

Classic LaTeX Look and Feel with LuaLaTeX and OpenType Fonts in different Grades of Blackness

exactphilosophy.net  /patafont

Grade A (0) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade B (10) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade C (19, default) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade D (32) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade E (44) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade F (57) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

Grade G (72) optical sizes: Hamburg Hamburg Hamburg

The quick brown fox jumps over the lazy dog: **bold**, *italic*, *slanted*, ***bold italic***, SMALL CAPS.
 Accents: Věra Dvořák, Grüße, déjà, ¡Olé! ¿señor?, cœur, naïve, façade, Ångström.
 Symbols: £ \$ § ¶ © ® ™ ° † ‡ • · ± × ÷ ¬ μ ∥ ... x² y³ 1^a 2^o; “curly” ‘quotes’, en—dash, em—dash;
 ligatures office waffle affix final flag. Sans: regular, **bold**, *oblique*. Typewriter: mono *italic* 0011I.

About

In a nutshell, this package transports the classic **look and feel** of the original $\text{\LaTeX}$ into the newer mainstream $\text{\TeX}$ world of $\text{\LuaTeX}$ with OpenType fonts and a lot more.

Besides the look and feel via fonts and their metrics, this package is also able to partly reproduce the same **layout**. The main limitation is hyphenation, where $\text{\LuaTeX}$ can only be partially convinced to do it as wrong as the original does in some cases.

Classic $\text{\LaTeX}$ and also largely $\text{\PDFLaTeX}$ / $\text{\OT1}$ yield essentially still the unified and styled look of what Donald E. Knuth originally implemented between about 1977 and 1990 with a trinity of $\text{\TeX}$ engine, **Computer Modern** fonts and **METAFONT**, plus Leslie Lamport's $\text{\LaTeX}$ format (1984) that made all of this widely accessible and together hugely successful until today, with immense community support.

Most of this has been transported into the new world: The Latin Modern fonts are very similar to the base Computer Modern fonts both in terms of glyphs and metrics, plus support a lot more glyphs; the $\text{\LuaTeX}$ engine (and others) can typeset with very similar qualities, plus a lot more, but one feature got lost: METAFONT.

With METAFONT you can make a font gradually **blacker**, change what is called the **grade** in typography. Originally METAFONT was used mainly for printing to paper, where depending on the printer device, you had to make the font blacker. Since you could only make a font print *more* black but not less black that way and low resolution screens would also make fonts generally look blacker, Latin Modern often looks quite thin today. (I think the print to paper part is not 100% correct, at least for low resolution printers, but better ask someone who was involved then for the gory details.)

As far as I could gather, OpenType fonts lack the singular quality to make a font blacker while still preserving its nature and not making a regular weight font look like a bold weight font. For this package I have used the original Computer Modern fonts in METAFONT, made them gradually blacker to bitmaps, traced them to outlines and transformed them to standard OpenType fonts I named **Computer Modern Graded**.

For you, as a user, this means basically three things:

- You can give your documents a bit more blackness to make them shine.

But be aware that what a user will see depends on the PDF viewer (and also still in print on where it is printed). This text, for example, is set with “Grade C” from this package. In many mainstream PDF viewers it should look good, but in particular on macOS it will typically be quite heavy, exactly because the visually sensitive people at Apple do correct for often encountered fonts that are rather thin. See also [Tips and Tricks](#) for how to change that on Mac.

- You can create or port documents in the classic style.

Note that the set of characters is somewhat limited, a bit less than half of Latin Modern, but still well over 300 characters with many accented combinations. If you create a tagged PDF, the text will be fully tagged/accessible, only so far not the math, unless you use a separate math font for that. And, yes, with usually just maybe a little hyphenation tweaking for foreign languages and accents, you should be able to port your thesis/paper/book to $\text{\LuaTeX}$.

- You could also use the fonts in this package outside of $\text{\LaTeX}$, anywhere where OpenType fonts can be used, including on websites.

To get the fonts in various packagings, visit my website at exactphilosophy.net/patafont

Next, basic usage of the package is described, which is really easy, followed by a bit more background, some advanced use cases, and a few rather administrative things like license and version history.

Basic Usage

This is the simplest way to use this package in $\text{\LaTeX}$:

```
\usepackage{cmgraded}
```

This selects the default grade of blackness C for all text font families and for math, the same grade in which this package documentation is written. You can select seven visual grades from A to G. For example, to select grade A, which is nominally no blackening, you could use:

```
\usepackage[grade=A]{cmgraded}
```

or lowercase:

```
\usepackage[grade=a]{cmgraded}
```

The grades A-G are chosen to be in equal steps in visual perception. There are numerical values behind them which will be explained later.

Visual	A	B	C	D	E	F	G
Blacker	0	10	19	32	44	57	72

Accordingly, you could also configure grade A numerically as 0:

```
\usepackage[grade=0]{cmgraded}
```

Usage per Font Family

If you do not want to use Computer Modern Graded for all font families, which are **serif** (“roman”), **sans**, **typewriter** (“mono”) and **math**, you can deselect them individually. All except serif and math:

```
\usepackage[serif=none,math=none]{cmgraded}
```

Only math:

```
\usepackage[text=none]{cmgraded}
```

or explicitly:

```
\usepackage[serif=none,sans=none,typewriter=none]{cmgraded}
```

Spacing between oblique and upright Fonts

The default behavior of space between oblique (slanted/italic) glyphs and upright glyphs, as in *leafLet*, reproduces what classic $\text{\LaTeX}$ and $\text{\PDFLaTeX}$ do, namely they provide a spacing that is part of the oblique font files. OpenType fonts do not have that feature, they instead have a feature where the extent of ink of the oblique glyph to the right is measured and a fixed amount is added.

By default, this package adds a callback in Lua that reproduces the $\text{\LaTeX}$ behavior exactly. Besides faithful reproduction, the spacing will be exactly the same independently of the chosen grade of blackness, which is typically preferred in typography because then you can tune blackness without even only occasionally breaking the layout of paragraph and pages. This can be turned off to get the standard OpenType mechanism, which is also what you would get if you used the fonts without using the package:

```
\usepackage[italiccorrection=modern]{cmgraded}
```

The default value is **classic**.

Spacing at the End of Sentences in Typewriter

Classic $\text{\LaTeX}$ behavior after the end of a sentence ($.?!:$) with monospaced fonts is to add two spaces. People used to insert two spaces on mechanical typewriters and also initially on computers, but this has since virtually disappeared, which is also why normally $\text{\LaTeX}$ does not do that. In this package for the typewriter font, the default is **classic**, which inserts two spaces. For modern behavior instead, set:

```
\usepackage[ttsentencespacing=modern]{cmgraded}
```

Spacing at the End of quoted Sentences

If the end of a sentence is followed by a closing double quote (e.g. ‘**e!**” **H**’), in classic $\text{\LaTeX}$ behavior the space to the next character is different than without the closing double quote, usually a bit smaller to keep the letters visually closer. To get this classic spacing in $\text{\LaTeX}$ with this package, use:

```
\usepackage[quotedsentencespacing=classic]{cmgraded}
```

This setting is **modern** by default because **classic** sets `\sfcode"201D=1000`, which is a global setting that affects all fonts in the document. But you could use the low-level setting with value 1000 or 0 at your desired places directly in the document.

Hyphenation

By default, nothing special is set for hyphenation, just use the packages **babel** or **polyglossia** for language-specific hyphenation as usual. If you want hyphenation that is closer to classic $\text{\LaTeX}$, which in particular does not allow to break parts of words after any accented character, you can set this:

```
\usepackage[hyphenation=accentsclassic]{cmgraded}
```

The default value is **modern**.

Note that if you have not been using **babel** with classic $\text{\LaTeX}$, that then the behavior with **accentsclassic** would probably be at least almost always equal, more precisely often equally wrong outside of purely English words. Here is how the German word “Metalloberfläche” is hyphenated depending on context, only **blue*** is correct:

	no language pkg	babel ngerman / polyglossia german
pdf/ot1	Met-al-loberfläche	Me-tal-lo-ber-fläche / -
lua modern	Met-al-lober-fläche	Me-tall-ober-flä-che*
lua accentsclassic	Met-al-loberfläche	Me-tall-ober-fläche

For the **look and feel** of classic $\text{\LaTeX}$, I estimate that correct hyphenation usually does no harm. Closing the gap completely would be unrealistic in a package anyways.

Math Font

Note that so far math is always using Type 1 (Adobe / PostScript) fonts, not OpenType fonts, which means that support for tagged/accessible math in PDFs is quite limited.

Actually the same is the case in $\text{\LaTeX}$ by default, where AMS Computer Modern Type 1 fonts are used, the same as in $\text{\PDF\LaTeX}$ by default; only if you use the package **unicode-math** does the default change to the Latin Modern Math font.

Faithful rendering with OpenType math fonts would only seem technically feasible in Lua $\text{\LaTeX}$ with quite some drawbacks (complexity, hacks, visual defects when used in MathML or Typst), see the later section [OpenType Math Fonts?](#) for more details.

Advanced and Fringe Use Cases

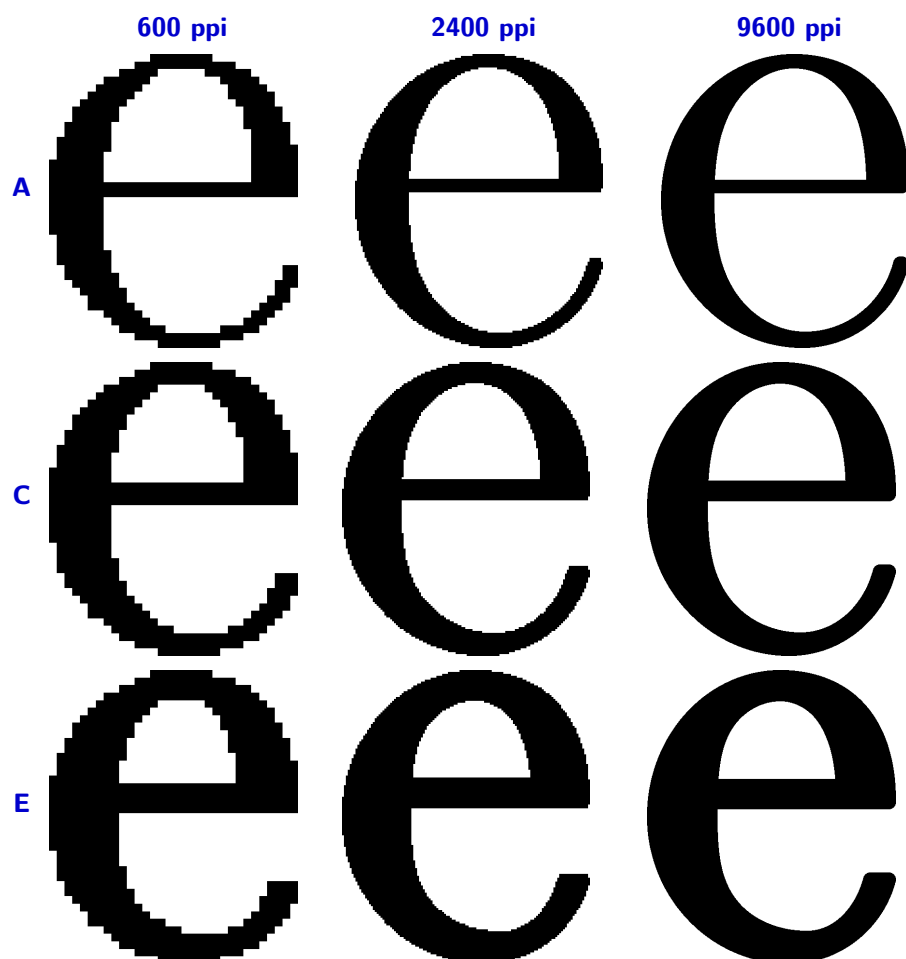
See the section [Advanced Usage](#) for how to use the fonts directly via the `fontspec` package and for unsupported but still relatively broadly working features with Type 1 fonts for text and/or for usage with other engines than Lua $\text{\TeX}$.

Background

METAFONT

The original $\text{\TeX}$ was based on a technology called METAFONT, designed like $\text{\TeX}$ and the Computer Modern fonts by Donald E. Knuth, starting from about 1977, see e.g. the books he wrote about METAFONT and font design.

What matters maybe most in the context of this package is that with METAFONT you can make a font *naturally blacker*, “make it look heavier but still the same font” — not even make a regular weight font look like the bold font of the same family — and keep all metrics in typesetting exactly intact.



Above is the letter “e” from Computer Modern Roman 10 pt rasterized at different resolutions and made blacker with grade A (0), C (19) and E (44). METAFONT operates

on mathematical descriptions of the glyphs and generates bitmaps.

What is called “grade” in this package is called `blacker` in METAFONT and is one of several knobs that can be turned to adjust the output, although probably the most important one. Originally the main purpose had been to achieve ideal output with given printers, see the file `modes.mf` in the TeX Live distribution. Example:

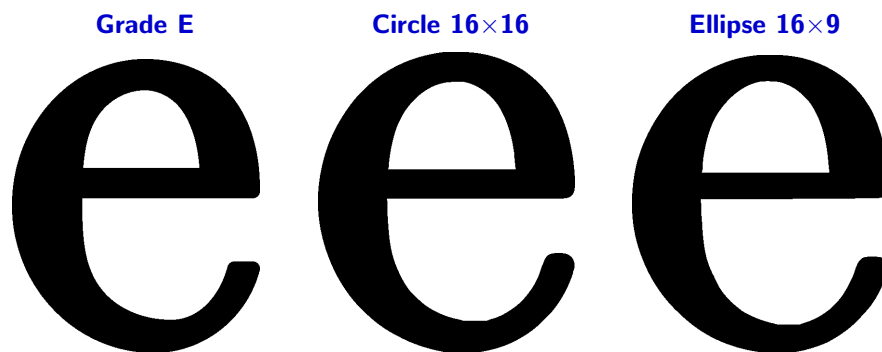
```
mode_def lexmarkr =  
  mode_param (pixels_per_inch, 1200);  
  mode_param (blacker, 2);  
  mode_param (fillin, 0);  
  mode_param (o_correction, 1);  
  mode_common_setup_  
enddef;
```

Here `fillin` and `o_correction` do nothing, only `blacker` and `pixels_per_inch` (ppi) do. The numerical grade in this package is actually the `blacker` value for 12000 ppi; in other words, the `lexmarkr` mode has grade 20. Technically, for this package fonts at different grades are rendered by rasterizing and blackening them to 9600 ppi bitmaps (at 12000 ppi some things break) and up to grade 72 (above some things break), and then tracing the bitmap images to outlines for OpenType fonts (3rd order Bézier curves).

Visual grades A-G mirror constant factors of stem widening between each grade.

Something like that would, of course, also be desirable for OpenType fonts or any other new font technologies in the future—wink wink nudge nudge. . .

Below an example with an earlier attempt of mine to make OpenType fonts heavier, namely rasterize at high resolution and then draw their outlines with a circular (disk) or ellipse pen. As you might recognize in the two images on the right, the nature of the font changes, which may in some cases be OK, especially with simpler fonts like one of the many Helvetica clones used all over the web, or also some typewriter fonts, where there is less variation of the thickness of the lines.



METAFONT OT1 to OpenType Fonts

The original Computer Modern in METAFONT consists of several font files of which each only has 128 glyphs (7 bits). When typesetting, those glyphs are retrieved in various ways, including composition like putting a ‘`ˆ`’ on top of an ‘`a`’ to typeset an ‘`ä`’ or even put a ‘`c`’ into a circle to get a ‘`©`’ symbol.

To form the Computer Modern Graded contemporary OpenType font files, that part of typesetting is baked into the fonts: Every glyph is first turned into an outline by the METAFONT to bitmap to trace process described above; what remains is to decide *which* glyph comes from *where*, and how to build the ones that never existed as a single glyph at all. Note that nothing is redrawn or otherwise redesigned—each symbol is the genuine Computer Modern glyph, blackened to the chosen grade.

- **Borrowing glyphs.** Most letters, digits and punctuation come straight from the text font. Many symbols, though, live in *other* Computer Modern fonts and are simply fetched from there: the section and paragraph marks (§ ¶), the daggers († ‡) and the large circle all come from the math *symbol* font, with a few more taken from the italic and slanted companions.
- **The \$ and £ switch.** A famous quirk of OT1 font encoding is that the same slot shows a **dollar** \$ in an upright font but a **pound** £ in an italic one. Generated OpenType fonts carry mutual copies of each. An upright, slanted, sans or typewriter face keeps its \$ and fetches the £ from an italic relative; an italic face keeps the £ and fetches the \$ from its upright or slanted one.
- **Composing accents.** Accented letters such as é â ñ ü are not stored ready-made; they are *composed*, exactly as the original T_EX did with its \accent primitive. The bare accent is centred over the base letter and raised to sit the right distance above the x-height (accents that hang *below*, like the cedilla of ç, are placed underneath instead). Because the package reproduces OT1 faithfully, a composed letter is given the **advance width of its base letter** and—just as under \accent—does not take part in kerning with its neighbours. The accent's ink may overhang the letter box a little (like the circumflex on î); only the advance is held to the base.
- **Composing symbols.** The same approach builds glyphs that T_EX itself only ever *assembled*: the **copyright** sign © is a letter c set inside a circle, ® a letter R in a circle, TM a raised TM, and the degree sign, the ordinal indicators (^a ^o) and the superscript digits (² ³) are small glyphs lifted into position.
- **Metrics and kerning.** None of the spacing is invented. The advance widths, side bearings and the whole **kerning** table are read from the original T_EX font metrics (the .tfm files) and carried over unchanged, so a paragraph set in these fonts breaks almost always where it always did for texts in English, less often so e.g. in German, due to typically different hyphenation in LuaT_EX, as already explained earlier on. The classic f-ligatures (fi fl ff ffi ffl) are kept as real glyphs.

In all transformations above, and this is the whole point of the exercise, the glyphs may get blacker, but the **metrics stay identical**, grade after grade.

Compared to Latin Modern OpenType text fonts, Computer Modern Graded contains a bit less than half of the glyphs, but still well over 300.

Below for reference how the text fonts are mapped. Grade goes from 0 to 72, while only 7 grades are included in this package; the others can be obtained via my website.

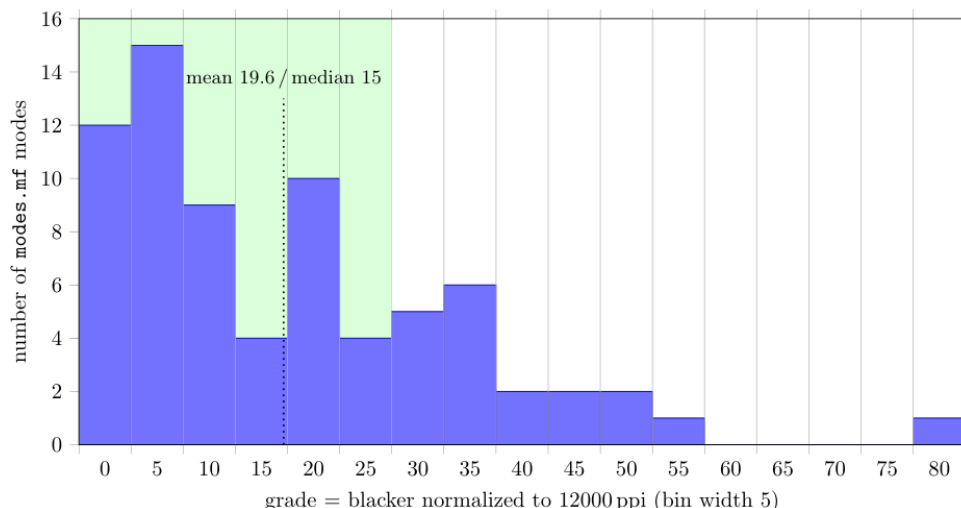
Metafont	OpenType
cmr{size}.mf	ComputerModernGradedG{grade}Serif{size}Regular.otf
cmti{size}.mf	ComputerModernGradedG{grade}Serif{size}Italic.otf
cmbx{size}.mf	ComputerModernGradedG{grade}Serif{size}Bold.otf
cmbxti{size}.mf	ComputerModernGradedG{grade}Serif{size}BoldItalic.otf
cmsl{size}.mf	ComputerModernGradedG{grade}Serif{size}Slanted.otf
cmbxsl{size}.mf	ComputerModernGradedG{grade}Serif{size}BoldSlanted.otf
cmcsc{size}.mf	ComputerModernGradedG{grade}Serif{size}SmallCaps.otf
cmss{size}.mf	ComputerModernGradedG{grade}Sans{size}Regular.otf
cmssi{size}.mf	ComputerModernGradedG{grade}Sans{size}Slanted.otf
cmssbx{size}.mf	ComputerModernGradedG{grade}Sans{size}Bold.otf
cmtt{size}.mf	ComputerModernGradedG{grade}Typewriter{size}Regular.otf
cmitt{size}.mf	ComputerModernGradedG{grade}Typewriter{size}Italic.otf
cmstt{size}.mf	ComputerModernGradedG{grade}Typewriter{size}Slanted.otf
cmtcsc{size}.mf	ComputerModernGradedG{grade}Typewriter{size}SmallCaps.otf

Fun Facts

Some fun facts before advanced use cases, to be taken with a grain of salt ;)

- The average blacker value used in the `modes.mf` file of TeX Live, not counting ones with blacker 0 and counting the ones where several reference the same settings only once, would be grade 19.6, i.e. very close to the default I chose from decades of using these fonts, especially Computer Modern Sans since 2005 on my website.

`modes.mf` grade distribution — 73 modes with non-zero blacker (58 grade-0 modes excluded)



- The thickness of a glyph stem increases between grade A and G with a factor close to the golden ratio $(\sqrt{5} + 1)/2 = 1.618\dots$
- Macs render fonts in PDFs typically heaviest, but did you know that you can change that with a setting called [AppleFontSmoothing](#) that you can set per application, computer or user? Ask your favorite AI how.

But note that it then affects everything that is drawn by Quartz in that context. If you set it for TeXShop, it will not only affect the rendered PDF, but also the $\text{\LaTeX}$ sources in the editor; if you set it for Safari, it will affect all web content. Desirable for the future would, of course, be a standard for how fonts are rendered across devices and that at least all major manufacturers would implement it; at least that would be a wish of mine.

Advanced Usage

Use OpenType Fonts directly with fontspec

This can be done just like with any OpenType font, e.g. serif at default grade:

```
\setmainfont{Computer Modern Graded Serif}
```

or at grade 44 (E):

```
\setmainfont{Computer Modern Graded G44 Serif}
```

Note that then the Lua callback for `italiccorrection` is not set, but if you want that behavior, you could just use the package without using fonts like this:

```
\usepackage[text=none,math=none]{cmgraded}
```

For more complex use cases, load the font files by their names, etc.

And note that the set of Unicode characters in these fonts is limited by what the original T_EX would be able to assemble from different font files of which each was limited to 128 glyphs (7 bit). Together with borrowed and assembled glyphs, this yields still well over 300 glyphs per font. See [cmgraded-list-glyphs.tex](#) in the examples of this package for listing exactly which glyphs are available.

Type 1 for Text / Using other Engines

As mentioned earlier, currently for math Type 1 fonts are used. Since Type 1 math already requires almost all text fonts, it was easy to also add the complete families of serif, sans and typewriter as Type 1 fonts. And those can in general be used with PDF_ΛT_EX, as well as with Lua_ΛT_EX and Xe_ΛT_EX.

Since that seems not to be where things are going in the T_EX world and some things might be difficult or impossible to get right without huge efforts, these are **not supported**, but you can use them. You could even create a package of your own that depends on this one for the font files and uses them differently. One day I might publish the software I used to create these fonts; then you could likely also apply it to different METAFONT fonts, and so on. Meaningful additions to this package that do not open a can of worms are not strictly excluded, but see also the next section.

Here is the full matrix of the family package keys, all default to auto:

key	values (default blue*)
text	auto* / opentype / type1 / none
- serif	auto* / opentype / type1 / none
- sans	auto* / opentype / type1 / none
- typewriter	auto* / opentype / type1 / none
math	auto* / type1 / none

The settings for serif, sans and typewriter overrule a text setting, inspired by inheritance in CSS for webpages. Here are the defaults and what is supported:

format	key	values (unsupported red*)	default/auto
Lua _Λ T _E X	text	opentype / type1* / none	opentype
	math	type1 / none	type1
Xe _Λ T _E X	text	opentype* / type1* / none	opentype
	math	type1* / none	type1
PDF _Λ T _E X	text	type1* / none	type1
	math	type1* / none	type1

To use any unsupported combination, you must set this:

```
\usepackage[allowunsupported=true]{cmgraded}
```

Since this package loads Type 1 fonts only at runtime, in order to use it with the old but lightning fast latex/dvips/ps2pdf toolchain, you have to manually add desired grades by enabling their map files. For example for the default grade:

```
updmap-sys --enable Map=cmgg19.map (typically with sudo)
```

These unsupported features are likely still useful *as-is* for some use cases; for an example see [Tips and Tricks](#) a bit later. The reason they are not supported is mainly that closing the gaps completely would usually be too complex to make sense in practically all cases.

Scope / Additional Glyphs

In a nutshell, the scope of this package is roughly the state of what [Knuth froze around 1990](#) and essentially including only features that were part of that in the sense of having been available without including extra style files written by others, except what Lamport implemented and published as the original $\text{\LaTeX}$ — in a nutshell.

The set of METAFONT fonts is limited. For text fonts to the three main families and of them only what could be made to fit into an OpenType font. For example, SmallCaps is part of it, but not bold fonts with regular spacing or upright italic fonts. For math fonts, it is the symbol fonts plus again the above three main families as Type 1 fonts.

Anything that emerged later, like T1 font encoding, is generally a non-goal for this package. Also early “third-party” packages like the babel-predecessor `german.sty`, which created a German low quote by lowering a high quote from the Computer Modern fonts, are generally not in scope.

But you can [add](#) such “emulated” [glyphs yourself](#), even such that you can then use the Unicode characters directly in the source and have them essentially correctly in the resulting PDFs: See [cmgraded-emulate-glyphs.tex](#) in the example files of this package for how to add German low quotes, lower umlauts and French guillemets that way.

Tips and Tricks

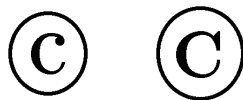
- Use [grade G](#) (72) to showcase all that is blackened and what is not.

To simulate bleeding and similar vintage effects with Computer Modern printed on paper, maybe even consider post-processing glyphs in ways similar to what the late [Fredrick Brennan](#) did with his hyperrealistic typewriter font [TT2020](#), where he even used 9 slightly different glyphs per character...

errors

- Use PDF $\text{\LaTeX}$ both with a [mode equivalent to one of the grades](#) to generate a PDF with Type 3 bitmap fonts and with the unsupported equivalent Type 1 text and math fonts, and then compare the results in detail.

Note that since TeX Live 2020, there have unfortunately been some subtle regressions in PDF $\text{\LaTeX}$. The $\text{\textcircled{c}}$ symbol originally — and in the OpenType fonts of this package — is a lowercase ‘c’ placed in a circle, but since 2020 an already composed glyph from jknappen METAFONT glyphs is used instead, with an uppercase ‘C’ inside a slightly larger circle as a single glyph...



- Finally, in case that you had missed it before:

If you have some [vintage](#) documents typeset with classic $\text{\LaTeX}$ or PDF $\text{\LaTeX}$ with OT1 encoding, now you could re-typeset them to tagged PDFs to make them more [accessible](#), albeit except the Type 1 math inside, while you might want to use something else as a math font via the [unicode-math](#) package...

And, obviously, when printing a document on a specific printer, print test pages with different grades and choose the one that looks best. For example, at home on my Epson inkjet printer on Mac, I print this document with Grade A.

Administrative

Thanks

I wrote the first version of this package in only three weeks in July 2026, closely helped by Claude Code (Opus 4.8 model).

However, this would not have been possible without quite some previous feedback from the folks at [tex.stackexchange](https://tex.stackexchange.com) mainly in autumn 2025, and learning from many, many failed attempts on my side, including growing OpenType fonts with a “pen”, which was useful with some simpler fonts that I use, but not with Computer Modern.

Also very helpful was the feedback from CTAN at the first package submission, including Karl Berry’s proposal to change the package name to `cmgraded` from `cmlegacy` (and before submission even `computermodernlegacy`), which I embraced as ideal, while “legacy” in its older, more positive sense as something from the past worth cherishing and preserving, still carries also a lot of the meaning of this package.

Thanks a lot everybody!

PATAFONT

The tool I created is called `PATAFONT`, as a pun both on Donald E. Knuth’s `METAFONT` and on Alfred Jarry’s Pataphysics.

So far the tool is not in a state to be publicly released, but one day it might.

For all the stuff I publish around $\text{T}_{\text{E}}\text{X}$ and fonts, always start here:

<https://www.exactphilosophy.net/patafont>

Besides this package also for local installation, you should also be able to download the remaining 66 of the grades 0-72 there. At 300 MB that would have been too much for a $\text{\LaTeX}$ package included into TeX Live. And you should also be able to find the fonts in formats immediately more useful for usage outside of $\text{\LaTeX}$, like compressed *.woff2 OpenType font files for websites, and maybe more in the future.

Issues

To report issues and to view open and resolved issues, also use the above link. Where exactly the link will lead you, might change in time, hence please usually start there.

What is a bug in context of this package? If the glyphs or metrics of the fonts are wrong in the sense of differing from the classic original. Different hyphenation is not a bug, as covered earlier. Also generally not a bug is “missing glyphs”, even that there is, for example, no straight quote “” in the serif and sans fonts. Scope is $\text{\LaTeX}$.

Validations

The fonts were validated with OpenType Sanitizer (OTS) and FontBakery. No real issues were detected, just two deliberate choices are worth mentioning:

- `mandatory_glyphs`: The `.notdef` glyph is blank; this is the same behavior as in Latin Modern. In practice this means that if you use a character that is not in the fonts, it will be typeset as a blank space instead of something like □ ◻ ◻.
- `name/family_and_style_max_length`: The font names are **>31** characters, which would only be an issue in font selection dialogs if you installed the OpenType fonts manually in the OS and used very old applications (esp. Windows GDI).

This package documentation is built as a tagged/accessible PDF. As of summer 2026 it takes three (unfortunately quite slow) passes of `lua $\text{\LaTeX}$` to build with all tags set. But in the end it verifies as PDF/A-4f and PDF/UA-2, as validated by veraPDF.

My Incentive

Simply to build my website in the same quality with `Lua $\text{\LaTeX}$` to tagged/accessible PDFs and digitized images as with `PDF $\text{\LaTeX}$` /OT1 with a mode with 1200 ppi and blacker 1.9, which would nominally be grade 19, but the different resolution and pottracing also make a difference in practice. As of 4 August 2026, I have migrated to grade 23, while for printing via IngramSpark a different grade might be better.

The fonts in the package are anyways beautiful in my eyes; sharing them with classic metrics after having used $\text{\LaTeX}$ for so long only seemed natural.

OpenType Math Fonts?

I explored the possibility during four days in August 2026 with a prototype.

A result faithful to the original seems only possible in `Lua $\text{\LaTeX}$` if you fix several unavoidable visual defects with custom handling in `*.sty` and `*.lua`. This adds quite some complexity and makes the fonts not usable without obvious defects in the other key places where OpenType math is used today: MathML (web browsers) and Typst.

In contrast, the OpenType text fonts in this package can be generally used outside of `Lua $\text{\LaTeX}$` with only rare spacing details not being identical to the original.

OpenType math fonts would have added about 15 MB to the package: 4 fonts in different sizes per grade, each containing $3\times$ all glyphs, in normal size plus script and scriptscript sizes. Generation and caching on-the-fly as METAFONT does in classic $\text{\LaTeX}$ and in `PDF $\text{\LaTeX}$` would be more economical, but would also add complexity.

Package and Licensing Info

This LaTeX package with name `"cmgraded"` contains OpenType and Type 1 (Adobe/PostScript) fonts named "Computer Modern Graded" that have been generated from the original "Computer Modern" fonts by Donald E. Knuth and TeX metrics by applying different grades of "blackness" via METAFONT that make the fonts gradually darker, while keeping metrics intact and not turning a regular weight font visually into a bold one.

Copyright (c) 2026 Alain Stalder (alain@exactphilosophy.net).
<https://www.exactphilosophy.net/patafont>

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c of this license or (at your option) any later version. The latest version of this license is in <https://www.latex-project.org/lppl.txt> and version 1.3c or later is part of all distributions of LaTeX version 2008 or later.

This work has the LPPL maintenance status 'maintained'.
Current Maintainer of this work is Alain Stalder (alain@exactphilosophy.net).

The fonts in this package are licensed separately under the SIL Open Font License v1.1, with Reserved Font Name "Computer Modern Graded", see the OFL.txt included with this package and note that the OpenType fonts also contain the full license inside in their name tables.

Version History

I am using [semantic versioning](#), with maybe very rare exceptions, when it makes sense.

2026-08-09 **1.0.2** (fonts 1.001)

- Tiny fix in the OpenType fonts: Added FontBBox to the CFF table.
- Documented exploration results for OpenType math fonts.
- Added an example with lowered umlauts in `cmgraded-emulate-glyphs.tex`.

2026-08-02 **1.0.1** (fonts 1.000)

- Tiny fix to make this package also basically workable with the original `latex/dvips/ps2pdf` toolchain, see the documentation for how to use it in detail.
- Tiny refinements in the documentation.

2026-07-29 **1.0.0** (fonts 1.000)

- First really public release after several weeks of valuable feedback by the community and some refinements and fixes, including to the package name.