The euler package

Frank Jensen*

1995/03/05

1 Introduction

The euler package provides a setup for using the AMS Euler family of fonts for math in LaTeX documents. The AMS Euler family was designed by Hermann Zapf, commissioned by the American Mathematical Society. "The underlying philosophy of Zapf's Euler design was to capture the flavor of mathematics as it might be written by a mathematician with excellent handwriting." [2]

The euler package is based on Knuth's macros for the book "Concrete Mathematics" [1]. Knuth's macros can be found through anonymous ftp to `labrea.stanford.edu`: look for the file `gkpmac.tex` in directory `pub/tex/local/lib`. The Euler fonts can be found through anonymous ftp to `e-math.ams.org`: look in directory `pub/tex/amsfonts`.

The purpose of the euler package is to provide the math part of the look of [1]. The other part (text fonts) is provided by the beton package. The reason for creating two packages is to make it easy to use the Euler math fonts together with other text fonts (in particular, it appears that the Euler fonts match many of the popular PostScript fonts pretty well).

Basically, the euler package provides the same setup (the same definitions, math codes, etc.) as `gkpmac.tex` with respect to the Euler fonts. However, Knuth [2] admits that the macros were written for one specific project, namely to typeset the "Concrete Mathematics" book [1]. So, the euler package actually does a little more than `gkpmac.tex`: for example, some 'exotic' symbols, present in CM math italic, are missing from the Euler fonts; the euler package takes care of this (whereas `gkpmac.tex` does not).

The use of the euler package requires a lot of care when entering the manuscript, since the package will cause, e.g., math numerals to come from the Euler Roman fonts; these numerals are easily distinguished from the normal text numerals. This implies that sloppy typing is clearly exposed: imagine how "`$x$ is either 1 or $-1$`" will appear when typeset! In general, the typist will in each case have to decide whether an input fragment is a math or a non-math entity. This holds true even for the tiniest details, such as punctuation characters (parentheses, colons, semicolons, commas, and periods).

Finally, a few words about the technical quality of the fonts (there are no doubts about their artistic quality): The fonts are not tuned for low resolution rendering: I recommend at least 600dpi. The fonts employ none of the tuning machinery of CM: e.g., they do not use the 'blacker' parameter specified in each Metafont

*Updates by Frank Mittelbach.

mode definition, indicating that they might look somewhat light on printers that normally require a large 'blacker' value.

2 What the `euler` package provides

The `euler` package defines two new math alphabet identifiers, `\mathscr` (Euler Script, uppercase letters only) and `\mathfrak` (Euler Fraktur, both lowercase and uppercase letters).

By default, the original meaning of `\mathcal` is preserved, but if the `euler` package is loaded with the `mathcal` option, `\mathcal` will produce Euler Script letters instead of the usual Computer Modern calligraphic letters.

Also, by default, the `euler` package does not redefine the `\mathbf` math alphabet identifier; it will still produce bold CM letters and digits. However, loading the `euler` package with the `mathbf` option will cause `\mathbf` to produce bold Euler Roman letters and digits. Another way to get bold symbols is to use the `\boldsymbol` command provided by the `amsbsy` package (part of AMS-LaTeX).

Some users do not like the `\hat` accent glyph present in the Euler fonts (the `euler` package uses this accent glyph be default). To use the `\hat` accent glyph from the text font instead, load the `euler` package with the `text-hat-accent` option.

In [1], the space around relations in displays was increased (from '5mu plus 5mu' to '10mu minus 3mu'). The `euler` package does not change this space, since this is (for most cases) controlled by the LaTeX `eqnarray` environment. Also, in [1], displays are left justified with a suitable indentation; we leave such matters to the general layout.

3 Hello world

First, we announce the package.

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{euler}[\filedate\space\fileversion]
\typeout{Package: `euler' \fileversion\space <\filedate> (FJ and FMi)}
```

4 Font and other definitions

The AMS Euler family consists of: Euler Roman (medium and bold), Euler Fraktur (medium and bold), Euler Script (medium and bold), and Euler Extension (medium only). The `.fd` files for these fonts are defined in the `amsfonts.fdd` file (part of the AMSFonts distribution).

4.1 Euler Roman

The Euler Roman fonts replace the Computer Modern Math Italic fonts, located in the 'letters' math symbol font:

```
\DeclareSymbolFont{letters}{U}{eur}{m}{n}
\SetSymbolFont{letters}{bold}{U}{eur}{b}{n}
```

In math mode, digits should come from the Euler Roman fonts. As in standard LaTeX, we assign the type `\mathalpha` to the digits which means that they will vary with math alphabets. One of the reasons for this is that footnote numbers are set in math mode, and we want these numbers to come from the text font; this is accomplished by setting the TeX primitive `\fam` to zero. (See Section 7 below.)

```
\DeclareMathSymbol{0}\mathalpha{letters}{"30}
\DeclareMathSymbol{1}\mathalpha{letters}{"31}
\DeclareMathSymbol{2}\mathalpha{letters}{"32}
\DeclareMathSymbol{3}\mathalpha{letters}{"33}
\DeclareMathSymbol{4}\mathalpha{letters}{"34}
\DeclareMathSymbol{5}\mathalpha{letters}{"35}
\DeclareMathSymbol{6}\mathalpha{letters}{"36}
\DeclareMathSymbol{7}\mathalpha{letters}{"37}
\DeclareMathSymbol{8}\mathalpha{letters}{"38}
\DeclareMathSymbol{9}\mathalpha{letters}{"39}
```

The uppercase greek letters are also taken from the Euler Roman fonts. We make them ordinary symbols (i.e., of type `\mathord`) as opposed to `plain.tex`, `gkpmac.tex`, and standard LaTeX, which make them vary according to `\fam`.

```
\DeclareMathSymbol\Gamma  \mathord{letters}{"00}
\DeclareMathSymbol\Delta  \mathord{letters}{"01}
\DeclareMathSymbol\Theta  \mathord{letters}{"02}
\DeclareMathSymbol\Lambda \mathord{letters}{"03}
\DeclareMathSymbol\Xi     \mathord{letters}{"04}
\DeclareMathSymbol\Pi     \mathord{letters}{"05}
\DeclareMathSymbol\Sigma  \mathord{letters}{"06}
\DeclareMathSymbol\Upsilon\mathord{letters}{"07}
\DeclareMathSymbol\Phi    \mathord{letters}{"08}
\DeclareMathSymbol\Psi    \mathord{letters}{"09}
\DeclareMathSymbol\Omega  \mathord{letters}{"0A}
```

Euler doesn't have the special variants of `\sigma` and `\rho`:

```
\let\varsigma=\sigma
\let\varrho=\rho
```

`\mathbf` If the `euler` package is loaded with the `mathbf` option, `\mathbf` should produce bold Euler Roman letters and digits.

```
\DeclareOption{mathbf}
    {\AtBeginDocument{\DeclareMathAlphabet\mathbf{U}{eur}{b}{n}}}
```

Note the use of `\AtBeginDocument`; it ensures that the redefinition overrides the default settings (Section 6).

4.2 Euler Fraktur

`\mathfrak` The Euler Fraktur fonts get their own math symbol font. We define a math alphabet identifier—`\mathfrak`—to access this symbol font. For compatibility with previous versions of the `euler` package, we define `\frak` as an alias for `\mathfrak` (the `amsfonts` package also provides this alias).

```
\DeclareSymbolFont{EulerFraktur}{U}{euf}{m}{n}
\SetSymbolFont{EulerFraktur}{bold}{U}{euf}{b}{n}
\DeclareSymbolFontAlphabet\mathfrak{EulerFraktur}
\@ifpackageloaded{amsfonts}{}{\newcommand\frak{\mathfrak}}
```

The following characters come from the Euler Fraktur symbol font (as defined in gkpmac.tex):

```
\DeclareMathSymbol{!}\mathord  {EulerFraktur}{"21}
\DeclareMathSymbol{(}\mathopen {EulerFraktur}{"28}
\DeclareMathSymbol{)}\mathclose{EulerFraktur}{"29}
\DeclareMathSymbol{+}\mathbin  {EulerFraktur}{"2B}
\DeclareMathSymbol{-}\mathbin  {EulerFraktur}{"2D}
\DeclareMathSymbol{=}\mathrel  {EulerFraktur}{"3D}
\DeclareMathSymbol{[}\mathopen {EulerFraktur}{"5B}
\DeclareMathSymbol{]}\mathclose{EulerFraktur}{"5D}
```

There is a curious detail here: The type of '!' is `\mathclose` in `plain.tex` (and standard LaTeX) but `\mathord` in `gkpmac.tex`. The reason for this is probably that one should not use Euler punctuation characters as part of the text,[1] and punctuation characters in math are usually operators. E.g., in CSP (Communicating Sequential Processes), '!' denotes an output operation, a tightly binding binary operator; there is usually no space around this operator (to indicate the high precedence).

The Euler Fraktur symbol font contains more punctuation characters than defined above. We want to use these characters as they look better when they appear in math formulas (there are some strange looking formulas in [1]). Even if one may argue that, e.g., semicolon is most often used in the text sense (in displays), it is more consistent and easier to remember if everything in math mode is Euler.

```
\DeclareMathSymbol{"}\mathord  {EulerFraktur}{"7D}
\DeclareMathSymbol{&}\mathord  {EulerFraktur}{"26}
\DeclareMathSymbol{:}\mathrel  {EulerFraktur}{"3A}
\DeclareMathSymbol{;}\mathpunct{EulerFraktur}{"3B}
\DeclareMathSymbol{?}\mathord  {EulerFraktur}{"3F}
\DeclareMathSymbol{^}\mathord  {EulerFraktur}{"5E}
\DeclareMathSymbol{`}\mathord  {EulerFraktur}{"12}
```

Here, we also changed the type of '?' to `\mathord` to be consistent with the type of '!'. Regarding the math codes assigned by `INITEX`: The Euler fonts do not have the glyphs for `\#`, `\$`, `\%`, and `@`, so these glyphs are still taken from the text fonts. Note also that we do not set the math code for the other quote character: it is active in math mode (code `"8000`); it produces primes (and should continue to do so).

```
\DeclareMathDelimiter{(}{EulerFraktur}{"28}{largesymbols}{"00}
\DeclareMathDelimiter{)}{EulerFraktur}{"29}{largesymbols}{"01}
\DeclareMathDelimiter{[}{EulerFraktur}{"5B}{largesymbols}{"02}
\DeclareMathDelimiter{]}{EulerFraktur}{"5D}{largesymbols}{"03}
```

`\oldstylenums` `\TextOldstyle` `\MathOldstyle`

We provide a new definition of the `\oldstylenums` macro: this definition allows us to modify the generation of non-aligning (aka "oldstyle") numerals in text and in math separately, using two hooks, `\TextOldstyle` and `\MathOldstyle`.

```
\DeclareRobustCommand\oldstylenums[1]%
  {\begingroup
      \spaceskip\fontdimen\tw@\font
```

[1] In particular, one should watch out for punctuation characters in displays: Is, e.g., an exclamation point part of a formula or part of the text? (With Computer Modern, it didn't really matter, but it does matter with Euler!)

```
      \TextOldstyle \MathOldstyle #1%
   \endgroup}
```

In math, the `\oldstylenums` macro should produce Euler oldstyle digits (located in the Euler Fraktur fonts):

```
\def\MathOldstyle{\mathgroup\symEulerFraktur}
```

By default, oldstyle digits in text come from the Computer Modern fonts:

```
\providecommand\TextOldstyle{\usefont{OML}{cmm}\f@series{it}}
```

4.3 Euler Script

`\mathscr` Like the Euler Fraktur fonts, the Euler Script fonts also get their own symbol font. We define a math alphabet identifier—`\mathscr`—to access this symbol font. For compatibility with previous versions of the `euler` package, we define `\scr` as an alias for `\mathscr`.

```
\DeclareSymbolFont{EulerScript}{U}{eus}{m}{n}
\SetSymbolFont{EulerScript}{bold}{U}{eus}{b}{n}
\DeclareSymbolFontAlphabet\mathscr{EulerScript}
\newcommand\scr{\mathscr}
```

`\mathcal` If the `euler` package is loaded with the `mathcal` option, `\mathcal` should produce Euler Script letters.

```
\DeclareOption{mathcal}{\renewcommand\mathcal{\mathscr}}
```

The following is equivalent to the definitions in `gkpmac.tex` concerning the Euler Script fonts:

```
\DeclareMathSymbol\aleph\mathord{EulerScript}{"40}
%\DeclareMathSymbol\equiv\mathrel{EulerScript}{"11}
%\let\cong=\equiv % lowres bars weren't spaced right
%\DeclareMathSymbol\leq \mathrel{EulerScript}{"14}
%\let\le=\leq % seems to have been eliminated
%\DeclareMathSymbol\geq \mathrel{EulerScript}{"15}
%\let\ge=\geq % ditto
\DeclareMathSymbol\Re   \mathord{EulerScript}{"3C}
\DeclareMathSymbol\Im   \mathord{EulerScript}{"3D}
```

Knuth used a slightly different version of the Euler Script fonts that had special '$\leq$' and '$\geq$' glyphs. Actually, the source for the AMS Euler Script medium (but not the bold) font also contains these glyphs; however, they are commented out (no explanation is given). The definitions of `\equiv` and `\cong` are also commented out in `gkpmac.tex` (with the reason as given above, i.e., problems with low resolution rendering).

```
\DeclareMathDelimiter\vert
                  \mathord{EulerScript}{"6A}{largesymbols}{"0C}
\DeclareMathDelimiter\backslash
                  \mathord{EulerScript}{"6E}{largesymbols}{"0F}
```

The Euler Script fonts constitute a partial replacement for the Computer Modern 'symbols' math symbol fonts. (None of the remaining definitions in this section were included in `gkpmac.tex`.)

```
%\DeclareMathSymbol{-}\mathbin{EulerScript}{"00} % already done
\DeclareMathSymbol{|}\mathord{EulerScript}{"6A}
\DeclareMathDelimiter{|}{EulerScript}{"6A}{largesymbols}{"0C}
```

(The minus signs in the Euler Fraktur and Script fonts are identical, i.e., the sources are identical.)

```
\DeclareMathSymbol\neg      \mathord{EulerScript}{"3A}
\let\lnot=\neg
\DeclareMathSymbol\wedge    \mathbin{EulerScript}{"5E}
\let\land=\wedge
\DeclareMathSymbol\vee      \mathbin{EulerScript}{"5F}
\let\lor=\vee
\DeclareMathSymbol\setminus\mathbin{EulerScript}{"6E}
\DeclareMathSymbol\sim      \mathrel{EulerScript}{"18}
\DeclareMathSymbol\mid      \mathrel{EulerScript}{"6A}
\DeclareMathDelimiter\arrowvert
         \mathord{EulerScript}{"6A}{largesymbols}{"3C} % ???
```

(There are no `\approx` or `\simeq` to go with `\sim`.) In Plain TEX, the characters `{`, `\`, and `}` are assigned math codes (and `\` is also given a delimiter code); these are the only visible ASCII (7-bit) characters that we don't give a math code (we don't assign math codes to any of the invisible ASCII characters).

The Euler Script fonts also contain a section sign (§):

```
\DeclareMathSymbol\mathsection\mathord{EulerScript}{"78}
```

4.4 Euler Extension

We allocate a new symbol font group for this font:

```
\DeclareSymbolFont{EulerExtension}{U}{euex}{m}{n}
```

This font contains some alternative versions of some Computer Modern symbols. The `amsmath` package (part of AMS-LATEX) redefines the `\coprod`, `\prod`, and `\sum` symbols; to make the `euler` package work correctly with the `amsmath` package, we need to take different actions depending on whether the `amsmath` package has been loaded before the `euler` package or not. Note: This will *not* work with the `amstex` package.

```
\@ifpackageloaded{amsmath}
   {\DeclareMathSymbol\coprod@\mathop{EulerExtension}{"60}%
    \DeclareMathSymbol\prod@   \mathop{EulerExtension}{"51}%
    \DeclareMathSymbol\sum@    \mathop{EulerExtension}{"50}}
   {\DeclareMathSymbol\coprod  \mathop{EulerExtension}{"60}%
    \DeclareMathSymbol\prod    \mathop{EulerExtension}{"51}%
    \DeclareMathSymbol\sum     \mathop{EulerExtension}{"50}}
```

We shall respectfully ask the user not to use the `amstex` package.

```
\AtBeginDocument
   {\@ifpackageloaded{amstex}%
       {\PackageWarningNoLine{euler}{Please do not use `amstex'.
                 Use `amsmath' instead}{}}}
```

```
\DeclareMathSymbol\intop  \mathop {EulerExtension}{"52}
\DeclareMathSymbol\ointop \mathop {EulerExtension}{"48}
\DeclareMathSymbol\braceld\mathord{EulerExtension}{"7A}
\DeclareMathSymbol\bracerd\mathord{EulerExtension}{"7B}
\DeclareMathSymbol\bracelu\mathord{EulerExtension}{"7C}
\DeclareMathSymbol\braceru\mathord{EulerExtension}{"7D}
\DeclareMathSymbol\infty  \mathord{EulerExtension}{"31}
```

It is not enough to substitute the Euler integral sign for the Computer Modern integral sign, if we are using the amsmath package. The multiple integral signs will look bad. The following will make them look better.

```
\AtBeginDocument
  {\@ifpackageloaded{amsmath}%
      {\def\intkern@{\mkern -6mu }%
       \def\ints@#1{\mkern -3mu
          \mathop{\mkern 3mu
             \intop
             \ifnum #1=\z@ \intdots@ \else
                \intkern@
                \ifnum #1>\tw@ \intop\intkern@
                   \ifnum #1>\thr@@ \intop\intkern@ \fi
                \fi
             \fi
             \intop
          }\ilimits@
       }%
      }{}}
```

All Euler arrows come from the Euler Extension font:

```
\DeclareMathSymbol\nearrow\mathrel{EulerExtension}{"25}
\DeclareMathSymbol\searrow\mathrel{EulerExtension}{"26}
\DeclareMathSymbol\nwarrow\mathrel{EulerExtension}{"2D}
\DeclareMathSymbol\swarrow\mathrel{EulerExtension}{"2E}

\DeclareMathSymbol\Leftrightarrow\mathrel{EulerExtension}{"2C}
\DeclareMathSymbol\Leftarrow     \mathrel{EulerExtension}{"28}
\DeclareMathSymbol\Rightarrow    \mathrel{EulerExtension}{"29}
\DeclareMathSymbol\leftrightarrow\mathrel{EulerExtension}{"24}
\DeclareMathSymbol\leftarrow     \mathrel{EulerExtension}{"20}
\let\gets=\leftarrow
\DeclareMathSymbol\rightarrow    \mathrel{EulerExtension}{"21}
\let\to=\rightarrow

\DeclareMathDelimiter\uparrow
              \mathrel{EulerExtension}{"22}{largesymbols}{"78}
\DeclareMathDelimiter\downarrow
              \mathrel{EulerExtension}{"23}{largesymbols}{"79}
\DeclareMathDelimiter\updownarrow
              \mathrel{EulerExtension}{"6C}{largesymbols}{"3F}
\DeclareMathDelimiter\Uparrow
              \mathrel{EulerExtension}{"2A}{largesymbols}{"7E}
\DeclareMathDelimiter\Downarrow
              \mathrel{EulerExtension}{"2B}{largesymbols}{"7F}
\DeclareMathDelimiter\Updownarrow
              \mathrel{EulerExtension}{"6D}{largesymbols}{"77}

\DeclareMathSymbol\leftharpoonup   \mathrel{EulerExtension}{"18}
\DeclareMathSymbol\leftharpoondown \mathrel{EulerExtension}{"19}
\DeclareMathSymbol\rightharpoonup  \mathrel{EulerExtension}{"1A}
\DeclareMathSymbol\rightharpoondown\mathrel{EulerExtension}{"1B}
```

Note the funny mixture of the Euler Script and Extension fonts:

```
\DeclareMathDelimiter\lbrace
              \mathopen{EulerScript}{"66}{EulerExtension}{"08}
```

```
\DeclareMathDelimiter\rbrace
             \mathclose{EulerScript}{"67}{EulerExtension}{"09}
```

5 More math

Actually, we still need some Computer Modern fonts: We need the equals sign for long/extensible (horizontal) double arrows, and we need the minus sign for the single arrows! The Euler Extension font is based on the Metafont code for Computer Modern, and we need the Computer Modern equals and minus signs to make long/extensible arrows.

We allocate math alphabets to produce the equals and minus signs for arrows. Note that since Euler arrows are only available in medium weight (they come from the Euler Extension font which is only available in medium weight), we let all math versions of these math alphabets refer to medium weight fonts.

```
\DeclareMathAlphabet\cm@equals@alphabet{OT1}{cmr}{m}{n}
\DeclareMathAlphabet\cm@minus@alphabet{OMS}{cmsy}{m}{n}
```

Note that we use OT1 encoding for the `\cm@equals@alphabet`, even though T1 encoding may be the default for the users system.

Also, note that we already have the Computer Modern `cmsy` family assigned to a symbol font (namely the 'symbols' symbol font). However, taking the minus sign directly from this symbol font would make long/extensible arrows come out wrong in the 'bold' math version. (It is perhaps "overkill" to allocate a math alphabet just for this purpose?)

`\Relbar` `\relbar`

The long arrows are made using the macros `\Relbar` (for double arrows) and `\relbar` (for single arrows). The standard LaTeX definitions of these macros use the '`=`' and '`-`' characters to produce the extension part of the long arrows. Since we have changed the math codes of these characters (they are now taken from the Euler Fraktur fonts in math mode), we have to redefine the `\Relbar` and `\relbar` macros.

```
\def\Relbar{\mathrel{\cm@equals@alphabet{\mathchar"703D}}}
\def\cm@minus{{\cm@minus@alphabet{\mathchar"7000}}}
\def\relbar{\mathrel{\smash\cm@minus}}
          % \smash, because - has the same height as +
```

`\leftarrowfill` `\rightarrowfill`

Unfortunately, since the definitions of the "arrow-filling" macros (`\leftarrowfill` and `\rightarrowfill`) in standard LaTeX still use explicit minus signs (instead of `\relbar`), we also have to redefine those macros:

```
\def\leftarrowfill
  {$\m@th\mathord\leftarrow\mkern-6mu%
    \cleaders\hbox{$\mkern-2mu\cm@minus\mkern-2mu$}\hfill
    \mkern-6mu\cm@minus$}
\def\rightarrowfill
  {$\m@th\cm@minus\mkern-6mu%
    \cleaders\hbox{$\mkern-2mu\cm@minus\mkern-2mu$}\hfill
    \mkern-6mu\mathord\rightarrow$}
```

We have defined `\cm@minus` to produce a Computer Modern minus sign of type `\mathord`. We thereby save a few `\mathord` tokens (compared to the definitions of `\leftarrowfill` and `\rightarrowfill` in standard LaTeX).

There are a few symbols in the original Computer Modern Math Italic fonts that are not present in the Euler fonts. We simply take them from the CM fonts.

```
\DeclareSymbolFont{cmmigroup}{OML}{cmm}{m}{it}
\SetSymbolFont{cmmigroup}{bold}{OML}{cmm}{b}{it}
\DeclareMathAccent\vec           \mathord{cmmigroup}{"7E}
\DeclareMathSymbol\triangleleft  \mathbin{cmmigroup}{"2F}
\DeclareMathSymbol\triangleright\mathbin{cmmigroup}{"2E}
\DeclareMathSymbol\star          \mathbin{cmmigroup}{"3F}
\DeclareMathSymbol\lhook         \mathrel{cmmigroup}{"2C}
\DeclareMathSymbol\rhook         \mathrel{cmmigroup}{"2D}
\DeclareMathSymbol\flat          \mathord{cmmigroup}{"5B}
\DeclareMathSymbol\natural       \mathord{cmmigroup}{"5C}
\DeclareMathSymbol\sharp         \mathord{cmmigroup}{"5D}
\DeclareMathSymbol\smile         \mathrel{cmmigroup}{"5E}
\DeclareMathSymbol\frown         \mathrel{cmmigroup}{"5F}
```

Note that the arrow head of `\vec` is slanted; this looks a bit strange in combination with an upright font such as Euler Roman.

Also, note that in the 'bold' math version, arrows with hooks will come out wrong (the hooks will be bold, but the arrow heads will be of medium weight). This can be repaired in a similar way as the long/extensible arrows, but is it worth it to allocate a math alphabet just for this purpose?

Finally, Knuth wanted a little extra space before the prime superscript:

```
\begingroup
   \catcode`\'=\active
   \gdef'{^\bgroup\mskip2mu\prim@s} % more space before '
\endgroup
```

Open question: The `\dots` and `\ldots` macros produce dots (periods) from the Euler fonts; the `\ddots` and `\vdots` macros produce dots from the text font; and the `\cdots` macro produces dots from the Computer Modern 'symbols' math symbol font. Should we do something about this?

6 Text fonts in math

The text fonts used in math through math alphabets and the 'operators' symbol font should be the same as the fonts used for the main text. (It is now easy to replace the fonts for the 'operators' symbol font, since the uppercase greek letters are now taken from the Euler Roman fonts.)

In order to reduce the total number of fonts needed, we shall—if possible—use text fonts in the same encoding for both text and math. We therefore determine whether T1 encoding has been selected as the default encoding. We define a switch—`\ifCorkEncoding`—for that purpose.

```
\newif\ifCorkEncoding
\edef\@tempa{\encodingdefault}\def\@tempb{T1}
\ifx\@tempa\@tempb
   \CorkEncodingtrue
\else
   \CorkEncodingfalse
\fi
```

Note that we determine the encoding to use at the time the `euler` package is loaded (*not* at the end of the preamble). This allows the user to retain OT1 encoding for text fonts used in math (other packages may depend on that) by selecting T1 encoding *after* the `euler` package has been loaded.

We redefine the 'operators' symbol font and all math alphabets defined by the LaTeX kernel. Due to the current implementation of math accents that take their glyphs from text fonts, all text fonts must be used in a common encoding. The only LaTeX 'base' package that defines extra math alphabets is the `oldlfont` package. However, this package cannot be used together with the `euler` package (it allocates too many symbol fonts). In other words, the following setup should suffice for most users.

```
\ifCorkEncoding
  \AtBeginDocument
    {\DeclareSymbolFont{operators}{T1}\rmdefault\mddefault\updefault
     \SetSymbolFont{operators}{bold}{T1}\rmdefault\bfdefault\updefault
     \DeclareMathAlphabet\mathbf{T1}\rmdefault\bfdefault\updefault
     \DeclareMathAlphabet\mathsf{T1}\sfdefault\mddefault\updefault
     \DeclareMathAlphabet\mathit{T1}\rmdefault\mddefault\itdefault
     \DeclareMathAlphabet\mathtt{T1}\ttdefault\mddefault\updefault
     \SetMathAlphabet\mathsf{bold}{T1}\sfdefault\bfdefault\updefault
     \SetMathAlphabet\mathit{bold}{T1}\rmdefault\bfdefault\itdefault}
\else
  \AtBeginDocument
    {\DeclareSymbolFont{operators}{OT1}\rmdefault\mddefault\updefault
     \SetSymbolFont{operators}{bold}{OT1}\rmdefault\bfdefault\updefault
     \DeclareMathAlphabet\mathbf{OT1}\rmdefault\bfdefault\updefault
     \DeclareMathAlphabet\mathsf{OT1}\sfdefault\mddefault\updefault
     \DeclareMathAlphabet\mathit{OT1}\rmdefault\mddefault\itdefault
     \DeclareMathAlphabet\mathtt{OT1}\ttdefault\mddefault\updefault
     \SetMathAlphabet\mathsf{bold}{OT1}\sfdefault\bfdefault\updefault
     \SetMathAlphabet\mathit{bold}{OT1}\rmdefault\bfdefault\itdefault}
\fi
```

Note that we perform the setup at the end of the preamble (and not while the `euler` package is being input). This implies that any changes made to, e.g., `\rmdefault` by a package loaded after the `euler` package are seen.

The standard LaTeX math setup assumes that OT1 encoded text fonts are used in math. When using T1 encoded fonts, we have to redefine the macros that access accents from the text fonts.

```
\ifCorkEncoding
   \DeclareMathAccent\grave\mathalpha{operators}{"00}
   \DeclareMathAccent\acute\mathalpha{operators}{"01}
   \DeclareMathAccent\tilde\mathalpha{operators}{"03}
   \DeclareMathAccent\ddot \mathalpha{operators}{"04}
   \DeclareMathAccent\check\mathalpha{operators}{"07}
   \DeclareMathAccent\breve\mathalpha{operators}{"08}
   \DeclareMathAccent\bar  \mathalpha{operators}{"09}
   \DeclareMathAccent\dot  \mathalpha{operators}{"0A}
\fi
```

`\hat` There is one more accent that is normally taken from the 'operators' symbol font, namely the `\hat` accent. This is the only accent provided by the Euler fonts; we shall use this accent by default. (A curious fact is that Knuth devoted a fairly large portion of `gkpmac.tex` to the construction of a ϕ with a hat accent—using the `\widehat` macro).

```
\DeclareMathAccent\hat\mathalpha{EulerFraktur}{"5E}
```

However, some users do not like this `\hat` accent. They prefer the accent glyph to be taken from the text font (similar to the other accents).

```
\DeclareOption{text-hat-accent}
  {\ifCorkEncoding
       \DeclareMathAccent\hat\mathalpha{operators}{"02}%
   \else
       \DeclareMathAccent\hat\mathalpha{operators}{"5E}%
   \fi}
```

`\hbar` The `\hbar` macro in standard LaTeX is composed of the letter 'h' from the 'letters' symbol font and the macron ('bar') accent from the 'operators' symbol font. With the 'letters' symbol font now being Euler Roman, the bar is no longer positioned correctly. We choose the following definition (the `9mu` in the original definition has been changed to `8mu`):

```
\ifCorkEncoding
  \DeclareMathSymbol\euler@hbar\mathord{operators}{"09}
\else
  \DeclareMathSymbol\euler@hbar\mathord{operators}{"16}
\fi
\def\hbar{{\euler@hbar\mkern-8muh}}
```

7 Footnotes

`\@makefnmark` Footnote numbers are not math entities, so they should be produced by digits from the text font. Since footnote markers (whether they are numbers or symbols) are produced in math mode, we can accomplish this by selecting `\operator@font`; this will cause digits to be taken from the 'operators' symbol font (which is the same as the font used for text; see previous section).

```
\def\@makefnmark{\hbox{$\operator@font^{\@thefnmark}\m@th$}}
```

8 Processing options

Finally, we must remember to process the options:

```
\ProcessOptions
⟨/package⟩
```

References

[1] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

[2] D. E. Knuth. Typesetting Concrete Mathematics. *TUGboat* 10(1):31–36, 1989.