fi
}
\def\Lbt@WarnOptionValue#1#2#3{%
  \PackageWarningNoLine{lucimatx}{%
    Unexpected value '#1'\MessageBreak
    of option '#2' instead of\MessageBreak
```

```
    #3%
  }
}
```

Now lets's declare the keys and the variables to store the values. These are for `scale`, `lucidascale`, `lucidasmallscale`:

```
\let\Lbt@scale\relax
\define@key{Lbt}{scale}[1.0]{%
  \def\Lbt@scale{#1}}
\newif\ifLbt@lucidascale\Lbt@lucidascalefalse
\newif\ifLbt@lucidasmallscale\Lbt@lucidasmallscalefalse
\define@key{Lbt}{lucidascale}[true]{%
  \Lbt@boolkey{lucidascale}{#1}}
\define@key{Lbt}{lucidasmallscale}[true]{%
  \Lbt@boolkey{lucidasmallscale}{#1}}
```

`onlymath`:

```
\newif\ifLbt@onlymath\Lbt@onlymathfalse
\define@key{Lbt}{onlymath}[true]{%
  \Lbt@boolkey{onlymath}{#1}}
```

`seriftt`:

```
\newif\ifLbt@seriftt\Lbt@serifttfalse
\define@key{Lbt}{seriftt}[true]{%
  \Lbt@boolkey{seriftt}{#1}}
```

`stdmathitalics`:

```
\newif\ifLbt@stdmathitalics\Lbt@stdmathitalicsfalse
\define@key{Lbt}{stdmathitalics}[true]{%
  \Lbt@boolkey{stdmathitalics}{#1}}
```

`romanfamily`:

```
\def\Lbt@rmfamily{bright}
\define@key{Lbt}{romanfamily}[bright]{%
  \def\Lbt@rmfamily{#1}}
```

`italics`:

```
\def\Lbt@italics{default}
\define@key{Lbt}{italics}[default]{%
  \def\Lbt@italics{#1}}
```

`math-style`:

```
\def\Lbt@style{tex}
\define@key{Lbt}{math-style}[tex]{%
  \def\Lbt@style{#1}}
```

`stdmathdigits`:

```
\newif\ifLbt@stdmathdigits\Lbt@stdmathdigitsfalse
\define@key{Lbt}{stdmathdigits}[true]{%
  \Lbt@boolkey{stdmathdigits}{#1}}
```

`nofontinfo`:

```
\newif\ifLbt@nofontinfo\Lbt@nofontinfofalse
\define@key{Lbt}{nofontinfo}[true]{%
  \Lbt@boolkey{nofontinfo}{#1}}
```

Evaluating the options:

```
\def\ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\Lbt@tempa\@empty
  \@for\CurrentOption:=\@classoptionslist\do{%
    \@ifundefined{KV@#1@\CurrentOption}%
    {}%
    {%
      \edef\Lbt@tempa{\Lbt@tempa,\CurrentOption,}%
      \@expandtwoargs\@removeelement\CurrentOption
        \@unusedoptionlist\@unusedoptionlist
    }%
  }%
  \edef\Lbt@tempa{%
    \noexpand\setkeys{#1}{%
      \Lbt@tempa\@ptionlist{\@currname.\@currext}%
    }%
  }%
  \Lbt@tempa
  \let\CurrentOption\@empty
}
\ProcessOptionsWithKV{Lbt}
\AtEndOfPackage{%
  \let\@unprocessedoptions\relax
}
```

10.2 Scaling

`\DeclareLucidaFontShape` is defined according to the scaling required through the options:

```
\ifx\Lbt@scale\relax
  \def\DeclareLucidaFontShape#1#2#3#4#5#6{%
     \DeclareFontShape{#1}{#2}{#3}{#4}{<->#5}{#6}}
\else
  \edef\Lbt@@scale{s*[\csname Lbt@scale\endcsname]}%
  \def\DeclareLucidaFontShape#1#2#3#4#5#6{%
    \DeclareFontShape{#1}{#2}{#3}{#4}{<->\Lbt@@scale #5}{#6}}
  \PackageInfo{lucimatx}{Scaling Lucida fonts to \Lbt@scale}
\fi
```

The implementation of `lucidascale` and `lucidasmallscale` is adopted from the `lucidabr` package. The particular scaling values actually come from Frank Mittelbach:

```
\ifLbt@lucidascale
  \def\DeclareLucidaFontShape#1#2#3#4#5{%
    \DeclareFontShape{#1}{#2}{#3}{#4}{%
    <-5.5>s*[1.04]#5%
    <5.5-6.5>s*[1.02]#5%
    <6.5-7.5>s*[.99]#5%
    <7.5-8.5>s*[.97]#5%
    <8.5-9.5>s*[.96]#5%
    <9.5-10.5>s*[.95]#5%
    <10.5-11.5>s*[.94]#5%
```

```
    <11.5-13>s*[.93]#5%
    <13-15.5>s*[.92]#5%
    <15.5-18.5>s*[.91]#5%
    <18.5-22.5>s*[.9]#5%
    <22.5->s*[.89]#5}{}%
  }
  \PackageInfo{lucimatx}{Nonlinear scaling (‘lucidascale’)}
\fi
\ifLbt@lucidasmallscale
  \def\DeclareLucidaFontShape#1#2#3#4#5{%
    \DeclareFontShape{#1}{#2}{#3}{#4}{%
    <-5.5>s*[.98]#5%
    <5.5-6.5>s*[.96]#5%
    <6.5-7.5>s*[.94]#5%
    <7.5-8.5>s*[.92]#5%
    <8.5-9.5>s*[.91]#5%
    <9.5-10.5>s*[.9]#5%
    <10.5-11.5>s*[.89]#5%
    <11.5-13>s*[.88]#5%
    <13-15.5>s*[.87]#5%
    <15.5-18.5>s*[.86]#5%
    <18.5-22.5>s*[.85]#5%
    <22.5->s*[.84]#5}{}%
  }
  \PackageInfo{lucimatx}{Nonlinear scaling (‘lucidasmallscale’)}
\fi
```

10.3 Choice of text fonts

If the text fonts are not to be touched, we issue a message only:

```
\ifLbt@onlymath
  \PackageInfo{lucimatx}{Changing only the math fonts}
\else
```

Otherwise, set up the default font families, according to the options specified:

```
  \renewcommand{\sfdefault}{hls}
  \renewcommand{\ttdefault}{hlst}
  \def\@tempa{bright}
  \ifx\Lbt@rmfamily\@tempa
    \renewcommand{\rmdefault}{hlh}
  \else
  \def\@tempa{fax}
  \ifx\Lbt@rmfamily\@tempa
    \renewcommand{\rmdefault}{hlx}
  \else
  \def\@tempa{casual}
  \ifx\Lbt@rmfamily\@tempa
    \renewcommand{\rmdefault}{hlcn}
  \else
  \def\@tempa{bright-osf}
  \ifx\Lbt@rmfamily\@tempa
    \renewcommand{\rmdefault}{hlhj}
  \else
    \Lbt@WarnOptionValue{%
```

```
      \Lbt@rmfamily}{romanfamily}{‘bright’, ‘bright-osf’, ‘fax’ or ‘casual’}%
  \fi
  \fi
  \fi
  \fi
  \ifLbt@seriftt
    \renewcommand{\ttdefault}{hlct}
  \fi
```

If the encoding is still OT1, switch to T1 instead:

```
  \edef\operator@encoding{\encodingdefault}
  \def\@tempa{OT1}
  \ifx\operator@encoding\@tempa
```

Switch to T1 only after selecting a new `\rmdefault`, to avoid spurious loading of an EC font, which may not exist:

```
    \fontfamily{\rmdefault}\RequirePackage[T1]{fontenc}
    \RequirePackage{textcomp}
    \PackageInfo{lucimatx}{%
      Default font encoding changed to T1;\MessageBreak
      package ‘textcomp’ loaded additionally}
  \fi
```

Current textcomp blocks `\oldstylenums` in family `hlh`, because OsF are not available in all shapes. Fix this:

```
  \@ifpackageloaded{textcomp}{%
      \ifx\DeclareEncodingSubset\@undefined
        \PackageWarningNoLine{lucimatx}{%
          Obsolete version of package textcomp found;\MessageBreak
          \string\oldstylenums\space may not work as expected}
      \else
        \DeclareEncodingSubset{TS1}{hlh}{1}
        \fi}{}
```

(Demi-)bold Lucida fonts are assigned the series `b`:

```
  \renewcommand{\bfdefault}{b}
```

If required, patch the font definitions of the roman font family and replace the default italics with the Calligraphic or Handwriting font:

```
  \def\@tempa{default}
  \ifx\Lbt@italics\@tempa
  \else
  \def\@tempa{calligraphic}
  \ifx\Lbt@italics\@tempa
    \normalfont
    \DeclareFontShape\encodingdefault\rmdefault{m}{it}%
                                      {<->ssub*hlce/m/it}{}
    \PackageInfo{lucimatx}{Using Lucida Calligraphic for italics}
  \else
  \def\@tempa{handwriting}
  \ifx\Lbt@italics\@tempa
    \normalfont
    \DeclareFontShape\encodingdefault\rmdefault{m}{it}%
                                      {<->ssub*hlcw/m/it}{}%
    \PackageInfo{lucimatx}{Using Lucida Handwriting for italics}
  \else
```

```
    \Lbt@WarnOptionValue{%
      \Lbt@italics}{italics}{‘default’, ‘calligraphic’ or ‘handwriting’}%
  \fi
  \fi
  \fi
\fi
```

10.4 Choice of math fonts

If required, use the ‘standard’ math italics (which are assigned the shape `itx`). The shape is stored in the macro `\letters@shape`, which will be evaluated when the math symbol fonts are actually set up:

```
\ifLbt@stdmathitalics
  \def\letters@shape{itx}
\else
  \def\letters@shape{it}
\fi
```

Declare three switches to indicate whether uc Greek, lc Greek and uc Latin are to be typeset upright, rather than slanted:

```
\newif\ifLbt@uprightGreek
\newif\ifLbt@uprightgreek
\newif\ifLbt@uprightLatin
```

Now set the above switches, according to the package options. They will be evaluated when the math symbols are declared:

```
\def\@tempa{tex}
\ifx\Lbt@style\@tempa
  \Lbt@uprightGreektrue
  \Lbt@uprightgreekfalse
  \Lbt@uprightLatinfalse
  \PackageInfo{lucimatx}{TeX math style}
\else
\def\@tempa{iso}
\ifx\Lbt@style\@tempa
  \Lbt@uprightGreekfalse
  \Lbt@uprightgreekfalse
  \Lbt@uprightLatinfalse
  \PackageInfo{lucimatx}{ISO math style}
\else
\def\@tempa{french}
\ifx\Lbt@style\@tempa
  \Lbt@uprightGreektrue
  \Lbt@uprightgreektrue
  \Lbt@uprightLatintrue
  \PackageInfo{lucimatx}{‘French’ math style}
\else
  \Lbt@WarnOptionValue{%
    \Lbt@style}{math-style}{‘tex’, ‘iso’ or ‘french’}%
\fi
\fi
\fi
```

10.5 Font info

This package makes a lot of redefinitions. The warnings can be rather annoying, so we make LaTeX write both info and warning messages to the log file only. If the option `nofontinfo` was selected, infos and warnings are entirely suppressed.

```
\ifLbt@nofontinfo
  \let\@font@info\@gobble
  \let\@font@warning\@gobble
\else
 \def\@font@info#1{%
       \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                   {LaTeX Font Info: \space\space\space#1}}%
 \def\@font@warning#1{%
       \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                   {LaTeX Font Warning: #1}}
\fi
```

10.6 Encoding of the 'operators' font

We start with evaluating the encoding to be used for operator names, `\mathrm`, `\mathsf` and `\mathtt` and store it in the macro `\operator@encoding`:

```
\edef\operator@encoding{\encodingdefault}
\def\@tempa{T1}
\ifx\operator@encoding\@tempa
\else
\def\@tempa{OT1}
\ifx\operator@encoding\@tempa
\else
\def\@tempa{LY1}
\ifx\operator@encoding\@tempa
\else
```

If the default encoding is not OT1, T1 or LY1, the package uses T1 for operator names etc. Assuming that the roman font family exists with this encoding is rather safe, yet a warning is issued:

```
  \def\operator@encoding{T1}
  \PackageWarningNoLine{lucimatx}%
  {Using T1 encoding for the math 'operators' font\MessageBreak
   regardless of the default text font encoding (\encodingdefault)}
\fi
\fi
\fi
```

10.7 Math font definitions

Instead of relying on external fd files, the package declares the math font families internally. Thus, potential problems with obsolete fd files using Y&Y-style font names are avoided:

```
\DeclareFontEncoding{LMR}{}{}
\DeclareFontSubstitution{LMR}{hlcm}{m}{n}

\DeclareFontFamily{OML}{hlcm}{\skewchar\font=127}
\DeclareLucidaFontShape{OML}{hlcm}{m}{n}{hlcrm}{}
```

```
\DeclareLucidaFontShape{OML}{hlcm}{b}{n}{hlcdm}{}
\DeclareLucidaFontShape{OML}{hlcm}{m}{it}{hlcrima}{}
\DeclareLucidaFontShape{OML}{hlcm}{b}{it}{hlcdima}{}
\DeclareLucidaFontShape{OML}{hlcm}{m}{itx}{hlcrim}{}
\DeclareLucidaFontShape{OML}{hlcm}{b}{itx}{hlcdim}{}

\DeclareFontFamily{OMS}{hlcy}{\skewchar\font=48}
\DeclareLucidaFontShape{OMS}{hlcy}{m}{n}{hlcry}{}
\DeclareLucidaFontShape{OMS}{hlcy}{b}{n}{hlcdy}{}

\DeclareFontFamily{OMX}{hlcv}{}
\DeclareLucidaFontShape{OMX}{hlcv}{m}{n}{hlcrv}{}

\DeclareFontFamily{LMR}{hlcm}{}
\DeclareLucidaFontShape{LMR}{hlcm}{m}{n}{hlcra}{}
\DeclareLucidaFontShape{LMR}{hlcm}{b}{n}{hlcda}{}
```

10.8 Math font setup

Beside the standard symbol fonts and math alphabets, two additional symbol fonts named 'arrows' and 'upright' will be declared. They hold the plethora of additional (AMS) symbols as well as the upright math alphabets.

```
\DeclareSymbolFont{letters}{OML}{hlcm}{m}{\letters@shape}
\DeclareSymbolFont{upright}{OML}{hlcm}{m}{n}
\DeclareSymbolFont{symbols}{OMS}{hlcy}{m}{n}
\DeclareSymbolFont{largesymbols}{OMX}{hlcv}{m}{n}
\SetSymbolFont{letters}{bold}{OML}{hlcm}{b}{\letters@shape}
\SetSymbolFont{upright}{bold}{OML}{hlcm}{b}{n}
\SetSymbolFont{symbols}{bold}{OMS}{hlcy}{b}{n}
\DeclareSymbolFont{arrows}{LMR}{hlcm}{m}{n}
\SetSymbolFont{arrows}{bold}{LMR}{hlcm}{b}{n}
\DeclareSymbolFont{operators}{\operator@encoding}{\rmdefault}{m}{n}
\SetSymbolFont{operators}{bold}{\operator@encoding}{\rmdefault}{b}{n}
\DeclareMathAlphabet\mathbf  \operator@encoding{\rmdefault}{b}{n}
\DeclareMathAlphabet\mbf     {OML}{hlcm}{b}{n}
\DeclareMathAlphabet\mathrm  \operator@encoding{\rmdefault}{m}{n}
\DeclareMathAlphabet\mathsf  \operator@encoding{\sfdefault}{m}{n}
\DeclareMathAlphabet\mathit  \operator@encoding{\rmdefault}{m}{it}
\DeclareMathAlphabet\mathtt  \operator@encoding{\ttdefault}{m}{n}
\DeclareMathAlphabet\mathfrak\operator@encoding{hlcf}{m}{n}
\SetMathAlphabet{\mathsf}{bold}{\operator@encoding}{\sfdefault}{b}{n}
\SetMathAlphabet{\mathrm}{bold}{\operator@encoding}{\rmdefault}{b}{n}
\SetMathAlphabet{\mathit}{bold}{\operator@encoding}{\rmdefault}{b}{it}
\SetMathAlphabet{\mathtt}{bold}{\operator@encoding}{\ttdefault}{b}{n}
\DeclareSymbolFontAlphabet{\mathbb}{arrows}
\DeclareSymbolFontAlphabet{\mathcal}{symbols}
```

10.9 Digits

Make digits come from the upright math font, if required:

```
\ifLbt@stdmathdigits
  \DeclareMathSymbol{0}\mathalpha{upright}{"30}
  \DeclareMathSymbol{1}\mathalpha{upright}{"31}
  \DeclareMathSymbol{2}\mathalpha{upright}{"32}
```

```
  \DeclareMathSymbol{3}\mathalpha{upright}{"33}
  \DeclareMathSymbol{4}\mathalpha{upright}{"34}
  \DeclareMathSymbol{5}\mathalpha{upright}{"35}
  \DeclareMathSymbol{6}\mathalpha{upright}{"36}
  \DeclareMathSymbol{7}\mathalpha{upright}{"37}
  \DeclareMathSymbol{8}\mathalpha{upright}{"38}
  \DeclareMathSymbol{9}\mathalpha{upright}{"39}
\fi
```

10.10 Math accents

The definitions of the math accents must take the `\operator@encoding` into account, which is no longer guaranteed to be OT1 (as with standard LaTeX).

```
\def\@tempa{T1}
\ifx\operator@encoding\@tempa
  \DeclareMathAccent{\acute}{\mathalpha}{operators}{1}
  \DeclareMathAccent{\grave}{\mathalpha}{operators}{0}
  \DeclareMathAccent{\ddot}{\mathalpha}{operators}{4}
  \DeclareMathAccent{\tilde}{\mathalpha}{operators}{3}
  \DeclareMathAccent{\bar}{\mathalpha}{operators}{9}
  \DeclareMathAccent{\breve}{\mathalpha}{operators}{8}
  \DeclareMathAccent{\check}{\mathalpha}{operators}{7}
  \DeclareMathAccent{\hat}{\mathalpha}{operators}{2}
  \DeclareMathAccent{\dot}{\mathalpha}{operators}{10}
  \DeclareMathAccent{\mathring}{\mathalpha}{operators}{6}
\else
\def\@tempa{OT1}
\ifx\operator@encoding\@tempa
  \DeclareMathAccent{\acute}{\mathalpha}{operators}{19}
  \DeclareMathAccent{\grave}{\mathalpha}{operators}{18}
  \DeclareMathAccent{\ddot}{\mathalpha}{operators}{127}
  \DeclareMathAccent{\tilde}{\mathalpha}{operators}{126}
  \DeclareMathAccent{\bar}{\mathalpha}{operators}{22}
  \DeclareMathAccent{\breve}{\mathalpha}{operators}{21}
  \DeclareMathAccent{\check}{\mathalpha}{operators}{20}
  \DeclareMathAccent{\hat}{\mathalpha}{operators}{94}
  \DeclareMathAccent{\dot}{\mathalpha}{operators}{95}
  \DeclareMathAccent{\mathring}{\mathalpha}{operators}{23}
\else % LY1
  \DeclareMathAccent{\acute}{\mathalpha}{operators}{19}
  \DeclareMathAccent{\grave}{\mathalpha}{operators}{18}
  \DeclareMathAccent{\ddot}{\mathalpha}{operators}{127}
  \DeclareMathAccent{\tilde}{\mathalpha}{operators}{126}
  \DeclareMathAccent{\bar}{\mathalpha}{operators}{22}
  \DeclareMathAccent{\breve}{\mathalpha}{operators}{21}
  \DeclareMathAccent{\check}{\mathalpha}{operators}{20}
  \DeclareMathAccent{\hat}{\mathalpha}{operators}{94}
  \DeclareMathAccent{\dot}{\mathalpha}{operators}{5}
  \DeclareMathAccent{\mathring}{\mathalpha}{operators}{23}
\fi\fi
```

Notice that `\mathring` has been added, which was forgotten in lucidabr 4.2.

10.11 Uppercase Latin alphabet

Make uc Latin letters come from the upright font, if required. This can occur only with the option `math-style=french`.

```
\ifLbt@uprightLatin
  \DeclareMathSymbol{A}{\mathalpha}{upright}{`A}
  \DeclareMathSymbol{B}{\mathalpha}{upright}{`B}
  \DeclareMathSymbol{C}{\mathalpha}{upright}{`C}
  \DeclareMathSymbol{D}{\mathalpha}{upright}{`D}
  \DeclareMathSymbol{E}{\mathalpha}{upright}{`E}
  \DeclareMathSymbol{F}{\mathalpha}{upright}{`F}
  \DeclareMathSymbol{G}{\mathalpha}{upright}{`G}
  \DeclareMathSymbol{H}{\mathalpha}{upright}{`H}
  \DeclareMathSymbol{I}{\mathalpha}{upright}{`I}
  \DeclareMathSymbol{J}{\mathalpha}{upright}{`J}
  \DeclareMathSymbol{K}{\mathalpha}{upright}{`K}
  \DeclareMathSymbol{L}{\mathalpha}{upright}{`L}
  \DeclareMathSymbol{M}{\mathalpha}{upright}{`M}
  \DeclareMathSymbol{N}{\mathalpha}{upright}{`N}
  \DeclareMathSymbol{O}{\mathalpha}{upright}{`O}
  \DeclareMathSymbol{P}{\mathalpha}{upright}{`P}
  \DeclareMathSymbol{Q}{\mathalpha}{upright}{`Q}
  \DeclareMathSymbol{R}{\mathalpha}{upright}{`R}
  \DeclareMathSymbol{S}{\mathalpha}{upright}{`S}
  \DeclareMathSymbol{T}{\mathalpha}{upright}{`T}
  \DeclareMathSymbol{U}{\mathalpha}{upright}{`U}
  \DeclareMathSymbol{V}{\mathalpha}{upright}{`V}
  \DeclareMathSymbol{W}{\mathalpha}{upright}{`W}
  \DeclareMathSymbol{X}{\mathalpha}{upright}{`X}
  \DeclareMathSymbol{Y}{\mathalpha}{upright}{`Y}
  \DeclareMathSymbol{Z}{\mathalpha}{upright}{`Z}
\fi
```

Otherwise, stay with the default definitions from the LaTeX kernel.

10.12 Uppercase Greek alphabet

Provide macros for upright letters. These macro names comply with many other packages:

```
\DeclareMathSymbol{\upGamma}{\mathord}{upright}{0}
\DeclareMathSymbol{\upDelta}{\mathord}{upright}{1}
\DeclareMathSymbol{\upTheta}{\mathord}{upright}{2}
\DeclareMathSymbol{\upLambda}{\mathord}{upright}{3}
\DeclareMathSymbol{\upXi}{\mathord}{upright}{4}
\DeclareMathSymbol{\upPi}{\mathord}{upright}{5}
\DeclareMathSymbol{\upSigma}{\mathord}{upright}{6}
\DeclareMathSymbol{\upUpsilon}{\mathord}{upright}{7}
\DeclareMathSymbol{\upPhi}{\mathord}{upright}{8}
\DeclareMathSymbol{\upPsi}{\mathord}{upright}{9}
\DeclareMathSymbol{\upOmega}{\mathord}{upright}{10}
```

Now decide whether `\Gamma`, `\Delta` etc. are to print upright or cursive letters:

```
\ifLbt@uprightGreek
  \let\Gamma\upGamma
```

```
  \let\Delta\upDelta
  \let\Theta\upTheta
  \let\Lambda\upLambda
  \let\Xi\upXi
  \let\Pi\upPi
  \let\Sigma\upSigma
  \let\Upsilon\upUpsilon
  \let\Phi\upPhi
  \let\Psi\upPsi
  \let\Omega\upOmega
\else
  \DeclareMathSymbol{\Gamma}{\mathord}{letters}{0}
  \DeclareMathSymbol{\Delta}{\mathord}{letters}{1}
  \DeclareMathSymbol{\Theta}{\mathord}{letters}{2}
  \DeclareMathSymbol{\Lambda}{\mathord}{letters}{3}
  \DeclareMathSymbol{\Xi}{\mathord}{letters}{4}
  \DeclareMathSymbol{\Pi}{\mathord}{letters}{5}
  \DeclareMathSymbol{\Sigma}{\mathord}{letters}{6}
  \DeclareMathSymbol{\Upsilon}{\mathord}{letters}{7}
  \DeclareMathSymbol{\Phi}{\mathord}{letters}{8}
  \DeclareMathSymbol{\Psi}{\mathord}{letters}{9}
  \DeclareMathSymbol{\Omega}{\mathord}{letters}{10}
\fi
```

10.13 Lowercase Greek alphabet

Provide macros for upright letters. These macro names comply with many other packages:

```
\DeclareMathSymbol{\upalpha}{\mathord}{upright}{11}
\DeclareMathSymbol{\upbeta}{\mathord}{upright}{12}
\DeclareMathSymbol{\upgamma}{\mathord}{upright}{13}
\DeclareMathSymbol{\updelta}{\mathord}{upright}{14}
\DeclareMathSymbol{\upepsilon}{\mathord}{upright}{15}
\DeclareMathSymbol{\upzeta}{\mathord}{upright}{16}
\DeclareMathSymbol{\upeta}{\mathord}{upright}{17}
\DeclareMathSymbol{\uptheta}{\mathord}{upright}{18}
\DeclareMathSymbol{\upiota}{\mathord}{upright}{19}
\DeclareMathSymbol{\upkappa}{\mathord}{upright}{20}
\DeclareMathSymbol{\uplambda}{\mathord}{upright}{21}
\DeclareMathSymbol{\upmu}{\mathord}{upright}{22}
\DeclareMathSymbol{\upnu}{\mathord}{upright}{23}
\DeclareMathSymbol{\upxi}{\mathord}{upright}{24}
\DeclareMathSymbol{\uppi}{\mathord}{upright}{25}
\DeclareMathSymbol{\uprho}{\mathord}{upright}{26}
\DeclareMathSymbol{\upsigma}{\mathord}{upright}{27}
\DeclareMathSymbol{\uptau}{\mathord}{upright}{28}
\DeclareMathSymbol{\upupsilon}{\mathord}{upright}{29}
\DeclareMathSymbol{\upphi}{\mathord}{upright}{30}
\DeclareMathSymbol{\upchi}{\mathord}{upright}{31}
\DeclareMathSymbol{\uppsi}{\mathord}{upright}{32}
\DeclareMathSymbol{\upomega}{\mathord}{upright}{33}
\DeclareMathSymbol{\upvarepsilon}{\mathord}{upright}{34}
\DeclareMathSymbol{\upvartheta}{\mathord}{upright}{35}
```

```
\DeclareMathSymbol{\upvarpi}{\mathord}{upright}{36}
\DeclareMathSymbol{\upvarrho}{\mathord}{upright}{37}
\DeclareMathSymbol{\upvarsigma}{\mathord}{upright}{38}
\DeclareMathSymbol{\upvarphi}{\mathord}{upright}{39}
```

With the 'French' math style, `\alpha`, `\beta` etc. are to print upright letters:

```
\ifLbt@uprightgreek
  \let\alpha\upalpha
  \let\beta\upbeta
  \let\gamma\upgamma
  \let\delta\updelta
  \let\epsilon\upepsilon
  \let\zeta\upzeta
  \let\eta\upeta
  \let\theta\upeta
  \let\iota\upiota
  \let\kappa\upkappa
  \let\lambda\uplambda
  \let\mu\upmu
  \let\nu\upnu
  \let\xi\upxi
  \let\pi\uppi
  \let\rho\uprho
  \let\sigma\upsigma
  \let\tau\uptau
  \let\upsilon\upupsilon
  \let\phi\upphi
  \let\chi\upchi
  \let\psi\uppsi
  \let\omega\upomega
  \let\varepsilon\upvarepsilon
  \let\vartheta\upvartheta
  \let\varpi\upvarpi
  \let\varrho\upvarrho
  \let\varsigma\upvarsigma
  \let\varphi\upvarphi
\fi
```

Otherwise we can stay with the default definitions supplied by the LaTeX kernel.

10.14 Lucida-specific patches and additions

These symbols come normally from the 'operators' font. Lucida New Math provides them in the 'letters' font, too. That's where we take them from, because the 'operators' font may vary:

```
\DeclareMathDelimiter{[}{\mathopen} {letters}{134}{largesymbols}{2}
\DeclareMathDelimiter{]}{\mathclose}{letters}{135}{largesymbols}{3}
\DeclareMathDelimiter{(}{\mathopen} {letters}{132}{largesymbols}{0}
\DeclareMathDelimiter{)}{\mathclose}{letters}{133}{largesymbols}{1}
\DeclareMathDelimiter{/}{\mathord}{letters}{61}{largesymbols}{14}
\DeclareMathSymbol{=}{\mathrel} {symbols}{131}
\DeclareMathSymbol{+}{\mathbin} {symbols}{130}
```

The following stuff is adopted from lucidabr. To close up gaps in special math characters constructed from pieces:

```
\def\joinrel{\mathrel{\mkern-4mu}}
```

Define some extra large sizes — always done using extensible parts:

```
\def\bigggl{\mathopen\bigg}
\def\bigggr{\mathclose\bigg}
\def\Bigggl{\mathopen\Bigg}
\def\Bigggr{\mathclose\Bigg}
```

Make open face brackets accessible, i.e. [[and]]:

```
\DeclareMathDelimiter{\ldbrack}
  {\mathopen}{letters}{130}{largesymbols}{130}
\DeclareMathDelimiter{\rdbrack}
  {\mathclose}{letters}{131}{largesymbols}{131}
```

Provide access to surface integral signs (linked from text to display size):

```
\DeclareMathSymbol{\surfintop}{\mathop}{largesymbols}{144}
```

lucidabr would also declare a number of 'medium' size integrals, which cannot change their size between text and display mode. IMHO, they should not be used with TEX, so I have omitted them; after all, they were never documented, anyway.

Some characters that need construction in CM exist complete in Lucida New Math:

```
\let\notin\undefined
\let\bowtie\undefined
\let\models\undefined
\let\cong\undefined
\let\angle\undefined
\DeclareMathSymbol{\notin}{\mathrel}{arrows}{29}
\DeclareMathSymbol{\bowtie}{\mathrel}{letters}{246}
\DeclareMathSymbol{\models}{\mathrel}{symbols}{238}
\DeclareMathSymbol{\cong}{\mathrel}{symbols}{155}
\DeclareMathSymbol{\angle}{\mathord}{symbols}{139}
\let\rightleftharpoons\undefined
\DeclareMathSymbol{\rightleftharpoons}{\mathrel}{arrows}{122}
\let\neq\undefined
\DeclareMathSymbol{\neq}{\mathrel}{arrows}{148}
```

These need undefining, too:

```
\let\sqsubset\undefined
\let\sqsupset\undefined
```

Adopted from lucidabr (why?):

```
\def\mathstrut{\vphantom{f}}
```

Here are some extra definitions of mathematical symbols and operators. Let's start with those that were in lucidabr, too:

```
\DeclareMathSymbol{\defineeq}{\mathrel}{symbols}{214}
\let\defineequal=\defineeq
\DeclareMathSymbol{\notequiv}{\mathrel}{arrows}{149}
\let\nequiv=\notequiv
\DeclareMathSymbol{\notapprox}{\mathrel}{arrows}{152}
\let\napprox=\notapprox
\DeclareMathSymbol{\notasymp}{\mathrel}{arrows}{243}
\let\nasymp=\notasymp
\DeclareMathSymbol{\notsubset}{\mathrel}{arrows}{198}
\let\nsubset=\notsubset
```

```
\DeclareMathSymbol{\notsupset}{\mathrel}{arrows}{199}
\let\nsupset=\notsupset
\DeclareMathSymbol{\notsubseteq}{\mathrel}{arrows}{200}
\let\nsubseteq=\notsubseteq
\DeclareMathSymbol{\notsupseteq}{\mathrel}{arrows}{201}
\let\nsupseteq=\notsupseteq
\DeclareMathSymbol{\notsimeq}{\mathrel}{arrows}{151}
\let\nsimeq=\notsimeq
\DeclareMathSymbol{\notsqsubseteq}{\mathrel}{arrows}{212}
\let\nsqsubseteq=\notsqsubseteq
\DeclareMathSymbol{\notsqsupseteq}{\mathrel}{arrows}{213}
\let\nsqsupseteq=\notsqsupseteq
\DeclareMathSymbol{\notcong}{\mathrel}{arrows}{153}
\let\ncong=\notcong
\DeclareMathSymbol{\notni}{\mathrel}{arrows}{31}
\DeclareMathSymbol{\leadsfrom}{\mathrel}{arrows}{141}
\let\leftsquigarrow\leadsfrom
\DeclareMathSymbol{\leftrightharpoons}{\mathrel}{arrows}{121}
% \DeclareMathSymbol{\arrowaxisright}{\mathord}{arrows}{55} % ?
% \DeclareMathSymbol{\arrowaxisleft}{\mathord}{arrows}{54} % ?
% \DeclareMathSymbol{\axisshort}{\mathord}{arrows}{45} % ?
\DeclareMathSymbol{\dashuparrow}{\mathord}{arrows}{57}
\DeclareMathSymbol{\dashdownarrow}{\mathord}{arrows}{59}
\DeclareMathSymbol{\image}{\mathrel}{letters}{198}
\DeclareMathSymbol{\original}{\mathrel}{letters}{197}
```

The following symbols were not available in lucidabr:

```
\DeclareRobustCommand{\longoriginal}{%
  \mathrel\circ\mkern-.07mu\relbar\relbar\mkern-.07mu\mathrel\bullet}
\DeclareRobustCommand{\longimage}{%
  \mathrel\bullet\mkern-.07mu\relbar\relbar\mkern-.07mu\mathrel\circ}
```

The 0.07 em were found by trial and error; without these, tiny gaps would remain between the shaft and the bullet or circle. (Why?)

```
\DeclareMathSymbol{\updownarrows}{\mathrel}{arrows}{111}
\DeclareMathSymbol{\downuparrows}{\mathrel}{arrows}{112}
\DeclareMathSymbol{\hateq}{\mathrel}{symbols}{210}
\DeclareMathSymbol{\coloneq}{\mathrel}{symbols}{205}
\DeclareMathSymbol{\eqcolon}{\mathrel}{symbols}{206}
```

The default \Leftrightarrow in Lucida New Math is ugly. The Lucida New Math fonts include also a longer one, which was already used by the lucidabr package, even though its length does not match the one-directional arrows:

```
\let\Leftrightarrow\undefined
\DeclareMathSymbol{\Leftrightarrow}{\mathrel}{arrows}{97}
```

10.15 AMS symbols

MSAM equivalents:

```
\DeclareMathSymbol{\boxdot}{\mathbin}{symbols}{237}
\DeclareMathSymbol{\boxplus}{\mathbin}{symbols}{234}
\DeclareMathSymbol{\boxtimes}{\mathbin}{symbols}{236}
\DeclareMathSymbol{\square}{\mathord}{arrows}{2}
\DeclareMathSymbol{\blacksquare}{\mathord}{arrows}{3}
```

```
\DeclareMathSymbol{\centerdot}{\mathbin}{arrows}{225}
\DeclareMathSymbol{\lozenge}{\mathord}{arrows}{8}
\DeclareMathSymbol{\blacklozenge}{\mathord}{arrows}{9}
\DeclareMathSymbol{\circlearrowright}{\mathrel}{arrows}{140}
\DeclareMathSymbol{\circlearrowleft}{\mathrel}{arrows}{139}
\DeclareMathSymbol{\rightleftharpoons}{\mathrel}{arrows}{122}
\DeclareMathSymbol{\leftrightharpoons}{\mathrel}{arrows}{121}
\DeclareMathSymbol{\boxminus}{\mathbin}{symbols}{235}
\DeclareMathSymbol{\Vdash}{\mathrel}{symbols}{240}
\DeclareMathSymbol{\Vvdash}{\mathrel}{letters}{211}
\DeclareMathSymbol{\vDash}{\mathrel}{symbols}{238}
\DeclareMathSymbol{\twoheadrightarrow}{\mathrel}{arrows}{37}
\DeclareMathSymbol{\twoheadleftarrow}{\mathrel}{arrows}{35}
\DeclareMathSymbol{\leftleftarrows}{\mathrel}{arrows}{113}
\DeclareMathSymbol{\rightrightarrows}{\mathrel}{arrows}{115}
\DeclareMathSymbol{\upuparrows}{\mathrel}{arrows}{114}
\DeclareMathSymbol{\downdownarrows}{\mathrel}{arrows}{116}
\DeclareMathSymbol{\upharpoonright}{\mathrel}{arrows}{117}
\DeclareMathSymbol{\downharpoonright}{\mathrel}{arrows}{119}
\DeclareMathSymbol{\upharpoonleft}{\mathrel}{arrows}{118}
\DeclareMathSymbol{\downharpoonleft}{\mathrel}{arrows}{120}
\DeclareMathSymbol{\rightarrowtail}{\mathrel}{arrows}{41}
\DeclareMathSymbol{\leftarrowtail}{\mathrel}{arrows}{40}
\DeclareMathSymbol{\leftrightarrows}{\mathrel}{arrows}{110}
\DeclareMathSymbol{\rightleftarrows}{\mathrel}{arrows}{109}
\DeclareMathSymbol{\Lsh}{\mathrel}{arrows}{123}
\DeclareMathSymbol{\Rsh}{\mathrel}{arrows}{125}
\DeclareMathSymbol{\rightsquigarrow}{\mathrel}{arrows}{142}
\DeclareMathSymbol{\leftrightsquigarrow}{\mathrel}{arrows}{145}
\DeclareMathSymbol{\looparrowleft}{\mathrel}{arrows}{63}
\DeclareMathSymbol{\looparrowright}{\mathrel}{arrows}{64}
\DeclareMathSymbol{\circeq}{\mathrel}{symbols}{208}
\DeclareMathSymbol{\succsim}{\mathrel}{symbols}{225}
\DeclareMathSymbol{\gtrsim}{\mathrel}{symbols}{221}
\DeclareMathSymbol{\gtrapprox}{\mathrel}{letters}{219}
\DeclareMathSymbol{\multimap}{\mathrel}{letters}{199}
\DeclareMathSymbol{\therefore}{\mathrel}{symbols}{144}
\DeclareMathSymbol{\because}{\mathrel}{symbols}{145}
\DeclareMathSymbol{\doteqdot}{\mathrel}{symbols}{202}
\DeclareMathSymbol{\triangleq}{\mathrel}{symbols}{213}
\DeclareMathSymbol{\precsim}{\mathrel}{symbols}{224}
\DeclareMathSymbol{\lesssim}{\mathrel}{symbols}{220}
\DeclareMathSymbol{\lessapprox}{\mathrel}{letters}{218}
\DeclareMathSymbol{\eqslantless}{\mathrel}{letters}{226}
\DeclareMathSymbol{\eqslantgtr}{\mathrel}{letters}{227}
\DeclareMathSymbol{\curlyeqprec}{\mathrel}{letters}{230}
\DeclareMathSymbol{\curlyeqsucc}{\mathrel}{letters}{231}
\DeclareMathSymbol{\preccurlyeq}{\mathrel}{letters}{228}
\DeclareMathSymbol{\leqq}{\mathrel}{symbols}{218}
\DeclareMathSymbol{\leqslant}{\mathrel}{letters}{224}
\DeclareMathSymbol{\lessgtr}{\mathrel}{symbols}{222}
\DeclareMathSymbol{\backprime}{\mathord}{letters}{200}
\DeclareMathSymbol{\risingdotseq}{\mathrel}{symbols}{204}
\DeclareMathSymbol{\fallingdotseq}{\mathrel}{symbols}{203}
```

```
\DeclareMathSymbol{\succcurlyeq}{\mathrel}{letters}{229}
\DeclareMathSymbol{\geqq}{\mathrel}{symbols}{219}
\DeclareMathSymbol{\geqslant}{\mathrel}{letters}{225}
\DeclareMathSymbol{\gtrless}{\mathrel}{symbols}{223}
\DeclareMathSymbol{\sqsubset}{\mathrel}{symbols}{228}
\DeclareMathSymbol{\sqsupset}{\mathrel}{symbols}{229}
\DeclareMathSymbol{\vartriangleright}{\mathrel}{letters}{46}
\DeclareMathSymbol{\vartriangleleft}{\mathrel}{letters}{47}
\DeclareMathSymbol{\trianglerighteq}{\mathrel}{symbols}{245}
\DeclareMathSymbol{\trianglelefteq}{\mathrel}{symbols}{244}
\DeclareMathSymbol{\bigstar}{\mathord}{arrows}{171}
\DeclareMathSymbol{\between}{\mathrel}{letters}{242}
\DeclareMathSymbol{\blacktriangledown}{\mathord}{arrows}{7}
\DeclareMathSymbol{\blacktriangleright}{\mathrel}{letters}{241}
\DeclareMathSymbol{\blacktriangleleft}{\mathrel}{letters}{240}
\DeclareMathSymbol{\vartriangle}{\mathrel}{arrows}{4}
\DeclareMathSymbol{\blacktriangle}{\mathord}{arrows}{5}
\DeclareMathSymbol{\triangledown}{\mathord}{arrows}{6}
\DeclareMathSymbol{\eqcirc}{\mathrel}{symbols}{207}
\DeclareMathSymbol{\lesseqgtr}{\mathrel}{letters}{232}
\DeclareMathSymbol{\gtreqless}{\mathrel}{letters}{233}
\DeclareMathSymbol{\lesseqqgtr}{\mathrel}{letters}{234}
\DeclareMathSymbol{\gtreqqless}{\mathrel}{letters}{235}
\DeclareMathSymbol{\Rrightarrow}{\mathrel}{arrows}{108}
\DeclareMathSymbol{\Lleftarrow}{\mathrel}{arrows}{106}
\DeclareMathSymbol{\veebar}{\mathbin}{letters}{210}
\DeclareMathSymbol{\barwedge}{\mathbin}{symbols}{246}
\DeclareMathSymbol{\doublebarwedge}{\mathbin}{symbols}{212}
\DeclareMathSymbol{\angle}{\mathord}{symbols}{139}
\DeclareMathSymbol{\measuredangle}{\mathord}{symbols}{140}
\DeclareMathSymbol{\sphericalangle}{\mathord}{symbols}{141}
\DeclareMathSymbol{\varpropto}{\mathrel}{symbols}{47}
\DeclareMathSymbol{\smallsmile}{\mathrel}{letters}{94} % ?
\DeclareMathSymbol{\smallfrown}{\mathrel}{letters}{95} % ?
\DeclareMathSymbol{\Subset}{\mathrel}{symbols}{248}
\DeclareMathSymbol{\Supset}{\mathrel}{symbols}{249}
\DeclareMathSymbol{\Cup}{\mathbin}{symbols}{250}
\DeclareMathSymbol{\Cap}{\mathbin}{symbols}{251}
\DeclareMathSymbol{\curlywedge}{\mathbin}{symbols}{132}
\DeclareMathSymbol{\curlyvee}{\mathbin}{symbols}{133}
\DeclareMathSymbol{\leftthreetimes}{\mathbin}{letters}{208}
\DeclareMathSymbol{\rightthreetimes}{\mathbin}{letters}{209}
\DeclareMathSymbol{\subseteqq}{\mathrel}{letters}{238}
\DeclareMathSymbol{\supseteqq}{\mathrel}{letters}{239}
\DeclareMathSymbol{\bumpeq}{\mathrel}{symbols}{200}
\DeclareMathSymbol{\Bumpeq}{\mathrel}{symbols}{199}
\DeclareMathSymbol{\lll}{\mathrel}{letters}{222}
\DeclareMathSymbol{\ggg}{\mathrel}{letters}{223}
\DeclareMathSymbol{\circledS}{\mathord}{letters}{202}
\DeclareMathSymbol{\pitchfork}{\mathrel}{letters}{243}
\DeclareMathSymbol{\dotplus}{\mathbin}{symbols}{137}
\DeclareMathSymbol{\backsim}{\mathrel}{letters}{248}
\DeclareMathSymbol{\backsimeq}{\mathrel}{letters}{249}
\DeclareMathSymbol{\complement}{\mathord}{letters}{148}
```

```
\DeclareMathSymbol{\intercal}{\mathbin}{letters}{217}
\DeclareMathSymbol{\circledcirc}{\mathbin}{symbols}{230}
\DeclareMathSymbol{\circledast}{\mathbin}{symbols}{231}
\DeclareMathSymbol{\circleddash}{\mathbin}{letters}{204}
\DeclareMathSymbol{\dashrightarrow}{\mathord}{arrows}{58}
\DeclareMathSymbol{\dashleftarrow}{\mathord}{arrows}{56}
\let\dasharrow\dashrightarrow
\DeclareMathDelimiter\ulcorner{\mathopen}{arrows}{91}{arrows}{91}
\DeclareMathDelimiter\urcorner{\mathclose}{arrows}{92}{arrows}{92}
\DeclareMathDelimiter\llcorner{\mathopen}{arrows}{93}{arrows}{93}
\DeclareMathDelimiter\lrcorner{\mathclose}{arrows}{94}{arrows}{94}
\edef\checkmark{\noexpand\mathhexbox{\hexnumber@\symarrows}AC}
\edef\circledR{\noexpand\mathhexbox{\hexnumber@\symletters}C9}
\edef\maltese{\noexpand\mathhexbox{\hexnumber@\symletters}CB}
```

NB: lucidabr 4.2 was lacking \doublebarwedge.

MSBM equivalents:

```
\DeclareMathSymbol{\lvertneqq}{\mathrel}{arrows}{222}
\DeclareMathSymbol{\gvertneqq}{\mathrel}{arrows}{223}
\DeclareMathSymbol{\nleq}{\mathrel}{arrows}{156}
\DeclareMathSymbol{\ngeq}{\mathrel}{arrows}{157}
\DeclareMathSymbol{\nless}{\mathrel}{arrows}{154}
\DeclareMathSymbol{\ngtr}{\mathrel}{arrows}{155}
\DeclareMathSymbol{\nprec}{\mathrel}{arrows}{229}
\DeclareMathSymbol{\nsucc}{\mathrel}{arrows}{230}
\DeclareMathSymbol{\lneqq}{\mathrel}{arrows}{220}
\DeclareMathSymbol{\gneqq}{\mathrel}{arrows}{221}
\DeclareMathSymbol{\nleqslant}{\mathrel}{arrows}{214}
\DeclareMathSymbol{\ngeqslant}{\mathrel}{arrows}{215}
\DeclareMathSymbol{\lneq}{\mathrel}{arrows}{218}
\DeclareMathSymbol{\gneq}{\mathrel}{arrows}{219}
\DeclareMathSymbol{\npreceq}{\mathrel}{arrows}{231}
\DeclareMathSymbol{\nsucceq}{\mathrel}{arrows}{232}
\DeclareMathSymbol{\precnsim}{\mathrel}{arrows}{235}
\DeclareMathSymbol{\succnsim}{\mathrel}{arrows}{236}
\DeclareMathSymbol{\lnsim}{\mathrel}{arrows}{224}
\DeclareMathSymbol{\gnsim}{\mathrel}{arrows}{226}
\DeclareMathSymbol{\nleqq}{\mathrel}{arrows}{216}
\DeclareMathSymbol{\ngeqq}{\mathrel}{arrows}{217}
\DeclareMathSymbol{\precneqq}{\mathrel}{arrows}{233}
\DeclareMathSymbol{\succneqq}{\mathrel}{arrows}{234}
\DeclareMathSymbol{\precnapprox}{\mathrel}{arrows}{237}
\DeclareMathSymbol{\succnapprox}{\mathrel}{arrows}{238}
\DeclareMathSymbol{\lnapprox}{\mathrel}{arrows}{227}
\DeclareMathSymbol{\gnapprox}{\mathrel}{arrows}{228}
\DeclareMathSymbol{\nsim}{\mathrel}{arrows}{150}
\DeclareMathSymbol{\ncong}{\mathrel}{arrows}{153}
\DeclareMathSymbol{\diagup}{\mathrel}{arrows}{11}
\DeclareMathSymbol{\diagdown}{\mathrel}{arrows}{12}
\DeclareMathSymbol{\varsubsetneq}{\mathrel}{arrows}{208}
\DeclareMathSymbol{\varsupsetneq}{\mathrel}{arrows}{209}
\DeclareMathSymbol{\nsubseteqq}{\mathrel}{arrows}{202}
\DeclareMathSymbol{\nsupseteqq}{\mathrel}{arrows}{203}
\DeclareMathSymbol{\subsetneqq}{\mathrel}{arrows}{206}
```

```
\DeclareMathSymbol{\supsetneqq}{\mathrel}{arrows}{207}
\DeclareMathSymbol{\varsubsetneqq}{\mathrel}{arrows}{210}
\DeclareMathSymbol{\varsupsetneqq}{\mathrel}{arrows}{211}
\DeclareMathSymbol{\subsetneq}{\mathrel}{arrows}{204}
\DeclareMathSymbol{\supsetneq}{\mathrel}{arrows}{205}
\DeclareMathSymbol{\nsubseteq}{\mathrel}{arrows}{200}
\DeclareMathSymbol{\nsupseteq}{\mathrel}{arrows}{201}
\DeclareMathSymbol{\nparallel}{\mathrel}{arrows}{247}
\DeclareMathSymbol{\nmid}{\mathrel}{arrows}{246}
\DeclareMathSymbol{\nshortmid}{\mathrel}{arrows}{244}
\DeclareMathSymbol{\nshortparallel}{\mathrel}{arrows}{245}
\DeclareMathSymbol{\nvdash}{\mathrel}{arrows}{248}
\DeclareMathSymbol{\nVdash}{\mathrel}{arrows}{250}
\DeclareMathSymbol{\nvDash}{\mathrel}{arrows}{249}
\DeclareMathSymbol{\nVDash}{\mathrel}{arrows}{251}
\DeclareMathSymbol{\ntrianglerighteq}{\mathrel}{arrows}{242}
\DeclareMathSymbol{\ntriangleleftеq}{\mathrel}{arrows}{241}
\DeclareMathSymbol{\ntriangleleft}{\mathrel}{arrows}{239}
\DeclareMathSymbol{\ntriangleright}{\mathrel}{arrows}{240}
\DeclareMathSymbol{\nleftarrow}{\mathrel}{arrows}{50}
\DeclareMathSymbol{\nrightarrow}{\mathrel}{arrows}{51}
\DeclareMathSymbol{\nLeftarrow}{\mathrel}{arrows}{102}
\DeclareMathSymbol{\nRightarrow}{\mathrel}{arrows}{104}
\DeclareMathSymbol{\nLeftrightarrow}{\mathrel}{arrows}{103}
\DeclareMathSymbol{\nleftrightarrow}{\mathrel}{arrows}{52}
\DeclareMathSymbol{\divideontimes}{\mathbin}{letters}{247}
```

`\emptyset` and `\varnothing` have changed their places, as compared with Computer Modern:

```
\DeclareMathSymbol{\emptyset}{\mathord}{letters}{156}
\DeclareMathSymbol{\varnothing}{\mathord}{symbols}{59}
```

```
\DeclareMathSymbol{\nexists}{\mathord}{arrows}{32}
\DeclareMathSymbol{\Finv}{\mathord}{letters}{144}
\DeclareMathSymbol{\Game}{\mathord}{letters}{145}
\DeclareMathSymbol{\eth}{\mathord}{operators}{240}
\DeclareMathSymbol{\eqsim}{\mathrel}{symbols}{153}
\DeclareMathSymbol{\beth}{\mathord}{letters}{149}
\DeclareMathSymbol{\gimel}{\mathord}{letters}{150}
\DeclareMathSymbol{\daleth}{\mathord}{letters}{151}
\DeclareMathSymbol{\lessdot}{\mathrel}{letters}{220}
\DeclareMathSymbol{\gtrdot}{\mathrel}{letters}{221}
\DeclareMathSymbol{\ltimes}{\mathbin}{letters}{206}
\DeclareMathSymbol{\rtimes}{\mathbin}{letters}{207}
\DeclareMathSymbol{\shortmid}{\mathrel}{letters}{244}
\DeclareMathSymbol{\shortparallel}{\mathrel}{letters}{245}
\DeclareMathSymbol{\smallsetminus}{\mathbin}{letters}{216} %?
\DeclareMathSymbol{\thicksim}{\mathrel}{symbols}{24} %?
\DeclareMathSymbol{\thickapprox}{\mathrel}{symbols}{25} %?
\DeclareMathSymbol{\approxeq}{\mathrel}{symbols}{157}
\DeclareMathSymbol{\succapprox}{\mathrel}{letters}{237}
\DeclareMathSymbol{\precapprox}{\mathrel}{letters}{236}
\DeclareMathSymbol{\curvearrowleft}{\mathrel}{arrows}{135}
\DeclareMathSymbol{\curvearrowright}{\mathrel}{arrows}{136}
%\DeclareMathSymbol{\digamma}{\mathord}{letters}{70} %?
```

```
\DeclareMathSymbol{\varkappa}{\mathord}{letters}{155}
\DeclareMathSymbol{\Bbbk}{\mathord}{arrows}{107}
\DeclareMathSymbol{\hslash}{\mathord}{letters}{157}
\DeclareMathSymbol{\hbar}{\mathord}{arrows}{27}
\DeclareMathSymbol{\backepsilon}{\mathrel}{letters}{251} %?
```

Certain AMS symbols have aliases:

```
\let\restriction\upharpoonright
\let\Doteq\doteqdot
\let\doublecup\Cup
\let\doublecap\Cap
\let\llless\lll
\let\gggtr\ggg
\let\notsim\nsim
```

The lucida fonts are lacking a few AMS symbols. Instead of letting them undefined, we make them result in useful error messages.

```
\newcommand{\digamma}{%
  \PackageError{lucimatx}{%
    The mathematical symbol \protect\digamma\space is not available\MessageBreak
    in the Lucida New Math fonts}%
    {Type <return> to proceed; the command will be ignored.}}
\newcommand{\mathyen}{%
  \PackageError{lucimatx}{%
    The mathematical symbol \yen\space is not available\MessageBreak
    in the Lucida New Math fonts}%
    {Type <return> to proceed; the command will be ignored.}}
\DeclareRobustCommand{\yen}{\ifmmode\mathyen\else\textyen\fi}
```

10.16 Encoding-specific math commands

`\mathsterling` needs to be taken from the 'operators' font, because it is not provided in the the Lucida New Math fonts.

```
\edef\operator@encoding{\encodingdefault}
\def\@tempa{LY1}
\ifx\operator@encoding\@tempa
    \DeclareMathSymbol{\mathsterling}{\mathord}{operators}{163}
\else
\def\@tempa{T1}
\ifx\operator@encoding\@tempa
  \DeclareMathSymbol{\mathsterling}{\mathord}{operators}{191}
\fi
\fi
```

10.17 LaTeX symbols

So-called LaTeX symbols are mapped to the most similar AMS symbol:

```
\let\leadsto\rightsquigarrow
\let\Join\bowtie
\let\Box\square
\let\Diamond\lozenge
\let\rhd\undefined
\let\lhd\undefined
```

```
\let\unlhd\undefined
\let\unrhd\undefined
\DeclareMathSymbol{\rhd}{\mathrel}{letters}{46}
\DeclareMathSymbol{\lhd}{\mathrel}{letters}{47}
\DeclareMathSymbol{\unlhd}{\mathrel}{symbols}{244}
\DeclareMathSymbol{\unrhd}{\mathrel}{symbols}{245}
```

10.18 Deferred declarations

The following is deferred until `\begin{document}`:

```
\AtBeginDocument{%
```

Distinguish whether or not amsmath is used, too:

```
\@ifpackageloaded{amsmath}{%
```

Adopted from lucidabr 4.2:

```
  \def\biggg{\bBigg@\thr@@}
  \def\Biggg{\bBigg@{3.5}}
```

From David Carlisle's reply to psnfss/2437:

```
  \def\rightarrowfill@#1{%
    \m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\mkern4.5mu\mathrel{\copy\z@}%
    \kern-\wd\z@
    \cleaders\hbox{$#1\mkern-2mu\box\z@\mkern-2mu$}\hfill%
    \mkern-4.5mu %
    \rightarrow$}%
  \def\leftarrowfill@#1{%
    \m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\leftarrow
    \mkern-4.5mu %
    \cleaders\hbox{$#1\mkern-2mu\copy\z@\mkern-2mu$}\hfill
    \kern-\wd\z@
    \mathrel{\box\z@}\mkern4.5mu$}
  \def\leftrightarrowfill@#1{\m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\leftarrow
    \mkern-12mu %
    \cleaders\hbox{$#1\mkern-2mu\box\z@\mkern-2mu$}\hfill
    \rightarrow$}%
```

The definition of `\surfint` must work together with the facilities provided by amsmath to control the placement of limits and dots:

```
\def\surfint{\DOTSI\surfintop\ilimits@}
```

The spacing of the multiple integrals provided by amsmath should match the ready-made `\surfint`:

```
\def\intkern@{\mkern-7.4mu\mathchoice{\mkern-.6mu}{}{}{}}
```

Also, placement of limits on multiple integrals must be corrected wrt/ Lucida:

```
\def\ints@b{\mathop\bgroup\let\ilimits@\egroup}%
```

Here comes the code to be used *without* amsmath. The definitions of `\big` & friends are adopted from lucidabr. Actually, they stem from Berthold Horn's `lcdlatex.tex` © 1991 – 1994 Y&Y.

```
}{%
  \def\big#1{{\hbox{$\left#1\vbox to8.20\p@{}\right.\n@space$}}}
```

```
  \def\Big#1{{\hbox{$\left#1\vbox to10.80\p@{}\right.\n@space$}}}
  \def\bigg#1{{\hbox{$\left#1\vbox to13.42\p@{}\right.\n@space$}}}
  \def\Bigg#1{{\hbox{$\left#1\vbox to16.03\p@{}\right.\n@space$}}}
  \def\biggg#1{{\hbox{$\left#1\vbox to17.72\p@{}\right.\n@space$}}}
  \def\Biggg#1{{\hbox{$\left#1\vbox to21.25\p@{}\right.\n@space$}}}
  \def\n@space{\nulldelimiterspace\z@ \m@th}
```

Without amsmath, the surface integral can be declared 'as usual':

```
\def\surfint{\surfintop\nolimits}
```

```
}
```

chemarr is an extension to amsmath. Its commands for extensible arrows must be adapted to Lucida. The value of 6.1 mu was found by trial and error.

```
\@ifpackageloaded{chemarr}{%
  \def\rightharpoonupfill@#1{%
    \m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\mkern6.1mu\mathrel{\copy\z@}%
    \kern-\wd\z@
    \cleaders\hbox{$#1\mkern-2mu\box\z@\mkern-2mu$}\hfill%
    \mkern-6.1mu %
    \rightharpoonup$}%
  \def\leftharpoondownfill@#1{%
    \m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\leftharpoondown
    \mkern-6.1mu %
    \cleaders\hbox{$#1\mkern-2mu\copy\z@\mkern-2mu$}\hfill
    \kern-\wd\z@
    \mathrel{\box\z@}\mkern6.1mu$}
}{}
```

In n-th root, don't want the 'n' to come too close to the radical. (Adopted from lucidabr 4.2). Here, too, we must make sure that a different definition possibly issued by amsmath is overwritten again:

```
\def\r@@t#1#2{\setbox\z@\hbox{$\m@th#1\sqrt{#2}$}%
  \dimen@\ht\z@ \advance\dimen@-\dp\z@
  \mkern5mu\raise.6\dimen@\copy\rootbox \mkern-7.5mu\box\z@}
```

amsmath provides its own definitions for the following symbols, which work for CM only. Make sure that the Lucida-specific definitions are used in the end:

```
\DeclareMathSymbol{\relbar@}{\mathord}{arrows}{45}
\def\relbar{\mathrel{\smash\relbar@}\mathrel{\mkern-2.5mu}}
\DeclareMathSymbol{\Relbar@}{\mathrel}{arrows}{61}
\def\Relbar{\Relbar@\mathrel{\mkern-2.5mu}}
```

The \mkern-2.5mu undoes the bogus 'italic correction' after joiners in LBMA. (Adopted from lucidabr 4.2.)

```
\let\hookleftarrow\undefined
\let\hookrightarrow\undefined
\let\mapsto\undefined
\let\doteq\undefined
\DeclareRobustCommand\longleftarrow{\leftarrow\relbar\mathrel{\mkern4mu}}
\DeclareRobustCommand\longrightarrow{\mathrel{\mkern4mu}\relbar\rightarrow}
\DeclareRobustCommand\Longleftarrow{\Leftarrow\Relbar\mathrel{\mkern4mu}}
\DeclareRobustCommand\Longrightarrow{\mathrel{\mkern4mu}\Relbar\Rightarrow}
\DeclareMathSymbol{\hookleftarrow}{\mathrel}{arrows}{60}
```

```
\DeclareMathSymbol{\hookrightarrow}{\mathrel}{arrows}{62}
\DeclareMathSymbol{\mapsto}{\mathrel}{arrows}{44}
\DeclareMathSymbol{\doteq}{\mathrel}{symbols}{201}
```

Ditto, because of latexsym and wasysym:

```
\let\mho\undefined
\DeclareMathSymbol{\mho}{\mathord}{letters}{146}
\let\lightning\undefined
\DeclareMathSymbol{\lightning}{\mathord}{arrows}{147}
} % AtBeginDocument
```

⟨/package⟩

References

[1] Frank Mittelbach and Michel Goosens: *The LaTeX Companion*. Addison Wesley, 2nd ed., 2004.