

The New Computer Modern FontFamily version 7.1.1

Antonis Tsolomitis

January 7, 2026

Contents

1	Introduction	4
2	How to load the fonts	5
3	The Latin alphabet	5
3.1	Ligatures and stylistic alternatives in Latin	5
3.2	Oldstyle numbers	6
3.3	Old Italic	7
3.4	Diacritics Stacking	7
3.4.1	Coloring diacritics	7
4	Greek	8
4.1	Other character variants	9
4.2	Prosodic symbols	10
4.3	Archaic Greek writing	11
4.4	Aegean Numbers	11
4.5	Support for Papyrology	12
4.6	Support for Chemistry	12
5	Cyrillic	12
6	Hebrew	13
7	Coptic and Epact Numbers	13
8	Cherokee	13
9	Canadian Aboriginal Syllabics	14
10	Devanagari	14
11	Medieval Latin and Uncial Greek	16

12 Braille	16
13 Ipa symbols	17
14 All Unicode combining marks	18
14.1 Extended and Supplement Combining Diacritical Marks	19
14.2 Diacritical Marks for Symbols and Half Marks	19
15 Editorial Marks	20
15.1 Scope and Intended Use	21
15.2 Combining Enclosing Marks	21
16 Phonetic Extensions and Supplement	22
17 Currency Symbols	23
18 Bold Sans	23
19 Food Allergies	23
20 Unicode Math coverage and options	24
20.1 Bold Math	25
20.2 Sans Serif Math	26
20.3 Optical sizes for more glyphs	27
20.4 Math Script	27
20.5 Math Kerning	28
20.6 Blackboard Bold	29
20.6.1 Small Caps for Blackboard Bold	30
20.7 Upright and extensible integrals	30
20.8 Additional and alternative characters in Math	31
20.8.1 Alternative symbol for the empty set	31
20.8.2 Additional symbols	31
20.8.3 Additional operators	32
20.8.4 “Smoother” changing radicals	32
20.9 Notes on Mathematics	32
21 The Medieval Latin and Uncial Greek glyph complement	33
22 The Aegean Numbers glyph complement	34

1 Introduction

The NewComputerModern FontFamily is a huge extension (“huge” in terms of the number of additional glyphs) of the `lm` fonts. It is not just a family adding random missing glyphs but it adds support for several more languages and shapes needed for academic (and not only) work. Currently it supports among

others, Greek¹, Cyrillic², Devanagari, Hebrew, Coptic, Cherokee and Canadian Aboriginal. Since it supports diacritics stacking the number of languages that use the Latin alphabet is greatly expanded. Diacritics stacking is also needed for Greek for papyrological work and this is also supported.

Version 4.0 adds to the classic design of computer modern new shapes for Latin and Greek, in particular it adds families for Medieval Latin and Uncial Greek matching in style to the main family.

In terms of weights and sizes, all of its shapes come in Regular, Book weights at 10 and 8 point sizes and in Bold at 10 points.

Mathematics is also supported in Regular and Book weights, currently providing a full coverage of the Unicode Math blocks (with a few more glyphs needed for Mathematics that Unicode has forgotten to encode).

What follows is a sequence of commands and results so as to show how to access all features of the fonts. Character tables are also included.

IMPORTANT: If you want to provide patches for the fonts please contact me before you create them. The fonts evolve quickly and you may not have the latest development version and your patches may not apply if created for the published version.

2 How to load the fonts

The simpler way to load the fonts is through the `fontsetup` package. The command

```
\usepackage[default]{fontsetup}
```

will load the Book weight of the NewCM family,

```
\usepackage[olddefault]{fontsetup}
```

will load the Regular weight, and

```
\usepackage[sansdefault]{fontsetup}
```

will load the Sans Serif NewCM family.

Notice that the Mono family has Italic fonts only for Regular and Book weights but for the Bold it has Oblique. To avoid confusion fontspec files are provided that provide the Oblique shape for Regular and Book using the Fake-Bold fontspec parameter.

Also notice that the fonts support the microtype package for fine typographic tuning. See the documentation of microtype for this.

3 The Latin alphabet

3.1 Ligatures and stylistic alternatives in Latin

The Serif font includes additional ligatures fb ffb ffh ffj ffk fft fh fj ft fk and the same with longs instead of f in the *default* liga table (in addition to the default

¹from Claudio Beccari's Greek.

²from the cmu package.

fi fl ffi ffl ff). It also includes an alternative k (in the cv01 table) and şp çh çk çt st ìl in the dlig table. Finally it also includes “end” versions for the letters a, e, m, n and r in the cv02 table. To access the alternative k load the relative font (here the Book weight) with

```
\setmainfont [CharacterVariant=1]{NewCM10-Book.otf}
```

To load the same font with the dlig table enabled use

```
\setmainfont [RawFeature=+dlig]{NewCM10-Book.otf}
```

and to load the font with endings variations use

```
\setmainfont [CharacterVariant=2]{NewCM10-Regular.otf}
```

Of course the above can be mixed separating the optional arguments with comma, or one can define a custom font say by using

```
\newfontfamily\myfont[<options to enable>]{NewCM10-Book.otf}
```

Book	k	a e m n r	sp çh çk çt st ìl
cv01	k		
cv02		ä ë m̄ n̄ r	
dlig			şp çh çk çt st ìl

3.2 Oldstyle numbers

Typically oldstyle numbers are available in `onum` Lookup and with the `\textsc` if `fontsetup` is loaded. Also available they are with `\oldstylenums`. There are two series, one is with variable widths and one with fixed width for use in tables. The code

```
\oldstylenums{0123456789}\addfontfeatures{Numbers=Tabular}
\textsc{0123456789}
```

gives

```
0123456789
0123456789
```

An alternative design is also provided for the number 1 in cv06. The code

```
\oldstylenums{0123456789}\addfontfeatures{CharacterVariant=6}
\oldstylenums{0\textcolor{red}{1}23456789}
\addfontfeatures{CharacterVariant=6,Numbers=Tabular}
\oldstylenums{0\textcolor{red}{1}23456789}
```

gives

```
0123456789
0I23456789
0I23456789
```



```

\definecolor{colorone}{RGB}{91,0,250}
\definecolor{colortwo}{RGB}{250,0,121}
\definecolor{colorthree}{RGB}{0,204,250}
\definecolor{colorfour}{RGB}{14,250,0}
\definecolor{colorfive}{RGB}{255,150,0}
\definecolor{colorgray}{gray}{0.8}
\newcommand{\emptydiacritic}{\char"034F}
\begin{document}
\Huge
{\ncmtest \color{colorgray}a\color{colorfour}\color{colortwo}%
\emptydiacritic\color{colorthree} \color{colorone} \color{colorfive} }
\end{document}

```



4 Greek

The full Unicode Greek block is supported, which is

- U0370–U03FF for monotonic, where missing glyphs, such as Heta (Ϝ), Pamphilian digamma (ϝ) etc, have been added. For example, it is now possible to write

βιβλίο instead of βιβλίο.

In order to auto select this conversion for middle beta and theta the StylisticSets `ss06` and `ss07` must be enabled with, say,

```
\addfontfeatures{StylisticSet=6,StylisticSet=7}
```

Important: if you add font features like the above in the preamble you need to set them with `\AtBeginDocument`. So in the preamble you must have

```
\AtBeginDocument{\addfontfeatures{StylisticSet=6,StylisticSet=7}}
```

To disable this feature you can do

```
\addfontfeature{RawFeature={-ss06},RawFeature={-ss07}}
```

Notice also that the original Beccari fonts for Greek used only the open theta ϑ. However, this violates the Unicode standard description for this glyph which proposes the closed theta θ. If you want the original behavior you need to enable `ss06`.

Source	βιβλίο	θυμήθηκα
<code>ss06</code> enabled	βιβλίο	θυμήθηκα
<code>ss06</code> and <code>ss07</code> enabled	βιβλίο	θυμήθηκα
<code>ss06</code> and <code>ss07</code> disabled	βιβλίο	θυμήθηκα

- U1F00–U1FFF for polytonic, and

- U10140–U1018F for ancient Greek numbers.

Θεώρημα 4.1 (Πυθαγόρειον) *Ἐν τοῖς ὀρθογωνίοις τριγώνοις τὸ ἀπὸ τῆς τῆν ὀρθῆν γωνίαν ὑποτείνουσας πλευρᾶς τετράγωνον ἴσον ἐστὶ τοῖς ἀπὸ τῶν τῆν ὀρθῆν γωνίαν περιεχουσῶν πλευρῶν τετραγώνοις.*

Small Caps is included (in Mono font too) and all polytonic accents of Greek. Ypogegrammeni is the default for all characters including Small Caps and prosigrammeni is offered as an alternative shape in the `ss01` lookup table:

	ypogegrammeni	prosigrammeni
regular	Ά ἤ Ξ ΑΗΩ	Ά₁ ἤ₁ Ξ₁ Α₁Η₁Ω₁
sans	Ά ἤ Ξ ΑΗΩ	Ά₁ ἤ₁ Ξ₁ Α₁Η₁Ω₁
mono	Ά ἤ Ξ ΑΗΩ	Ά₁ ἤ₁ Ξ₁ Α₁Η₁Ω₁

The prosigrammeni alternates can be accessed with

```
\textprosigrammeni{<text>}
```

or the

```
{\prosigrammeni <text>}
```

of the `fontsetup` package.

4.1 Other character variants

Guillemots (left and right) have a different shape for Greek. For this to work the fonts must be loaded with the `cv04` character variant.

Compare the default guillemots: «» with Greek guillemots: «».

There is a serious problem with Unicode and the Greek anoteleia (U0387); the Greek semicolon. Unicode “thinks” that this character is the same with periodcentered (U00B7). This influences the way keyboards are configured by several vendors such as xorg. Anoteleia is a dot written at x-height and not at 1/2 the x-height as the periodcentered. Although Unicode recognizes the problem³, although they recognize that with their current standard you can not correctly write the Greek language, they refuse to fix it, justifying it by saying the magical words “backwards compatibility” (to a ...mistake, one could add).

NewComputerModern can not allow this, as it defies the purpose of its existence, which is to properly write every supported language. So enabling the CharacterVariant 04 (`cv04`) in addition to correct guillemots for Greek it maps periodcentered (produced by the keyboards (in Greek Linux keyboards by AltGr+q) to proper anoteleia.

It also fixes a long standing issue with the Greek apostrophe (´)(U1FBD) which is not the same with quoteright (´)(U2019). U1FBD named as “Greek Koronis” by Unicode is the proper character.

³personal communication

Another problem that has to do with quotes inside quotes. The internal quotes in Greek should be written with the characters `quotedblleft` and `quotedblbase`⁴. For example, this is correct for Greek

«άλφα “βήτα»,»

But the keyboards only produce `quotesingle` which is already mapped to apostrophe and it is difficult to remember the names `quotedblleft` and `quotedblbase`. So when enabling `cv04` one can define the commands

```
\newcommand\leftgrquotes{\char"201C}
```

and

```
\newcommand\rightgrquotes{\char"201E}
```

for the rare case one needs quotes inside quotes. The `fontsetup` package does this automatically for Greek if the `xgreek` package has been loaded *before* the `fontsetup` package or when the language is set to Greek by, say, the `Babel` package. Otherwise, for non-Greek documents with small passages of Greek, the author may enable `cv04` by creating a custom command such as

```
\newfontfamily\propergreek[CharacterVariant=4]{NewCM10-Book.otf}
```

A phrase with Greek quotes inside quotes, proper anoteleia, and proper apostrophe is

«φώναζε: “απ’ ἐξω την προπαίδεια»,»· σαν εκδίχηση ακουγόταν...

4.2 Prosodic symbols

In Greek philology and in linguistics it is often needed to stack accent-type symbols above letters, even if they are not vowels. Although rare in writings, it is for example valid to place dieresis over the consonants π , τ and χ of the nasal complexes $\mu\pi$, $\nu\tau$, $\gamma\chi$ when it is necessary to show that these are pronounced, as written, voiceless, and not voiced. For example,

κομῆρέσα, αντιανῆτανῆτικός, ελεγχῆτης

(see previous footnote). The fonts support this writing if a combining dieresis is placed after the letter to receive it. The combining dieresis is the character U0308. On Linux desktops this is easily entered pressing `Alt+Control+u`, release them, and type the sequence `0308` and space.

More than that, in linguistics, they need to combine several accents above Greek letters. All this stacking of accents is supported by the fonts. For example, one can write

ᾱ̃ ἑ̃ Ᾱ̃ Ῥ̃ ῶ̃ ᾶ̃ Ḳ̃ †

by placing the combining accents from the Unicode block U0300–U0362 plus the usual Greek accents *after* the letter. So the above was typed as

α⁻⁷ ε^{˘˘} ᾶ[˘] Ῥ[˘] ῶ^{˘˘} ᾶ^{˘˘} Ḳ^{˘˘} †

⁴Μανόλης Τριανταφυλλίδης, *Νεοελληνική Γραμματική της δημοτικής*, Ανατύπωση της έκδοσης του ΟΕΣΒ 1941 με διορθώσεις, Θεσσαλονίκη 2002, σελ. 66, ενότητα 133.

4.5 Support for Papyrology

Papyrology needs to declare that a glyph is missing from the papyrus or the papyrus is worn at this point and the papyrologist adds the missing glyph but it is not clear from the papyrus. This is done by adding a dot below the glyph and it is supported for all Greek glyphs in the upright fonts monotonic or polytonic:

Α Ᾱ ῥΗ ᾗ ῶ ῑ

where in the source we just typed the dot below (char U0323) after the glyph. This feature is supported for the 4th bce and 6th bce Greek in Sans:

Γ Ε Ω Μ Ε Τ Ρ Ι Α Γ Ε Ω Μ Ε Τ Ρ Ι Α

More support for papyrology is available. Check Section 16.

4.6 Support for Chemistry

It happens often that Greek upright characters are needed in Chemistry. People often have trouble with this (and this is why packages such as `chemgreek` exist). If Greek keyboard is available then it is easy; you just type in Greek, say `β-glucan` to get “β-glucan”. But many writers do not have the Greek keyboard enabled, and they do not need to. Usually they type `β-glucan` but the result “β-glucan” is not satisfying. One can use the “up” versions typing `β-glucan` but still the result “β-glucan” looks more Math than Chemistry. To help with this, the `fontsetup` package provides commands such as `\chemAlpha`, `\chemalpha`, `\chemBeta`, `\chembeta`, etc. So this information essentially would only belong to the `fontsetup` documentation if it was not for kappa and rho. If we type in Greek `κ-compound` we get “κ-compound” which is not satisfying, as kappa is too cursive for this use. So the NewCM family provides an alternative kappa for this reason and this is how `\chemkappa` is defined in `fontsetup`:

```
\newcommand{\chemkappa}{\textrm{\char"03F0}}:
```

We write `\chemkappa-compound` and now get “κ-compound”.

(The `\textrm` command in the above definition is there to make the command work in math mode too.) Similar is the situation for `\chemrho` (ρ) and `\chemrhoalt` (ϱ).

5 Cyrillic

Cyrillic is supported using the glyphs from the `cyrillic-modern` (default) and `cmu` package but it has considerable improvements (for example, more glyphs and the quality of the bold sans (see below)). `cmu` design can be enabled with `RawFeature=cmu`.

Default Didot style	cmu style
ЖЗКЯжзфжж	ЖЗКЯжзфжж
<p>Я помню чудное мгновенье: Передо мной явилась ты, Как мимолетное виденье, Как гений чистой красоты. (Пушкинъ)</p>	<p>Я помню чудное мгновенье: Передо мной явилась ты, Как мимолетное виденье, Как гений чистой красоты. (Пушкинъ)</p>

Again, as in Greek there is a different kind of guillemots for Cyrillic which are available in CharacterVariant 3 (cv03). Compare:

Defaults guillemots: «» Cyrillic guillemots: «» Greek guillemots: «»

Same is the situation with Cyrillic emdash which is shorter than the default:

Default emdash: —

Cyrillic emdash: —

6 Hebrew

The Hebrew blocks U0590–U05FF and Hebrew Presentation forms UFB1D–UFB4F are fully covered. and A few letters from Hebrew:

צֵלְשֵׁשׁוֹרֵתְגַבְרָא

7 Coptic and Epact Numbers

The Coptic language is fully supported. This covers the Coptic blocks in the Greek and Coptic Unicode block (U03E2–U03EF), the full Coptic Unicode block (U2C80–U2CFF) and the Coptic Epact Numbers (U102E0–U102FF). A few letters from Coptic and Epact numbers follow:

ΛΟΓΟΣ ΝΑΙΓΥΠΤΙΟΣ > ω ~ ρ ε

8 Cherokee

Both Unicode blocks U13A0–13FF and UAB70–UABBF for Cherokee are supported. The samples below were kindly provided by Sedi Eastwood:

Sample 1:

EGPC:IG6H90M TTP0M
ECP:ICTH90M TTP0M

Live/exist in a manner that there is never a reason or purpose to let go of one another.
(gvjalijvdijadayohisdi ijehesdi
gvtsalitsvditsadayohisdi itsehesdi)

The Devanagari fonts were developed with the help of निरंजन (Niranjan) whose name appears in the copyright section of the fonts and I also thank him for providing the samples below. It should also be noted that the design is original and based on old handwritten books. In old civilizations, such as the Indian one, it is only natural that the design be affected by how the handwritten books look. Ink creates “drop”-like serifs on paper that absorbs it, and the tool used to write also affects the look of the script. All these were taken into account. A Sanskrit sample from बृहदारण्यकोपनिषद् (bṛhadāraṇyakopaniṣad) follows:

<p>ॐ पूर्णमदः पूर्णमिदं, पूर्णात्पूर्णमुदच्यते। पूर्णस्य पूर्णमादाय पूर्णमेवावशिष्यते॥</p> <p>That^a is complete; this^b too is complete. From one complete comes the other. Taking out one complete from the other too results in a complete.</p> <hr/> <p>^athe outer world ^bthe inner world</p>

Next is a beautiful part of a poem in Marathi by तुकाराम (Tukaram) and its translation:

<p>जें कां रंजलें गांजलें। त्यांसि म्हणे जो आपुलें॥ १ ॥ तो चि साधु ओळखावा। देव तेथें चि जाणावा॥</p>	<p>Only the one who treats the downtrodden people equally is a sage^a. One may sense the essence of god there.</p> <hr/> <p>^a “The wise” of course, not the plant.</p>
---	---

Devanagari Unicode letters (range U0900–U097F) are also available as variables (letters) and numbers in the Regular and Book Math fonts. They are available as usually in three weights in the Math fonts so that the color is balanced when in script size (eg in exponents or indices). For this to work a version of fontsetup package greater or equal to 1.8 with options default or olddefault loaded is needed. This is because Devanagari letters are not Math variables in Unicode standard and hence not supported currently as such by the unicode-math package. To show this possibility next is a theorem in Hindi (mixing with Greek):

प्रमेय (Πυθαγόρας (पिथागोरास)) अगर समकोण त्रिभुज के कर्ण की लंबाई को ‘अ’ और अन्य दो भुजाओं की लंबाई को ‘क’ और ‘ख’ कहते हैं, तो भुजाओं की लंबाई के वर्गों की जोड़, कर्ण के वर्ग जितनी होती है, अर्थात् $अ^2 = क^2 + ख^2$ ।

However, if only Devanagari numbers are needed with the source using arabic numerals then one can use the Stylistic Set 04 of the Math font. So the command `\setmathfont[StylisticSet=4]{NewCMMath-Regular.otf}` with source:

```


$$\sum_{n=0}^{\infty} \frac{1}{n!} x^n = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \frac{1}{4!} x^4 + \dots = e^x.$$


$$9! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9.$$


$$2^{2^2}$$


```

will have the following effect:

$$\sum_{n=0}^{\infty} \frac{1}{n!} x^n = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \frac{1}{4!} x^4 + \dots = e^x.$$

$$9! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9.$$

$$2^{2^2}$$

which when the math font is reset to use Arabic numbers with `\setmathfont{NewCMMath-Book.otf}` it gives:

$$\sum_{n=0}^{\infty} \frac{1}{n!} x^n = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \frac{1}{4!} x^4 + \dots = e^x.$$

$$9! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9.$$

$$2^{2^2}$$

11 Medieval Latin and Uncial Greek

The family includes new shapes for medieval and uncial Greek. The `fontsetup` package provides `\textuncial{<text>}` and `{\uncial text}` to use this shape. Let us write a sentence in this shape:

ΤΗ ΕΡΥΘΑΓΩΡΕΑΝ ΤΗΘΕΩΡΗΜΑ ΕΙΝΑΙ ΑΠΟ ΤΑ ΣΗΜΑΝΤΙΚΟΤΕΡΑ ΘΕΩΡΗΜΑΤΑ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ.

Medieval Latin and Uncial Greek fonts will give a lot of “missing slot” warnings if the microtype package is loaded. To suppress these warnings use

```
\DeclareMicrotypeAlias{NewCMUncial10-Book.otf}{TU-empty}
```

and similarly for the other NewCMUncial fonts.

12 Braille

Braille, both 6dot (uni2801–uni283F) as well as 8dit (uni2840–uni28FF) patterns are included in two versions. The Regular font provides the characters for sighted persons (such as teachers) so they can easily see which dots are on and which off. The Sans font contains the true Braille characters. I decided to have the sighted version in the Regular font since a blind person does not need the real Braille pattern, as those are produced by embossers. The Braille patterns here are meant as fonts to typeset text mainly for sighted persons.

	6dot	8dot
Regular version	⋮⋮⋮⋮⋮⋮	⋮⋮⋮⋮⋮⋮⋮⋮
Sans version	⋮⋮⋮⋮⋮	⋮⋮⋮⋮⋮

13 Ipa symbols

IPA symbols are included and following a suggestion of Huanyu Liu the kerning found in `tipa` package has been added here and further improved. Moreover the letters eth, eng, beta, theta and chi exists in IPA-style in the fonts and are accessible in the `ss05` lookup table since they are in a different design from the Latin and Greek letters. You can access this lookup table using the `\textipa` command of the `fontsetup` package.

	Non-IPA	IPA
Regular	ð ɳ β θ χ	ð ɳ β θ χ
Sans	ð ɳ β θ χ	ð ɳ β θ χ

I am grateful to निरंजन (Niranjan) for suggesting and testing all the IPA improvements that follow:

The joining of two characters such as \underline{ts} , \underline{dz} , \widehat{kp} , \widehat{tj} etc is also supported. The low tie is the character U035C and the upper tie is U0361. The \underline{ts} is produced by typing the sequence t then the low tie and then s. Similarly \widehat{kp} is produced by typing the sequence k then the upper tie and then p. For \widehat{tj} and \underline{dz} the fonts have a contextual chaining substitution table that uses a tie which prints lower so it does not touch j and z. In cases that one wants to show the tie as $x \frown y \smile z$ then one needs to enable the `cv5` character variant since the tie characters (U035C and U0361) are marks and not base glyphs. One can do that with a command such as

```
\newfontfamily{\showtie}[CharacterVariant=5]{NewCM10-Book.otf}
```

There is also an older practice that such sequences are joint into a ligature. This is not the modern way of writing but it seems than many people in the linguistics still prefer it. The fonts support this, if one enables the `lipa` table (local ipa) of the fonts. This can be done by adding the `RawFeature+=lipa` to the font specs when loading it and it is done automatically with the commands `\textoldipa{arg}` and `\oldipatext arg` of the `fontsetup` package.

For example, `\textoldipa{ts, tj}` produces ts , tj .

All other symbols of the `tipa` package are supported. Some examples are:

$\text{p}^{\text{̄}}$, $\text{t}^{\text{̄}}$, $\text{l}^{\text{̄}}$, $\text{d}^{\text{̄}}$, $\text{j}^{\text{̄}}$, $\text{s}^{\text{̄}}$, $\text{e}^{\text{̄}}$, $\text{k}^{\text{̄}}$, $\text{m}^{\text{̄}}$, $\text{x}^{\text{̄}}$, $\text{t}^{\text{̄}}$, etc.

It is worth noting that all of the above is also available in the Mono family:

$\text{t}^{\text{̄}}\text{s}^{\text{̄}} | \text{t}^{\text{̄}}\text{j}^{\text{̄}} | \text{t}^{\text{̄}}\text{p}^{\text{̄}} | \text{t}^{\text{̄}}\text{l}^{\text{̄}} | \text{t}^{\text{̄}}\text{d}^{\text{̄}} | \text{t}^{\text{̄}}\text{j}^{\text{̄}} | \text{t}^{\text{̄}}\text{s}^{\text{̄}} | \text{t}^{\text{̄}}\text{e}^{\text{̄}} | \text{t}^{\text{̄}}\text{k}^{\text{̄}} | \text{t}^{\text{̄}}\text{m}^{\text{̄}} | \text{t}^{\text{̄}}\text{x}^{\text{̄}} | \text{t}^{\text{̄}}$

Another issue is that IPA used to use the Italic alternation of “g” even in the upright design. This restriction was lifted in 1949 as can be seen on page 3 in [IPArevis].

Even though the normal upright shape is accepted as an alternation, there are linguists who prefer the older (Italic) shape “g” and hence we are making it the default in the stylistic set dedicated for IPA (ss05).

Another such case is with the “a” shape. It has a different visual form in Italics “a” which resembles to a distinct symbol of IPA leading to a wrong transcription. In order to avoid this confusion, we have used the slanted version “a” of “a” in ss05. The other vowel symbol (ie, ɑ (U0251)) is also available in the fonts.

Next we give a real example of using IPA symbols. The quote below is from *A course in phonetics*, Ladefoged, Peter and Johnson, Keith, (2014), Cengage learning, pp. 285–286, and the `StylisticSet=05` has been enabled:

Other examples of this phenomenon have also been mentioned. We saw that before [ŋ], English does not distinguish between tense and lax vowels. Consequently, the vowel that occurs in, for example, sing has a quality between that of [i] and [ɪ]. Similarly, there is no distinction between tense and lax vowels before [ɹ]. The vowel in here in most forms of American English is also intermediate between [i] and [ɪ]. The principle of maximum perceptual separation also accounts for some of the differences between languages. French has two high rounded vowels, [u] as in *tout* [tu] “all,” and [y] as in *tu* [ty] “you.”

14 All Unicode combining marks

As NewComputerModern has top priority the support of Academic work, all combining marks are supported in all fonts. Please notice that not all slots in these blocks are filled in with glyphs by Unicode standard. Some slots are empty and reserved for the future. All glyphs described by Unicode are in NewCM.

These are the Unicode blocks:

- Combining Diacritical Marks U0300–U036F
- Combining Diacritical Marks Extended U1AB0–U1AFF
- Combining Diacritical Marks Supplement U1DC0–U1DFF
- Combining Diacritical Marks for Symbols U20D0–U20FF
- Combining Half Marks UFE20–UFE2F

All blocks support stacking. Examples, other than the main diacritical block U0300–U036F, follow in the next subsections.

14.1 Extended and Supplement Combining Diacritical Marks

Both blocks are supported:

Extended U1AB0–U1AFF. These are used to represent fine-grained phonetic, prosodic, and tone features, primarily in linguistics, dialectology, and musicology. They are combining characters, so they attach to a base letter. Example:

mǎ mà mâ mǎ mà mà
mǎ mà mà

Used for advanced tone marks and diacritic combinations in phonetics and linguistics (eg, tonal contour notation in African and East Asian languages). For example, Yoruba, Zulu, Mandarin, or reconstructed Proto-Bantu tone patterns.

Supplement U1DC0–U1DFF. Adds additional combining marks used in linguistic fieldwork, IPA transcription, and medieval manuscripts. Example:

ạ̌ ạ̣̌ ạ̣̣̌

Supplementary diacritics used in phonetic transcription and historical linguistics, allowing precise representation of speech sounds and manuscript variants.

Medieval Latin manuscripts often abbreviate -us, -um, per, pro, etc, with special marks that can now be encoded as combining diacritics.

In nō dī nostri Īhu x̄pī (In nō dī nostri Īhu x̄pī)
In nō dī nostri Īhu x̄pī → In nomine domini nostri Iesu Christi

14.2 Diacritical Marks for Symbols and Half Marks

Both blocks are supported:

Marks for Symbols U20D0–U20FF. This block is used in Mathematical notation and the symbols are available in the Math fonts. However, there are needs for this block outside Mathematics. So they have also been added in the text fonts. For example, in Linguistics an arrow may mark a rising or a falling tone as in ā, ǎ. Another example is in automata theory: a circled state \textcircled{q} or a squared proposition $\text{□}p$. Here these symbols are typed as text (this is why they have the metrics of q and p respectively, and consequently the imposed circle does not create additional horizontal space). However, the NewCM Math fonts include the precomposed glyphs for use in mathematics mode (U2460–U24FF). For example, circled p and f are the glyphs U24DF and U24D5 respectively. Commands to access

those in math mode, such as `\circledp` are provided by the style file `newcomputermodern.sty` through the `fontsetup` package. The metrics here are different as the circle has its own metrics. Compare text: \textcircled{p} and \textcircled{f} with math $|\textcircled{p}|$ and $|\textcircled{f}|$.

Half Marks UFE20–UFE2F. Again constuctions using these symbols are available in Math fonts. They are also added to text fonts for non-math cases. An example is in linguistics for double diacritics over two letters: \widehat{ab} , double tilde spanning two characters used for nasalization spanning a diphthong or a contour tone that applies to two vowels: \widetilde{ou} .

Also used in Medieval and textual criticism. For example, spanning macrons: \bar{no} . Represents $\bar{n}o$ (“no-” from nobis, noster, etc.) But with a macron spanning both letters to indicate the omitted syllable. Scholars of medieval Latin and Greek use exactly this.

To give an example that relates to Cyrillic we show the “titlo” used in early Cyrillic manuscripts and Old Slavonic: used to mark abbreviations, contractions, sacred names, and numeric values. The FE2E/FE2F pair lets a single titlo be encoded as two combining halves that attach to the leftmost and rightmost letters of the span and visually join into one continuous overline. However, when more than two characters are used, or characters with different heights, there is no provision by Unicode how these characters should join. Necessarily the titlo will break. However, a new command `\titlo` has been introduced in `fontsetup` package (that applies to `newcomputermodern.sty` as well) that produces proper titlos on more than two letters and of unequal heights. I thank David Carlisle for his help with this.

Use case: when an editor wants a single titlo covering two or more adjacent Cyrillic letters, they insert the left-half after the first letter and the right-half after the last letter of the span.

- Two-letter titlo; Characters: U+043A U+FE2E U+043D U+FE2F: $\overline{кн}$
- Three-letter titlo with different heights; Characters: U+043A U+FE2E U+0430 U+0431 U+FE2F: $\overline{каб}$
- Titlo used with a numeral (titlo often marks numeric use of letters); Characters: U+0432 U+FE2E U+0438 U+FE2F: $\overline{бѡ}$
- Corrected output using the `\titlo` command with `\titlo{каб}`: $\overline{каб}$.

15 Editorial Marks

NewComputerModern supports a range of combining editorial marks intended for use in scholarly text, critical editions, manuscript transcription, and linguis-

tic annotation. These marks function as textual modifiers applied to arbitrary base characters and are distinct from precomposed mathematical symbols.

15.1 Scope and Intended Use

Editorial marks are used to indicate:

1. manuscript witnesses and scribal hands in critical apparatus,
2. editorial interventions and conjectural readings,
3. layered annotation in palaeography and codicology,
4. feature marking and classification in linguistics.

In these contexts, the mark conveys editorial status, not mathematical meaning.

15.2 Combining Enclosing Marks

The font provides full support for combining enclosing marks, including:

U20DD COMBINING ENCLOSING CIRCLE

U20DE COMBINING ENCLOSING SQUARE

U20DF COMBINING ENCLOSING DIAMOND

These characters attach to a preceding base glyph. The enclosing shape adapts to the base character's size and style (e.g. italic, small caps, Greek, Cyrillic), enabling consistent use in running text.

Example (editorial sigla):

ⓐ ⓑ Ⓕ ⓒ Ⓖ Ⓢ

(the last example is in small caps). Editorial marks are positioned using dedicated anchors in the text font. The enclosing shapes are centered optically relative to the base glyph rather than aligned to the mathematical axis. Letters with ascenders or descenders are adjusted to preserve visual balance inside the enclosure.

Editorial combining marks should not be confused with precomposed circled letters used in mathematical notation (eg automata theory). Mathematical circled symbols are atomic glyphs provided by math fonts and are accessed via Unicode or math alphabets. Combining enclosing marks in NewComputerModern are intended exclusively for textual and editorial use.

16 Phonetic Extensions and Supplement

The block U1D00–U1D7F contains letters and small capitals used in the IPA and related phonetic alphabets, and the block U1D80–U1DBF adds further small capitals and modifier letters for phonetic and prosodic notation. The small letters here are *not* to be confused with typographical small capitals. The designs here are at the x-height of the font which is smaller than small capitals. Small capitals should still be called with commands such as `\textsc` or similar. Examples:

- (i) [ɸ̥ɑɾɪ] [ɸ̥ɑɾɪ] [ɸ̥ɑɾɪ] [ɸ̥ɑɾɪ]
- (ii) [m̥ɑn] [m̥ɑn] [m̥ɑn] [m̥ɑn m̥ɑn] meaning “man” pronounced with a bilabial trill
- (iii) [ɸ̥ɑm̥ɑ] [ɸ̥ɑm̥ɑ] [ɸ̥ɑm̥ɑ] [ɸ̥ɑm̥ɑ] could represent a voiced implosive /ɸ̥ama/ in a phonetic study.
- (iv) t^ha k^wa n̥i l̥ʷa t^ha k^wa n̥i l̥ʷa t^ha k^wa n̥i l̥ʷa aspirated t, labialized k, palatalized n, velarized l
- (v) t^ʃ d^ʊ n^ɹ t^ʃ d^ʊ n^ɹ t^ʃ d^ʊ n^ɹ detailed articulatory descriptions and experimental phonetics.

Editors of medieval and ancient texts sometimes mark uncertain or supplied letters with combining strokes or dots above/below.

ṡaṅctus ṡaṅctus ṡaṅctus

may be used to indicate a word reconstructed or uncertain in a manuscript. More examples from the block U1D80–U1DBF:

Upright	Italic
ḃ ḟ Ḑ ḙ Ḟ ḡ ḣ ḥ Ḱ Ḷ Ḹ Ṁ ṁ Ṃ Ṇ Ṱ Ṷ Ṹ Ṻ Ṽ ṽ Ṿ Ṽ ṽ Ṿ	<i>ḃ ḟ Ḑ ḙ Ḟ ḡ ḣ ḥ Ḱ Ḷ Ḹ Ṁ ṁ Ṃ Ṇ Ṱ Ṷ Ṹ Ṻ Ṽ ṽ Ṿ Ṽ ṽ Ṿ</i>
Sans Upright	Sans Oblique
ḃ ḟ Ḑ ḙ Ḟ ḡ ḣ ḥ Ḱ Ḷ Ḹ Ṁ ṁ Ṃ Ṇ Ṱ Ṷ Ṹ Ṻ Ṽ ṽ Ṿ Ṽ ṽ Ṿ	<i>ḃ ḟ Ḑ ḙ Ḟ ḡ ḣ ḥ Ḱ Ḷ Ḹ Ṁ ṁ Ṃ Ṇ Ṱ Ṷ Ṹ Ṻ Ṽ ṽ Ṿ Ṽ ṽ Ṿ</i>

Even though these are designed for phonetics, some medievalists and historical linguists use these modifier letters to show reduced vowels, superscripts, or lenited consonants in Old English or Middle English.

h^ʷ’s he^ʷ’rɪŋ ç^ʰld h^ʷ’s he^ʷ’rɪŋ ç^ʰld h^ʷ’s he^ʷ’rɪŋ ç^ʰld

could represent small, superscript-like vowels in “his”, “hearing”, “child”, when reproducing manuscript letter size distinctions or marginal additions. Modifier letters used to reproduce reduced or superscript vowels from early English manuscripts and paleographic editions.

17 Currency Symbols

Support for currency symbols block U20A0–U20C0 in all text fonts (except unicals).

Roman: € ¢ £ ¤ ¥ ¨ ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
 Sans: € ¢ £ ¤ ¥ ¨ ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
 Mono: € ¢ £ ¤ ¥ ¨ ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾

Interesting is the design of Spanish Peseta in the Mono family (seen enlarged in the margin) were we have to fit essentially three letters in the width of the Mono font.



Especially for the Rupee sign (U20B9) the upright serified non-Devanagari fonts have the “latinized” version, but to avoid changing fonts they also provide the Devanagari version in CharacterVariant 7. So the code

```
\begin{LARGE}
\begin{center}
100\,₹ \hspace{2cm} {\addfontfeatures{CharacterVariant=7}100\,₹}
\end{center}
\end{LARGE}
```

gives

100 ₹ 100 ₹

18 Bold Sans















lm fonts and cmu fonts do not contain a properly made BoldSans. Their BoldSans is a stroke-extension of the Sans with rounded corners. NewCM fixes that and provides a true BoldSans:

LM	NewCM
X E E	X E Я Д Л

19 Food Allergies

Food allergy symbols have long been proposed to be accepted to Unicode standard but there has not been any progress up to now. The Sans 10 Regular and

Book include standard allergy symbols in the Private Use Area. Each glyph is named after what it represents. For example, the soya symbol is named “soya” so copying the symbol from a pdf file and pasting elsewhere you will get its name, that is “soya”. The symbols are in positions U+E033 to U+E040 and can be accessed using the commands of the next table.

<code>\char"E033</code>	 CRUSTACEANS	<code>\char"E034</code>	 EGGS
<code>\char"E035</code>	 GLUTEN	<code>\char"E036</code>	 FISH
<code>\char"E037</code>	 LUPIN	<code>\char"E038</code>	 MILK
<code>\char"E039</code>	 MOLLUSCS	<code>\char"E03A</code>	 MUSTARD
<code>\char"E03B</code>	 PEANUT	<code>\char"E03C</code>	 SESAME
<code>\char"E03D</code>	 SOYA	<code>\char"E03E</code>	 TREENUTS
<code>\char"E03F</code>	 CELERY	<code>\char"E040</code>	 SO ₂

20 Unicode Math coverage and options

NewCM provides full Unicode math support, that is all Mathematics Unicode Slots presented in <http://www.unicode.org/charts/> in the Math weights, Regular, Book and Bold. These blocks are:

Mathematical Symbols

Arrows (uni2190–uni21FF)

Supplemental Arrows-A (uni27F0–uni27FF)
 Supplemental Arrows-B (uni2900–uni297F)
 Supplemental Arrows-C (u1F800–u1F8FF)
 Additional Arrows (uni2B00–uni2BFF)
 Miscellaneous Symbols and Arrows (uni2B00–uni2BFF)

Mathematical Alphanumeric Symbols

(u1D400–u1D7FF)
 Arabic Mathematical Alphabetic Symbols
 (u1EE00–u1EEFF)
 Letterlike Symbols (uni2100–uni214F)

Mathematical Operators

(uni2200–uni22FF)
 Basic operators: Plus, Factorial
 (uni0000–uni007F)
 Division, Multiplication
 (uni0080–uni00FF)
 Supplemental Mathematical Operators
 (uni2A00–uni2AFF)
 Miscellaneous Mathematical Symbols-A
 (uni27C0–uni27EF)
 Miscellaneous Mathematical Symbols-B
 (uni2980–uni29FF)
 Floors and Ceilings (uni2308–uni230B)
 Invisible Operators (uni2061–uni2064)

Geometric Shapes (uni25A0–25FF)

Additional Shapes (uni2B00–uni2BFF)
 Box Drawing (uni2500–uni257F)
 Block Elements (uni2580–uni259F)
 Geometric Shapes Extended (u1F780–u1F7FF)

Unfortunately, the `unicode-math` package does not provide commands currently for the hundreds of extra glyphs that have been added in order to fully cover the above Unicode ranges. The user can consult the Unicode charts at the above link and access the required glyph with `\char"#` where `#` is the Unicode number of the slot the glyph belongs to.

For example, `\char"2BDA` will give the Hygeia symbol (uni2BDA) the Rod of Asclepius as shown above (grayed and scaled $\times 8$). The glyph that appeared in TUGboat (see [AT]), being more realistic will be moved to a new font in the future with ornaments.



20.1 Bold Math

A complete math font, such as NewCM, contains all alphabetic characters in bold too. These characters are typically accessed using the `\mathbf` command.

However, this is not true for bold versions of symbols. This creates difficulties when a user has some math in chapter or section titles, or when a user wants to create a poster with colored background. In the later case the Regular and Book weights look too light (especially with dark backgrounds) and one is in need of a real Bold Math font, that has everything in Bold.

NewCM, starting from version 6.0.0 provides an independent Bold Math font for the first time for a Computer Modern font family. Let us compare:

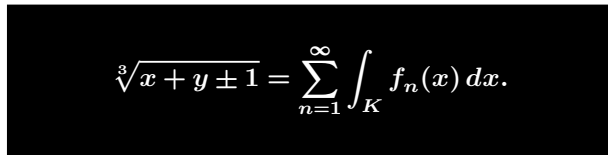
Book inline: $\sqrt[3]{x + y \pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx$ and the same in display

$$\sqrt[3]{x + y \pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx.$$

Bold inline: $\sqrt[3]{x + y \pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx$ and the same in display

$$\sqrt[3]{x + y \pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx.$$

Bold Math with colors inverted:



In order to use the Bold font for chapter and section titles, when using the `default` or `olddefault` options of the `fontsetup` package, change the math version to bold with `\mathversion{bold}` *before* the commands for chapter and section and switch back to normal with `\mathversion{normal}` *afterwards*.

The `unicode-math` package, according to its documentation has still some troubles with “versions” and the `range` options. These seem to affect at least the calligraphic and script math alphabets. In normal version for example the commands `\symcal` and `\symscr` work as expected, but when one switches to the bold version the `\symscr` fails. In this case one can use `\symbfscr`: \mathcal{A} \mathscr{A} (which was `\symcal A` `\symscr A`).

20.2 Sans Serif Math

As of version 7.0.0, the family includes a full-featured Math Sans font. Up to now such a font did not exist in the CM family (although partial solutions existed), and it posed a serious problem for scientific writing especially in the preparation of presentations. The font supplied with NewCM covers all Unicode math slots but it also provides some new features. The lowercase Latin alphabet has been completely re-worked so that it really looks as it should when used for Math variables. The letters are

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|

In Large

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|

Moreover the calligraphic capitals have been re-worked to match better with the sans serif style (see subsection 20.4 for how to select them)

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Same goes for the `\mathbb/\symbb` capital letters. They have been adjusted to match the Sans design, both in style and weight:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

It is also worth noticing that the SansSerifed and the Serifed letters have swapped slots in the font. So if using the MathSans font and you write for example `\symsf{ABCDabcd}` you will get the Serifed version(!): ABCDabcd. This choice facilitates converting a document with Serifed fonts using Sans for emphasis or differentiation to keep these characteristics when changed to Sans.

One can see the Sans Serif Math font in action in the provided file `testmath-newcm.pdf` which comes from the \mathcal{MS} -L^AT_EX bundle (here the logo is written in NewCMSans).

20.3 Optical sizes for more glyphs

So far the fonts provided optical sizes for 1st and 2nd order exponents for letters. This was not true though for binary operators and some symbols commonly used in mathematics. For example, the `\perp` symbol (\perp) often appears in 2nd order exponents and then it appeared very thin. Some printers could even hardly print its thin lines. Now, such glyphs plus several binary operators, such as $+$, $-$, \pm , \dagger etc are now provided in optical sizes, so that expressions such as

$$+^{++} \perp^{\perp\perp} ** \frac{|P_{F^\perp}(K)|}{e^{(x+y)^*}}$$

appear on screen and print properly on printers. Zoom or print and compare with `latinmodern-math` font:

$$+^{++} \perp^{\perp\perp} ** \frac{|P_{F^\perp}(K)|}{e^{(x+y)^*}}$$

20.4 Math Script

Calligraphic letters are accessed as usual with `\mathcal` or `\symcal`, producing

ABCDEFGHIJKLMNOPQRSTUVWXYZ

However, mathematicians often need a second level of “scriptness”. The fonts provide an alternative calligraphic, a script design at StylisticSet 1. For this to work one has to re-set the math font using

`\setmathfont[StylisticSet=1]{NewCMMath-Book.otf}`

(or the Regular version). So the following code

the engine does not see the characters H and period as consecutive characters to apply the kern, and the kern is lost.

The code of the table above is as follows:

```
\begin{tabular}{c|c}
  Before & Current (with Lua) \\ \hline
  {\setmathfont{latinmodern-math.otf}\Large}
  ${\symcal H.}^{\{\symcal H,\}^{\{\symcal H,\}}}$
  ${H\/.}^{\{H.\}^{\{H\/,,\}H,\}^{\{H\/,H,\}}}$
  & {\setmathfont{NewCMMath-Regular.otf}\Large}
  ${\symcal H.}^{\{\symcal H,\}^{\{\symcal H,\}}}$
  ${H\/.}^{\{H.\}^{\{H\/,,\}H,\}^{\{H\/,H,\}}}$
\end{tabular}
```

We also notice that Lua \TeX has a parameter set by the command `\mathitalicmode`

with default value 0. If set to 2 with `\mathitalicmode=2` then Lua \TeX uses regular letter kerning in math mode but breaks other things like the end-points of integrals (see [web1]). We conclude this subsection realizing that there is no single engine (among \TeX and Lua \TeX) that supports all features.

20.6 Blackboard Bold

The fonts contain as default the AMS blackboard bold. These are:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 0123456789 π γ Γ Π Σ *Ddeij*

They also contain a blackboard bold that matches the design of Computer Modern but respecting the fact that most users have been used for a long period to the AMS bb design. Compare the default

$\mathbb{R} \in \mathbb{R} \quad \mathbb{Q} \in \mathbb{Q}$

with the new design

$\mathbb{R} \in \mathbb{R} \quad \mathbb{Q} \in \mathbb{Q}$

To access this design one needs to load the math font enabling the `ss03` stylistic set using for example

`\setmathfont[StylisticSet=3]{NewCMMath-Book.otf}` Then the above blackboard bold design changes to

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 0123456789 π γ Γ Π Σ *Ddeij*

If using the latest `fontsetup` then you can choose the NewCM blackboard bold with the option `newcmbb`.

20.6.1 Small Caps for Blackboard Bold

It happens one to need Blackboard Bold in Small Caps. For example, this may arise when running heads are in small caps and contain a Blackboard Bold symbol. Consider for example the case when a chapter is named “The Lebesgue measure on \mathbb{R} ”. Then the running head in small caps will look awkward:

THE LEBESGUE MEASURE ON \mathbb{R}

In such cases, a small caps version of the number sets is needed. NewCM provides the needed glyphs in `ss05`. So setting `\setmathfont[StylisticSet=5]{NewCMMath-Book.otf}` the above running head becomes:

THE LEBESGUE MEASURE ON \mathbb{R}

These glyphs are provided in the CM style (instead of the AMS `bb`) by enabling `ss03` too. So the command `\setmathfont[StylisticSet=3,StylisticSet=5]{NewCMMath-Book.otf}` will produce:

THE LEBESGUE MEASURE ON \mathbb{R}

The same is true for the Regular and Bold Math fonts.

20.7 Upright and extensible integrals

The Math fonts (both Regular and Book weights) include upright integrals in the `ss02` `StylisticSet`. Use with

```
\setmathfont[StylisticSet=2]{NewCMMath-Book.otf}
or
\setmathfont[StylisticSet=2]{NewCMMath-Regular.otf}
```

or use the `upint` option of the `fontsetup` package with

```
\usepackage[upint,default]{fontsetup}
```

for the Book weight, or

```
\usepackage[upint,olddefault]{fontsetup}
```

for the regular weight.

Moreover, extensible integrals are supported by the fonts but *NOT* by the Unicode TeX engines. The following code is a trick so that extensible integrals can be constructed using Lua^ATeX. The result is shown at the end of the article. What the code below does, is that it defines the slot `uni222B` (integral) as a delimiter. And then this is extended as a delimiter with the mechanism that the font provides.

```

\documentclass{article}
\usepackage[default]{fontsetup}
\begin{document}
$
\Uleft\Udelimiter 0 0 "222B
\begin{pmatrix}
1\2\3\4\5\6\7\8\9
\end{pmatrix}
\Uright.
$
\end{document}

```

$$\left(\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \right)$$

20.8 Additional and alternative characters in Math

20.8.1 Alternative symbol for the empty set

There is a mismatch between the \TeX world for the empty set and the Unicode standard. For \TeX the default empty set is \emptyset , that is a slashed zero. Unicode suggests that the proper character is \varnothing , and suggests the switch to the slashed zero if \varnothing is followed by the Variation Selector UFE00. However, this method is not yet supported by Unicode \TeX engines. Since the fonts must comply to the Unicode standard the math fonts follow the Unicode suggestion. To make things work as expected in \TeX , the user must load the Math fonts with `CharacterVariant=1`. For example, with

```
\setmathfont[CharacterVariant=1]{NewCMMath-Regular.otf}
```

This will map `\emptyset` to \varnothing . If `CharacterVariant=1` is not loaded the `\emptyset` will give the proper Unicode character, that is \emptyset .

If the user want to have access to both symbols, \emptyset with `\emptyset` and \varnothing with `\varnothing` in the same document then this can be done with the next four lines.

```

\setmathfont[CharacterVariant=1]{NewCMMath-Regular.otf}
\AtBeginDocument{%
\renewcommand{\emptyset}{\char"2205}%
\renewcommand{\varnothing}{\char"2300}}

```

The `fontsetup` package defaults to the \TeX -default empty set (\emptyset) and provides the option ‘`varnothing`’ to easily switch to the alternative character.

20.8.2 Additional symbols

It also provides four more arrows that correspond to the commands

```

\nrightrightarrows ( $\rightrightarrows$ )  \nleftleftarrows ( $\nleftleftarrows$ )
and
\twoheadhookrightarrow ( $\twoheadhookrightarrow$ )  \twoheadhookleftarrow ( $\twoheadhookleftarrow$ )

```

and supported by the `default` and `olddefault` options of the `fontsetup` package. These symbols are not in the Unicode Standard and so they are added in the Private Area of the fonts.

20.8.3 Additional operators

An operator for convolution of functions seems to have long been forgotten from both Unicode and the TeX world. The `default` and `olddefault` options of the `fontsetup` package define a new command `\convolution` which behaves just like the `\sum` and `\int` operators. The convolution of N functions in inline mode: $\ast_{n=1}^N f_n$ and the same in display mode:

$$\ast_{n=1}^N f_n.$$

20.8.4 “Smoother” changing radicals

One more radical size has been added that improves the way radical sizes change. Compare the previous state

$$\sqrt{q} \quad \sqrt{q_1^2} \quad \sqrt{q_1^{23}} \quad \sqrt{\frac{1}{2}}$$

with the new one

$$\sqrt{q} \quad \sqrt{q_1^2} \quad \sqrt{q_1^{23}} \quad \sqrt{\frac{1}{2}}.$$

20.9 Notes on Mathematics

Extensible tildes and hats produce different results with XeLaTeX and LuaLaTeX because these unicode engines treat differently the width of characters. In particular, XeLaTeX handles italic correction as part of the character width but LuaLaTeX does this only if a character follows. This affects how extensible accents like `\widehat` or `\widetilde` select the proper size. With XeLaTeX, `\widetilde{Y}` `\widetilde{X}` will give the expected result $\widetilde{Y}\widetilde{X}$; but with LuaLaTeX the letter Y is narrower (and its italic correction is not taken into account if it stands alone) and gets the plain tilde size, as in \tilde{Y} . To bypass this with LuaLaTeX one has to use a zero width character after a letter such as Y so that LuaLaTeX takes into account the italic correction of Y . For example with

`\widetilde{Y\Uchar"200D}`

the result will be correct (U200D is the “Zero Width Joiner” in Unicode).

I thank Ulrike Fischer for this solution.

21 The Medieval Latin and Uncial Greek glyph complement

Table 1: NewCMUncial10-Book.otf (features: ExternalLocation=../otf/)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Basic Latin																
U+0020-002F		!	"	#	\$	%	&	'	()	*	+	,	-	.	-
U+0030-003F	o	1	2	3	4	5	6	7	8	9	:	;	<	=	>	-
U+0040-004F	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ		
U+0050-005F	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	-	-	-	-	-
U+0060-006F	-	ⓐ	ⓑ	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ
U+0070-007F	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	-	-	-	-	-
Latin-1 Supplement																
U+00A0-00AF	-	-	-	-	-	-	-	-	-	¨	-	-	-	-	-	-
U+00B0-00BF	-	-	-	-	-	-	-	-	·	-	-	-	-	-	-	-
Greek and Coptic																
U+0370-037F	-	-	-	-	´	,	-	-	-	-	-	-	-	-	-	;
U+0380-038F	-	-	-	-	´	ˆ	Ⲁ	·	Ⲅ	Ⲇ	Ⲉ	Ⲋ	-	Ⲍ	-	Ⲑ
U+0390-039F	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ
U+03A0-03AF	Ⲑ	ⲑ	-	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ
U+03B0-03BF	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ
U+03C0-03CF	Ⲑ	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ	ⲗ	ⲙ	ⲏ	ⲑ	ⲓ	ⲕ
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Total number of glyphs shown from NewCMUncial10-Book.otf: 160

22 The Aegean Numbers glyph complement

<code>\aegeanseparator</code>	ι	<code>\aegeaneighthundred</code>	⊞
<code>\aegeanseparatordot</code>	·	<code>\aegeanninehundred</code>	⊞⊞
<code>\aegeancheckmark</code>	×	<code>\aegeanonethousand</code>	⊕
<code>\aegeanone</code>	ι	<code>\aegeantwothousand</code>	⊕⊕
<code>\aegeantwo</code>	″	<code>\aegeanthreethousand</code>	⊕⊕ ⊕
<code>\aegeanthree</code>	‴	<code>\aegeanfourthousand</code>	⊕⊕
<code>\aegeanfour</code>	‖	<code>\aegeanfivethousand</code>	⊕⊕ ⊕
<code>\aegeanfive</code>	‖‖	<code>\aegeansixthousand</code>	⊕⊕
<code>\aegeansix</code>	‖‖‖	<code>\aegeanseventhousand</code>	⊕⊕ ⊕
<code>\aegeanseven</code>	‖‖‖‖	<code>\aegeaneightthousand</code>	⊕⊕ ⊕
<code>\aegeaneight</code>	‖‖‖‖‖	<code>\aegeanninethousand</code>	⊕⊕ ⊕
<code>\aegeanine</code>	‖‖‖‖‖‖	<code>\aegeantenthousand</code>	+
<code>\aegeanten</code>	-	<code>\aegeantwentythousand</code>	⊕⊕
<code>\aegeantwenty</code>	=	<code>\aegeanthirtythousand</code>	⊕ ⊕
<code>\aegeanthirty</code>	≡	<code>\aegeanfourtythousand</code>	⊕⊕
<code>\aegeanfourty</code>	==	<code>\aegeanfiftythousand</code>	⊕⊕ ⊕
<code>\aegeanfifty</code>	≡≡	<code>\aegeansixtythousand</code>	⊕⊕
<code>\aegeansixty</code>	≡≡≡	<code>\aegeanseventythousand</code>	⊕⊕ ⊕
<code>\aegeanseventy</code>	≡≡≡≡	<code>\aegeaneightythousand</code>	⊕⊕ ⊕
<code>\aegeaneighty</code>	≡≡≡≡≡	<code>\aegeanninetythousand</code>	⊕⊕ ⊕
<code>\aegeanninety</code>	≡≡≡≡≡≡	<code>\aegeanweightbaseunit</code>	⊞
<code>\aegeanonehundred</code>	°	<code>\aegeanweightfirstsubunit</code>	⊚
<code>\aegeantwohundred</code>	°°	<code>\aegeanweightsecondsubunit</code>	⊞
<code>\aegeanthreehundred</code>	°°°	<code>\aegeanweightthirdsubunit</code>	⊚
<code>\aegeanfourhundred</code>	°°°°	<code>\aegeanweightfourthsubunit</code>	⊕
<code>\aegeanfivehundred</code>	°°°°°	<code>\aegeandrymeasurefirstsubunit</code>	⊞
<code>\aegeansixhundred</code>	°°°°°°	<code>\aegeanliquidmeasurefirstsubunit</code>	⊕
<code>\aegeansevenhundred</code>	°°°°°°°	<code>\aegeansecondsubunit</code>	⊕
		<code>\aegeanthirdsubunit</code>	⊕

References

- [AT] Antonis Tsolomitis, *The NewComputerModern font family*, TUGboat Vol. 42, No. 1, 2021.
- [IPAreV] Council actions on revisions of the IPA, Phonetic Representation: b) Revision of the IPA, Journal of the International Phonetic Association, Volume 23, Issue 1, June 1993, pp. 32–34.
- [web1] <https://tex.stackexchange.com/questions/697498/luatex-unicode-math-and-italic-correction>