The TEX Gyre Fonts

Their development and their use in LaTEX and ConTEXt

Abstract
This article describes the TEX Gyre fonts, their development, and how to use them in LaTEX and ConTEXt. NOTE: I have not been personally involved in the development of these fonts. I have just tried to dig up the information about them from public (but sometimes a bit difficult to find) information.

Keywords
LaTEX, ConTEXt, fonts, OpenType, Type 1, PDFTEX, XeTEX, LuaTEX, Computer Modern, Latin Modern, TEX Gyre, Unicode, encoding, MetaPost, MetaType1

Introduction

In 1985 Apple released a new printer, the *LaserWriter* for the amazing price of $6,995. It used quite a new concept: it had a *Postscript* interpreter embedded.

The Computer Science department where I worked at that time bought one of these, and for quite a time this was the printer we used to print our documents, including those generated with TEX (but also with the Unix program `troff`).

PostScript was described as a *page description language*, i.e., a language to describe graphical pages, but in fact it was a complete programming language with a graphical engine built-in. It had just been developed by Adobe Systems, which had been founded a few years earlier.

The *Laserwriter* contained four built-in PostScript font families: Times, Helvetica, Courier, and Symbol, the first three also including bold and italic/oblique variants.

The next release of Postscript (PostScript Level 2) included additional standard fonts: ITC Avant Garde Gothic, ITC Bookman, Helvetica (extra variants), New Century Schoolbook, Palatino, ITC Zapf Chancery, and ITC Zapf Dingbats. With all the variants this added up to 35 fonts. These fonts are also available in Adobe's PDF processors.

These fonts could also be licensed for use in other programs, but they were commercial fonts, so they couldn't be distributed freely, except (partially) when embedded in documents that used them. This was a problem for the Open Source community.

In 1988 L. Peter Deutsch developed a Postscript clone *Ghostscript* for the GNU Project, and released it under the GNU General Public License (GPL)[1].

Ghostscript could not include the Adobe fonts, of course. It included several fonts that were developed as clones of the Adobe fonts, with special emphasis on having the same *metrics* of these, although the pixels could be slightly different. Later these were replaced by fonts from the German foundry URW++[2], which were placed under the GPL in 1996. They have 35 fonts similar to the original Adobe fonts, but pixelwise slightly different, and under different names. In 2009 these fonts have also been released under the LaTeX Public Project License (LPPL), which is less restrictive than the GPL. These fonts are commonly called the URW Base 35 fonts or URW Core 35 fonts. They can also be used with other software.

The TEX Gyre collection of fonts is a reimplementation of 33 of these fonts (not included are the Symbol and Zapf Dingbats fonts), based on the URW++ fonts. They have a much larger collection of glyphs (character forms), so that they are also usable in other than Western-European languages. They should cover all European languages that have a Latin-based script[8] as well as some other Latin-based alphabets such as Vietnamese and Navajo. They are also available in OpenType format, which gives them more options, like in the area of ligatures.

The TEX Gyre fonts have been developed by the GUST e-foundry[3], a project from GUST, the Polish TEX Users Group (Grupa Użytkowników Systemu TeX). They have developed a tool MetaType1, a tool to develop Postscript Type 1 and OpenType fonts from MetaPost sources. They have used this to convert and extend several TEX fonts. The actual funding is by DANTE e.V., GUTenberg, NTG, TUG, CS TUG and GUST. GUST is very active in TEX related font (type) activities through its e-foundry.

MetaType1

MetaType1[26, 25, 27] is a collection of scripts and programs that can be used as a tool for creating Type 1 fonts using MetaPost. MetaPost[16] is a derivative of MetaFont; it has a similar syntax as MetaFont. There

are, however, some differences between MetaFont and MetaPost, for example MetaPost does not support certain operations for erasing pixels, or rounding with respect to pixels. And the MetaPost program by default generates Postscript images[4] rather than the bitmaps that MetaFont produces (MetaPost can also output SVG or PNG, but this is not relevant for fonts). On the other hand, MetaPost has new functionality that MetaFont does not have, for example the possibility of measuring of the length of a path (`arclength` and `arctime` operators). Also important for fonts is the fact that MetaPost has the ability to generate `TFM` files, just like MetaFont. So this can be used to generate outlines of a MetaFont font from its MetaFont sources, but these sources probably have to be adapted to comply with the differences between MetaFont and MetaPost and with the restrictions of Type 1 fonts, such as lines crossing each other, which is OK in a Postscript image, but not allowed in a Type 1 font. This requires a fair amount of manual work. Furthermore many MetaFont characters are created by drawing (thick) lines with a pen, and these have to be converted to outlines.

Additionally, the MetaType1 distribution contains the `pf2mt1` package for converting converting a Type 1 font into a MetaPost source. This conversion preserves as many of the font parameters as possible. The resulting font can then be adapted and extended and converted by MetaType1 into a Type 1 font. For example, new accented letters can be constructed from unaccented letters and existing or new accents.

Another thing that has to be added to the generated Type 1 font is *hints*. These are Type 1 commands that control the discretization of the outlines; see [17] for details. MetaType1 did include modules for setting hints but later the generation of hints was done with the Open Source program `FontForge`.

MetaType1 is available on CTAN[5] and is included in TeXlive.

Taco Hoekwater described this process of converting MetaFont sources to Type 1 fonts with MetaPost quite some time ago[17]. He used a program `MetaFog`[35] that converts the pen strokes to outlines, and also resolves crossing outlines. So with `MetaFog` most MetaFont constructions can be used unlike with MetaType1, but it is a commercial program[6]. Moreover it runs only on Windows systems. MetaType1 by design uses only free software, so this is not an option for them.

Another possibility for generating Type 1 fonts from MetaFont sources is to generate bitmaps with MetaFont and then use tracing software to convert the bitmaps to outlines. However, this is not an exact process; it gives only an approximation. An example of this approach is TEXtrace by Péter Szabó[54], which uses MetaFont to generate bitmaps of the font at a high resolution (7227 DPI) and then uses the AutoTrace program by Martin Weber[58] or the `potrace` program by Peter Selinger[52] to make outlines of these bitmaps in Type1 format. The conversion is fully automatic and does not need manual intervention. The process is described in Péter Szabó's article[53].

Karel Píška reported on his experience with TEXtrace[47] and compared the different approaches (with TEXtrace, MetaType1 and `MetaFog`) of converting MetaFont fonts to Type 1 in [48]. He concludes that the quality of the converted fonts from TEXtrace is not always optimal, and to get optimal quality outlines manual postprocessing is still required; see also [47]. Moreover, the generated Type 1 fonts don't include any hints, so for best results these should also be added.

MetaType1 now has a successsor Fontplant[7].

The Computer Modern Fonts

The story actually starts with the construction of the Latin Modern fonts around 2002. This was the first big font project with MetaType1, although it had been used before to construct the Polish font Antykwa Półtawskiego[24].

The need for the Latin Modern fonts arose because of the limitations of the original Computer Modern (CM) fonts.

TEX was originally developed with 7-bit characters internally. This meant that the input was basically ASCII code (although Knuth's own system accepted some extra characters in the low ASCII range). And TeX's fonts had a maximum of 128 characters each, basically the ASCII characters, some accents and some special characters, such as ß, æ, œ, Æ, Œ, some ligatures and some Greek capital letters. Accented letters had to be constructed by placing an accent above or below the unaccented letter.

There were two big problems with this:

1. The number of accents was limited, and so for some languages it was impossible to construct all characters that are in use in their alphabet.

2. It was not possible to properly hyphenate words that contain accented letters. TEX's hyphenation algorithm only works on letters[8], not on combinations of letters and accents. For English this isn't a problem, but as soon as users from other language groups started using TEX this became a serious problem.

To alleviate this Knuth changed the TEX implementation to accept 8-bit characters. But this did not immediately resolve the problem. One is that for the additional 128 byte values there was no standard encoding (i.e., correspondence between the number and an actual character). Or alternatively you can say there were just too many standards. The second problem was that TEX's

standard fonts (Computer Modern) still did not have accented characters, although now there was space for them. Similarly, the question is what encoding to use if the new 128 places in a font are used. These two are separate: The encoding of the input is chosen by the author of the document; the encoding in a font is chosen by the font designer. So these two become independent. In LaTeX there are two packages: `inputenc` for choosing the encoding for the input file[9] and `fontenc` for selecting which output encoding will be used. Modern LaTeX has UTF-8 as its default input encoding, and ASCII is a subset of this. So you only have to specify the input encoding if it is another one, for example:

```
\usepackage[latin9]{inputenc}
```

This will select the single-byte encoding for Western-European languages (Latin-9=ISO-8859-15) that also contains the Euro-sign.

The `fontenc` package will look for an appropriate font in the encoding(s) specified. You can specify more than one encoding; the last one will be the default used (this is important for hyphenation) but the information for the others will also be loaded, so that characters that need this can use the proper font. For example:

```
\usepackage[T3,T1]{fontenc}
```

For the whole story about these encodings, see The LaTeX Companion, Third Edition[45], I, chapter 9, pp. 647–776.

In 1990, at the Cork TUG Conference, a standard 8-bit encoding for TeX text fonts was developed, called T1; whereas the original (7-bit) Computer Modern text font encoding was called OT1. The T1 encoding supports accented letters of quite a number of languages that use a Latin-based alphabet, but the limit of 256 characters caused it to be far from sufficient for all Latin-based alphabets, for example characters for many African languages[55] and Vietnamese[56] are missing, let alone others like the Cyrillic alphabet, Hebrew, Arabic, Chinese, etc. Within the constraints of an 8-bit encoding this can only be solved by supplying many more font encodings.

In the meantime several other font encodings have been developed, for example TS1, a supplement for special characters not in T1 (including the € sign); the corresponding fonts are called Text Companion Fonts. Other encodings are T2A, T2B, T2C for the Cyrillic alphabet; T3 for the phonetic alphabet; T4 for several African languages that use a Latin-based alphabet; T5 for Vietnamese; LGR for Greek. See [44] for an exhaustive description, including font tables for the various encodings. In your local TeX installation, you can get the same information with the command '`texdoc encguide`'.[10]

A problem with the T1 encoding is that not all characters that it contains are supported by commercial fonts, like the original Adobe Type 1 fonts, although the missing characters are not widely used. Even worse, the characters from the TS1 encoding are not found in many commercial fonts either.

Newer TeX implementations, like XeTeX and LuaTeX, internally use Unicode. They also support Unicode-encoded fonts directly. This is called TU encoding in LaTeX. Documents that are to be processed by these engines should not use the `fontenc` package, but rather the `fontspec` package. See the section 'Using Unicode LaTeX' below. We will refer to the TeX implementations as *Unicode TeX* and the 8-bit TeX implementations as *classical TeX*. This includes PDFTeX.

Classical TeX is still restricted by its 8-bit heritage. In classical LaTeX in its current implementations internally non-ASCII characters are translated to a TeX command name, and later, when they are typeset with a specific font, these are substituted with the proper numerical value of the character in the particular encoding of the font (if available), or maybe it is constructed from different parts. Even constructs like `\'{a}` with the T1 font encoding (or another one that contains the character á) will eventually generate the single character á in the output file, whereas with the OT1 font encoding it will use the letter a with a separate accent above it.

Extensions of the original Computer Modern (CM) fonts that support the T1 font encoding and others were developed by Jörg Knappen and Norbert Schwarz between 1995 and 1997: the European Computer Modern (EC) fonts. The corresponding fonts that support the TS1 encoding are called *Text Companion (TC) Fonts*. There isn't a report about these, but there is the README file and two articles about the predecessor of these fonts, the DC fonts[37, 38, 39].

Knappen also created fonts for some 40 African languages that have a latin-based script, plus the European languages Maltese and Sami. These are called the FC fonts[36]. They are based on the Computer Modern fonts, and their naming is obtained by replacing `cm` with `fc`, e.g., `fcr10` corresponds to `cmr10`, with the new characters added in the upper 128 character positions. The font encoding is T4, and the lower 128 characters in this encoding are identical to those in the T1 encoding. The EC, TC and FC fonts are only available as MetaFont fonts (i.e., bitmaps).

The EC fonts are automatically selected when the `\usepackage[T1]{fontenc}` command is used in the document when DVI is produced. But see the next section for PDF. In the Unicode-based processors the Latin Modern fonts are used, see below.

Type 1 versions

With the advent of Postscript and later PDF as preferable document format, there arose a need for *scalable* fonts, for example in Type 1 format. The CM fonts as well as the EC fonts were MetaFont-based, and MetaFont generates bitmaps (`.pk` files). However bitmaps should have the resolution of the printer where the document is printed, or the screen where it is displayed. Even for printers with the same resolution, it may be necessary to fine-tune the fonts for the particular printer model, otherwise the output will be sub-optimal. This makes it difficult to interchange generated Postscript or PDF files. The only practical solution is to include the fonts in outline format, so that the rendering (conversion to pixels) is done by the printer itself or by the printer driver. For commercial fonts this is not a problem as these are distributed in outline format. But the Metafont-generated fonts were not. Therefore there was a quest to convert these also to outline fonts.

The CM fonts have been converted to Type 1 format (`.pfb` files) by Blue Sky Research and Y&Y, Inc. from MetaFont-generated bitmaps with commercial tracing software, with hand-crafted hints and corrections applied[2]. Originally these were commercial fonts: they had to be bought to use them. But later they have been donated to The American Mathematical Society (AMS), and their copyright was transferred to AMS. Nowadays they are distributed under a free License[1]. Sometimes they are referred to as the AMS fonts. In classical TEX these fonts are used when PDF is generated from a document with CM fonts (i.e., if font encoding OT1 is used), as can be seen in the `.log` file. It should include something like: `.../fonts/type1/public/amsfonts/cm/cmr10.pfb`.

So these AMS fonts cover the CM fonts, but not the EC fonts, which are usually preferred for languages that have accented letters, as explained above.

The EC and TC fonts together with some extra fonts from the Computer Modern Family (Computer Modern Concrete, Computer Modern Bright) were converted to Type 1 format by Vladimir Volovich[57] under the name `CM-Super`, using TEXtrace and postprocessing for optimizing and hinting. This is a fully automatic process and is described in detail in the article.

Fonts with the same design and size, but different encodings were combined together in one Type 1 font to save space. For example the fonts `ecrm1000` (T1), `tcrm1000` (TS1) and the Cyrillic fonts `larm1000` (T2A), `lbrm1000` (T2B), `lcrm1000` (T2C) are all derived from Computer Modern Roman 10pt, and the converted forms of these are all combined into a single Type 1 font `sfrm1000` (`sf` for super font). This is possible because a Type 1 font can contain more than 256 characters. Characters are identified by a *name*, a / followed by an identifier. For example `/A` indicates the letter A; `/braceleft` indicates { and `/eacute` indicates é, etc. To use the characters in a document, an encoding must be used, which is an array of 256 names. But multiple encodings can be used in a document that use the same Type 1 font (`.pfb`). The CM-Super font collection supports glyphs from the font encodings T1, TS1, T2A, T2B, T2C and X2 (another Cyrillic encoding).

When one of these font encodings is used in a document, the proper font (e.g. `sfrm1000`) is used with the chosen encoding applied. In fact classical LaTEX is still set up to use the EC fonts, but when PDF is generated there is a mapping that says that for example `ecrm1000` (European Computer Modern Roman 10pt) is to be mapped to `sfrm1000` with the T1 encoding applied. The log file will then contain something like `.../fonts/type1/public/cm-super/sfrm1000.pfb`.

Latin Modern

The Latin Modern (LM) fonts were developed by the Polish TeX Users Group (GUST) as an improved and extended version of the Computer Modern (CM) fonts. The project started in 2002, and the work was done by the e-foundry, i.e., Bogusław Jackowski and Janusz M. Nowackii. The work was financed by a number of TEX user groups, namely DANTE e.V., GUTenberg, NTG, TUG, CS TUG and GUST. The project was done with MetaType1, as described above. See the project page[11].

The developers wanted to generate a multilingual replacement of the Computer Modern fonts in outline format. Of course around the same time the CM-Super outline fonts were also develeoped as described above, with approximately the same purpose, but the Latin Modern project has a few additional requirements.

- Because the CM-Super fonts were generated by tracing bitmaps with TEXtrace, the the resulting outline fonts were not always of the best quality, although they are good. See the comparison of TEXtrace with MetaType1 in [48].
- Also the resulting fonts are not exactly reproducible from source code. A different tracing software or a newer version of this software could generate slightly different outlines.
- The automatic hinting was done by a commercial program Fontlab, and this is also an uncertain factor, as a new version or another program, like Fontforge, could generate different hinting.
- It is difficult to make reproducible corrections to the font.

So the project developers decided to take a route to generate the fonts from a MetaPost source, that contains all the information to exactly reproduce the outlines. And this is possible with MetaType1.

The MetaFont source is not derived from the Computer Modern MetaFont source as this would require many changes, as described above. Instead they are generated from the Type 1 AMS fonts with the script `PF2MT1`. These sources are then the basis to create source code for new characters, for example by combining existing letters with new accents. The fonts support several encodings, such as T1, TS1, QX (a variant of T1 for Slavonic languages), T5 (Vietnamese) L7x (Lithuanian) and a few special ones. Also some additional fonts, not in the Computer Modern collection were made, such as a bold typewriter fonts, and a condensed (narrow) typewriter font. More details can be found in [20, 21, 50, 51].

To use the Latin Modern fonts in classical LaTeX, include the following in the document preamble:

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
```

To use the LM fonts only in a shorter piece of text the following construction can be used (inside a group):

```
\fontencoding{T1}\fontfamily{lmr}\selectfont
And then the text in LM.
```

The `\fontencoding{T1}` can be omitted if the document already uses this encoding.

In the Unicode LaTeXs and in ConTeXt Latin Modern is the default font.

The LM fonts are available in both Type 1 and OpenType formats. In classical LaTeX Type 1 is used; whereas in the Unicode implementations OpenType is used. The Unicode implementations also support Type 1, but OpenType is the preferred format. The OpenType fonts can also be used in non-TeX software such as word processors. The CM-Super fonts, on the other hand, are only available in Type 1 format.

The Latin Modern family of fonts consists of 72 text fonts and 20 mathematics fonts. Each of the LM text fonts contains more than 700 glyphs. See '`texdoc lm-info`'[10] for detailed information about the fonts. They are almost the same as Computer Modern for those characters that are present in both, but not 100%. They have done their best to make sure that the character widths of the shared characters are the same, but the character shapes may be slightly different. So replacing CM with LM can change a document slightly. As a comparison here are two samples, one with Computer Modern, and the other one with Latin Modern. Close inspection reveals tiny differences. In particular the diaereses in the bold fonts are clearly different, but also the dots on *i* and *j* in the *italic* fonts, but you have to zoom in to see the difference. These are deliberate changes and they are described in [21].

Computer Modern
`\fontencoding{OT1}\fontfamily{cmr}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Latin Modern
`\fontencoding{T1}\fontfamily{lmr}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

The TeX Gyre fonts

As described in the Introduction above, the TeX Gyre fonts are essentially a reimplementation of the original standard Postscript fonts, based on the URW++ fonts. However, they contain many more glyphs, especially in the area of accented characters.

GUST had already attempted to make more Polish (polonized) versions of these fonts under the name *Quasi* or *Qfonts*. This was done under the GUST e-foundry by Bogusław Jackowski, Janusz M. Nowacki and Piotr Strzelczyk. However, after the success of the construction of Latin Modern with MetaType1, in 2006 Hans Hagen initiated the TeX Gyre project to provide a new set of such fonts[14]. This time it was done by Bogusław Jackowski, Janusz M. Nowacki, Piotr Strzelczyk and Marcin Woliński, and sponsored by CS TUG, DANTE e.V., ConTeXt Group, GUST, NTG, TUG India and TUG. The Vietnamese characters were added by Hàn Thế Thành.

A nice overview of how the fonts look can be seen at [46].

The TeX Gyre fonts were created by first converting the URW++ fonts (which were released under a permissive license) to MetaPost with `pf2mt1`[19]. The resulting MetaPost sources were used as the basis to construct new characters, similar to the process used for Latin Modern. From these sources the Type 1 and later OpenType fonts were generated. Detailed info can be found in [27, 33, 30, 15]. For a peek in the (past) development of TeX Gyre see [43].

The fonts are licensed under the GUST Font License[12], which is legally identical to the LaTeX Project Public License (LPPL).

Table 1 shows the correspondence between the original Postscript fonts and the TeX Gyre fonts. (For brevity I have left out the intermediate URW++ fonts.) The 'abr' columns are the codes that LaTeX uses in its font files. For the Postscript fonts this is 'p' followed by a 2 letter abbreviation of the font name; for the TeX Gyre fonts it is the same with 'p' replaced by 'q'. This makes them look a bit strange.

Table 1: Correspondence between Postscript and TEX Gyre fonts

Postscript	abr	TEX Gyre	abr
ITC Avant Garde Gothic	pag	Adventor	qag
ITC Bookman	pbk	Bonum	qbk
Courier	pcr	Cursor	qcr
New Century Schoolbook	pcs	Schola	qcs
Helvetica	phv	Heros	qhv
Palatino	ppl	Pagella	qpl
Times	ptm	Termes	qtm
ITC Zapf Chancery	pzc	Chorus	qzc

All the TEX Gyre fonts have Upright (Regular), Italic/Slanted, Bold and Bold Italic versions, except Chorus, which has only Medium Italic. Heros has also a condensed version, corresponding to Helvetica Narrow, also in the 4 variants. Thus we have 33 fonts.

Detailed information of each of these fonts can be obtained by the command '`texdoc ⟨abr⟩`', or '`texdoc ⟨abr⟩-info`', where ⟨*abr*⟩ is the entry in the fourth column of table 1.

Using TEX Gyre fonts in LaTEX

How we use these fonts in LaTEX depends on two things:

1. Do we use classical or Unicode LaTEX?
2. Do we want a whole document to use the font, or just a part of the document?

Using classical LaTEX

If we want a complete document in one of the TEX Gyre fonts in classical LaTEX, we can use a package to accomplish this. For example, to have a document use Termes as its main font, use:

```
\usepackage[T1]{fontenc}
\usepackage{tgtermes}
```

Note however, that this only changes the 'Roman' font family, i.e. the fonts selected by `\textrm` and its bold and italic variants. This family is indicated by the command `\rmfamily`. LaTEX does have two other font families, the Sans Serif family (indicated by `\sffamily`) and the Typewriter family (`\ttfamily`). For a detailed description see The LaTEX Companion, Third Edition[45], I, chapter 9, especially pp. 665–672, 688–691.

In table 2 you find the package name for each of the fonts, and in the third column it indicates which family will be changed by the package. For example, Adventor and Heros are Sans Serif fonts, and their packages therefore only change the `\sfdefault`. The main text will still use the default `\rmfamily`, which is CM or EC. The third column will be discussed later.

Table 2: Packages for TEX Gyre fonts

Name	package	what
Adventor	tgadventor	sf
Bonum	tgbonum	rm
Cursor	tgcursor	tt
Schola	tgschola	rm
Heros	tgheros	sf
Pagella	tgpagella	rm
Termes	tgtermes	rm
Chorus	tgchorus	rm

Table 3: Packages options for TEX Gyre fonts

option	meaning
`oldspacing`	use wider character spacing[a]
`scale`	scale factor[b]
`matchlowercase`	scale to match lower case letters
`matchuppercase`	scale to match upper case letters
`condensed`	Only for Heros: use the narrow font

[a] This does nothing for Adventor, Cursor and Heros.
[b] Default for Adventor: 0.9; Bonum and Heros: 0.95; Chorus: 1.1; others 1.0.

If you want your whole document to be in the selected font (Adventor or Heros), add the following command:

```
\renewcommand\familydefault{\sfdefault}
```

On the other hand, the fonts indicated with 'rm' do not change the `\sffamily` and `\ttfamily`, so these stay in CM or EC. But you can combine the different packages. So if you want your main font to be Bonum, your Sans Serif to be Adventor, and your Typewriter font to be Cursor, use `\usepackage{tgbonum,tgadventor,tgcursor}`.

In the unlikely case that you would want your document completely in the typewriter font Cursor, you would then have to use

```
\renewcommand\familydefault{\ttdefault}
```

The packages have a number of options which you can specify with
`\usepackage[⟨options⟩]{⟨package⟩}`.
See table 3 for an overview.

The fonts sometimes have different sizes, which makes them not combine well. You can manually scale them with the `scale` option, or automatically have them scaled such that the lowercase letters become as tall as the other fonts (`matchlowercase` option), or the upper-

case letters (`matchuppercase`). Only one of these makes sense for a particular font.

Using TEX Gyre in a part of the text

To use one of these fonts in just a part of the text in a document that uses a different font, we cannot use the packages mentioned above as these change the font for the whole document. However we can use some lower-level commands, in particular the commands `\fontfamily` and `\selectfont`. For example

```
\begingroup
\fontencoding{T1}\fontfamily{qbk}\selectfont
This is text in the \TeX\ Gyre Bonum font.\\
\textit{And this is the italic version.}\\
\textbf{And Bold and \textit{Bold Italic}}.
\endgroup
```

selects the Bonum font. We use the 3-letter code from the corresponding 'abr' in table 1. We can leave out the `\fontencoding{T1}` part if the document already uses this encoding or if we don't care. Or maybe we want to use a different font encoding because our language requires this.

As shown here we should use this construction inside a TEX group with `\begingroup` ... `\endgroup`, between braces or inside an environment, so that the normal font will be resumed after this piece of text. Of course it is recommended to make a macro if you need it more than once.

The result of the above is

This is text in the TEX Gyre Bonum font.
And this is the italic version.
And Bold and *Bold Italic*.

Because this selects a font family, it also supports bold and italic.

However, this text looks too big compared with the surrounding text. We can compensate this by adding a `\fontsize` command, for example `\fontsize{9}{12}` seems to work well in this example. Here is the same text with this added.

This is text in the TEX Gyre Bonum font.
And this is the italic version.
And Bold and *Bold Italic*.

Unfortunately we cannot use the nice package options here.

Using Unicode LaTEX

In the Unicode LaTEX engines the font handling is very different. The internal coding of characters in these engines is Unicode, and by default the fonts are also accessed in Unicode. Moreover they access the standard fonts as used by the operating system, which nowadays usually is in OpenType format.

So we should not use the `fontenc` package and the specific packages for the TEX Gyre fonts as described above (although sometimes it may still work). Instead we use the `fontspec` package, with the command `\setmainfont`, for example

```
\usepackage{fontspec}
\setmainfont{TeX Gyre Bonum}
```

There are also commands to set the Sans Serif and Typewriter fonts. So to set the main font to be Bonum, the Sans Serif to Adventor, and the Typewriter font to Cursor as we did above, now we use

```
\usepackage{fontspec}
\setmainfont{TeX Gyre Bonum}
\setsansfont{TeX Gyre Adventor}
\setmonofont{TeX Gyre Cursor}
```

As you see, we use the normal font name, as it is known in the operating system, and stored in the font (`.otf`) file. These engines don't need all the special LaTEX support files that classical LaTEX needs; they get all the required information from the font itself.

These commands also automatically select the corresponding bold and italic versions to be used with `\textbf` and `\textit` if these are available. This is the case for all TEX Gyre fonts except Chorus, which comes only in a Medium Italic version. These commands also allow for additional options to be specified, for example to scale the font or select special features. See the `fontspec` documentation, accessible with '`texdoc fontspec`'.[10]

When the `\set`*`(xxxx)`*`font` commands are used in the preamble, they set the default fonts for the whole document. However, we can also use them inside a group to change the font for the text inside that group. Of course the `\usepackage{fontspec}` must still be given in the preamble.

If you want the document to be processable by both PDFTEX and the Unicode engines, you could use the following construction that dynamically checks what to use.

```
\usepackage{iftex}
\ifpdftex
  \usepackage[T1]{fontenc}
  \usepackage{tgbonum}
\else
  \usepackage{fontspec}
  \setmainfont{TeX Gyre Bonum}
\fi
```

Special Caveat for XeLaTEX

There is a difference between XeLaTEX and LuaLaTEX in finding the fonts. LuaLaTEX will find the fonts specified in `\setmainfont`, `\setsansfont`, and `\setmonofont` both in the font directories of the operating system and in the TEX distribution. However, XeLaTEX sees only

the fonts in the system font directories, not those in the TEX distribution[13]. Without further action this will lead to an error message in the above examples, as the TEX Gyre fonts usually are not installed as system fonts.

There are a few ways to correct this.

1. One way would be to copy these fonts to the system font directories. This is not recommended, as updates to these fonts in future TEX distributions would not be propagated to the system fonts, and it duplicates the space requirements. And it will not work if you want to share your document.

2. Another way is to specify a font *file name* instead of a font name, e.g., `\setmainfont{texgyrebonum-regular.otf}`. The file name is the lowercase version of the font name with spaces removed, followed by `-regular.otf`. However, this only sets the regular/upright/roman version of the font, and does not affect the bold and italic versions. These have to be specified separately, e.g.,

```
\setmainfont{texgyrebonum}[
    Extension = .otf,
    UprightFont = *-regular,
    BoldFont = *-bold,
    ItalicFont = *-italic,
    BoldItalicFont = *-bolditalic
]
```

   The * will be replaced by the first argument. This works for all the TEX Gyre fonts, except TEX Gyre Chorus, where you would have to use the following:

```
\setmainfont{texgyrechorus-mediumitalic.otf}
```

3. A more general way is to define a `.fontspec` file, e.g., create a file `TeXGyreBonum.fontspec`

```
\defaultfontfeatures[TeX Gyre Bonum]
{
  Extension = .otf,
  UprightFont = texgyrebonum-regular,
  BoldFont = texgyrebonum-bold,
  ItalicFont = texgyrebonum-italic,
  BoldItalicFont = texgyrebonum-bolditalic,
}
```

   Here the file name is the font name without spaces, and the font file names inside contain this name in lowercase. You cannot use * in this file for the file names. Note that this is specific for the TEX Gyre fonts. Other fonts may have different conventions for file names. If you put the `.fontspec` file in a place where TEX searches for TEX files[14], you can then use `\setmainfont{TeX Gyre Bonum}` without problems. I created `.fontspec` files for all TEX Gyre fonts in my local latex directory, i.e., `~/Library/texmf/tex/latex` on MacOS.[15]
   The `TeXGyreChorus.fontspec` file for TEX Gyre Chorus is simpler, as it has only a Medium Italic font. I have chosen to use this for both Upright and Italic:

```
\defaultfontfeatures[TeX Gyre Chorus]
{
  Extension = .otf,
  UprightFont = texgyrechorus-mediumitalic,
  ItalicFont = texgyrechorus-mediumitalic,
}
```

   But if you want to share your document, you must also share the required `.fontspec` files.

Use of the TEX Gyre fonts in ConTEXt

The command to select the default font for a document in ConTEXt is `\setupbodyfont`. We can use the lowercase version of the TEX Gyre font name (without TEX Gyre), e.g., 'bonum'[16] and add some options, like the point size. For example:

```
\setupbodyfont[bonum,12pt]
\starttext
The rápido fox jumps über den schönen cão.\\
{\bf The rápido fox jumps über den schönen cão.}\\
{\it The rápido fox jumps über den schönen cão.}
\stoptext
```

This will generate about the same output as the Bonum sample below.

However, there is a significant difference between `\setupbodyfont` and the corresponding LaTEX packages. In ConTEXt, when you specify a Sans Serif or Typewriter fonts, the whole document will be in these fonts, so it does not only change the Roman font.

On the other hand, if you specify a Serif font, like 'bonum' in the above example, the Sans Serif and Typewriter fonts will be taken from Latin Modern, as Bonum does not have these.

To get the combination of Bonum, Adventor and Cursor, as in the LaTEX examples above, you would have to define a *font family*:

```
\definefontfamily[myfont][rm][TeX Gyre Bonum]
\definefontfamily[myfont][sans][TeX Gyre Adventor]
\definefontfamily[myfont][mono][TeX Gyre Cursor]
                                        [rscale=1.4]

\setupbodyfont[myfont,10pt]
\starttext
The rápido fox jumps über den schönen cão.\\
{\bf The rápido fox jumps über den schönen cão.}\\
{\it The rápido fox jumps über den schönen cão.}\\
```

```
% Sans Serif = Adventor
{\ss The rápido fox jumps über den schönen cão.}\\
% Typewriter = Cursor
{\tt The rápido fox jumps über den schönen cão.}
\stoptext
```

Note that we use the full font name. We also have added an option to scale the Typewriter font Cursor, just for demonstration purposes.

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über schönen cão.

To change the font mid-way in a document (for example in a group), you use the command `\switchtobodyfont` or `\definedfont`:

```
\switchtobodyfont[adventor]
This text will be in Adventor

\definedfont[name:texgyreheroscn-regular*default
                                        at 12 pt]
This text will be in Heros Condensed at 12pt.

\definedfont[file:texgyrepagella-regular.otf*default
                                          at 24pt]
This will be in Pagella at 24pt.
{\bf But this will still be Adventor Bold.}
```

This text will be in Adventor This text will be in Heros Condensed at 12pt. This will be in Pagella at 24pt. **But this will still be Adventor Bold.**

Note that `\definedfont` can use a font name or a font file name. In this case we specified a Regular font, so this does not switch the Bold and Italic fonts. After the * a feature set can be specified. In this case the default feature set, which includes ligatures and kerning amongst others. Without such a specification these are not used. You can find a list of some of the features in Willi Egger's tutorial about fonts in ConTEXt[7]. This document also tells how to find the features that a font supports, and how to define your own feature sets.

The font names and file names can be found with the following command:

```
mtxrun --script fonts --list --pattern=texgyre* --all
```

More information can be found in [7]. Note: there is an error in this document: it uses 'scholas' instead of 'schola'.

Samples of the TEX Gyre fonts

Adventor
`\fontencoding{T1}\fontfamily{qag}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Bonum
`\fontencoding{T1}\fontfamily{qbk}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Cursor
`\fontencoding{T1}\fontfamily{qcr}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Schola
`\fontencoding{T1}\fontfamily{qcs}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Heros
`\fontencoding{T1}\fontfamily{qhv}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Heros Cn
`\fontencoding{T1}\fontfamily{qhvc}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Pagella
`\fontencoding{T1}\fontfamily{qpl}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Termes
`\fontencoding{T1}\fontfamily{qtm}`

The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.
The rápido fox jumps über den schönen cão.

Chorus
`\fontencoding{T1}\fontfamily{qzc}`

The rápido fox jumps über den schönen cão.

TEX Gyre Math fonts

The GUST e-foundry also developed Math fonts to fit with the following TEX Gyre fonts: Bonum, Pagella, Schola, Termes, as well as Latin Modern. These are OpenType fonts that support Unicode Math. Therefore, they can only be used with the Unicode engines, including ConTEXt (and also with other non-TEX software).

The TEX Gyre Math family also includes TEX Gyre DejaVu Math[17] which does not have a corresponding text font in the TEX Gyre family.

It is beyond the scope of this article to discuss these fonts. More information can be found in [32, 5]. Examples of these fonts can be found at [6]. And also in The LaTEX Companion, Third Edition[45], II, chapter 10, especially pp. 269–270, 273, 276, 281, 285.

ConTEXt also has some additional Math Companion fonts, such as `TeXGyreTermesMath-Companion-Regular.otf`. These contain extensions and corrections to the corresponding 'normal' math font. These are applied automatically by ConTEXt, but they would be difficult to use in LaTEX. These companion fonts are also available for some other OpenType math fonts (Latin Modern, RSFS and XITS). It is beyond the scope of this article to discuss these. The fonts are available on CTAN[18], in TeXlive in the directory `texmf-dist/fonts/opentype/public/context`, and in the contextgarden Github repository[19].

Acknowledgements

I want to thank Frans Goddijn, Hans Hagen, Taco Hoekwater and Bogusław Jackowski for their feedback and recommendations.

Bibliography

NOTE: *This bibliography contains more than what has been cited in the text. It tries to be a reasonably comprehensive bibliography about the TEX Gyre fonts and related matters.*
All URLs have been accessed in March/April 2025.

[1] AMS. Computer Modern and AMSFonts in Type 1 (PostScript) form. Web page, 1997. `https://www.ams.org/publications/type1-fonts`.

[2] Nelson H. F. Beebe and Berthold Horn. Blue Sky Research and Computer Modern fonts. Web page, 2006. `http://www.math.utah.edu/%7Ebeebe/fonts/bluesky.html`.

[3] Amit Raj Dhawan. Macros to change text & math fonts in TEX: 19 beautiful variants. *CTAN, August*, 2009.

[4] GUST e-foundry. TEX Gyre Termes, 2012. `https://www.gust.org.pl/projects/e-foundry/tex-gyre/termes`.

[5] GUST e-foundry. The TEX Gyre math fonts. Web page, 2014. `https://www.gust.org.pl/projects/e-foundry/tex-gyre/math`.

[6] GUST e-foundry. GUST e-foundry math fonts. Web page, 2015. `https://www.gust.org.pl/projects/e-foundry/math/gust_e-foundry-math_fonts-presentation.pdf`.

[7] Willi Egger. Handling fonts in ConTEXt. Slides, 2021. 15th ConTEXt Meeting, Bassenge, Belgium. `https://meeting.contextgarden.net/2021/talks/2021-09-23/typescripts-presentation.pdf`.

[8] Michael Everson. The alphabets of europe. Web page, 2015. `https://www.evertype.com/alphabets/`.

[9] GUST. *TEX Gyre Collection*, 2019. `https://www.gust.org.pl/projects/e-foundry/tex-gyre`.

[10] Hans Hagen. Do we need a font system in TEX? *TUGboat*, 29(1):28–33, 2007. Proceedings of the 17th European TEX Conference 2007, Bachotek, Poland. `https://tug.org/TUGboat/tb29-1/tb91hagen.pdf`.

[11] Hans Hagen. Plain TEX and OpenType. *TUGboat*, 30(2):241–242, 2009. `https://www.tug.org/TUGboat/tb30-2/tb95hagen-opentype.pdf`.

[12] Hans Hagen, Jerzy Ludwichowski, and Volker Schaa. The new font project: TEX Gyre. *MAPS*, 34(Najaar 2006):47, 2006.

[13] Hans Hagen, Jerzy B Ludwichowski, and Volker RW Schaa. The new font project: TEX Gyre. *TUGBoat*, 27(2):230, 2005.

[14] Hans Hagen, Jerzy B. Ludwichowski, and Volker RW Schaa. The new font project: TEX Gyre. *TUGboat*, 27(2):250–253, 2006. Proceedings of the TUG 2006 Conference,

Marrakech, Morocco. https://www.tug.org/TUGboat/Articles/tb27-2/tb87hagen-gyre.pdf.

[15] Jim Hefferon. A first look at the TEX Gyre fonts. *TUGboat*, 30(2):236, 2009. https://www.tug.org/TUGboat/tb30-2/tb95heff-gyre.pdf.

[16] John D. Hobby and the MetaPost development team. MetaPost – *A User's Manual*. TUG, 2024. https://www.tug.org/docs/metapost/mpman.pdf.

[17] Taco Hoekwater. Generating Type 1 fonts from MetaFont sources. *TUGboat*, 19(3), 1998. Proceedings of the 1998 Annual TUG Meeting, Toruń, Poland. https://www.tug.org/TUGboat/tb19-3/hoek1.pdf.

[18] Taco Hoekwater. Opentype fonts in LuaTEX. *TUGboat*, 29(1):34–35, 2007. Proceedings of the 17th European TEX Conference 2007, Bachotek, Poland. https://tug.org/TUGboat/tb29-1/tb91hoekwater.pdf.

[19] Bogusław Jackowski and Jerzy B. Ludwichowski Janusz M. Nowacki. Recent advances in LUG's font projects, 2007. DANTE 2007, Münster, Germany. https://www.gust.org.pl/projects/e-foundry/tex-gyre/afp05.pdf.

[20] Bogusław Jackowski and Janusz M. Nowacki. Latin Modern: Enhancing computer modern with accents, accents, accents. *TUGboat*, 24(1):64–74, 2003. Proceedings of the TUG 2003 Conference, Hawaii, USA. https://www.tug.org/TUGboat/Articles/tb24-1/jackowski.pdf.

[21] Bogusław Jackowski and Janusz M. Nowacki. Latin Modern: How less means more. *TUGboat*, 27(0):171–178, 2006. Proceedings of the 15th European TEX Conference 2005, Pont-à-Mousson, France. https://www.tug.org/TUGboat/Articles/tb27-0/jackowski.pdf.

[22] Bogusław Jackowski, Janusz. M. Nowacki, et al. TEX Gyre Adventor, the technical documentation of the font. Technical report, The TEX Gyre Project, 2006. https://www.gust.org.pl/projects/e-foundry/tex-gyre/adventor/qag-info.pdf.

[23] Bogusław Jackowski, Janusz M. Nowacki, and Jerzy B. Ludwichowski. What is the TEX Gyre project about? Slides, 11 2006. Proceedings of the TUG 2006 Conference, Marrakech, Morocco. https://www.gust.org.pl/projects/e-foundry/tex-gyre/nfp11.pdf.

[24] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. Antykwa Półtawskiego: a parameterized outline font. *MAPS*, 25:86–101, 2000. This is a copy of a paper in the Proceedings of EuroTeX'99 Paperless TEX, 1999, pp. 117–141. https://www.ntg.nl/maps/25/13.pdf.

[25] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. METATYPE1: generating Postscript Type 1 fonts using Metapost, 2001. BachoTeX 2001. https://ftp.gust.org.pl/pub/GUST/contrib/BachoTeX2001/B01-jns1.pdf.

[26] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. MetaType1: a METAPOST-based engine for generating Type 1 fonts. *MAPS*, 26:11–119, 2001. https://www.ntg.nl/maps/26/15.pdf.

[27] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. Programming PS Type 1 fonts using MetaType1: Auditing, enhancing, creating. *TUGboat*, 24(3):575–581, 2003. https://www.tug.org/TUGboat/Articles/tb24-3/jackowski.pdf.

[28] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. TEX Gyre Adventor: A universal sans-serif font. *EuroTeX Proceedings*, pages 88–97, 2009. https://www.gust.org.pl/projects/e-foundry/tex-gyre/adventor.

[29] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. TEX Gyre Bonum: An elegant serif font. *EuroTeX Proceedings*, pages 112–121, 2010. https://www.gust.org.pl/projects/e-foundry/tex-gyre/bonum.

[30] Bogusław Jackowski, Piotr Pianowski, and Piotr Strzelczyk. TEX Gyre text fonts revisited. *TUGboat*, 39(3):212–216, 2018. https://www.tug.org/TUGboat/tb39-3/tb123jackowski-gyre.pdf.

[31] Bogusław Jackowski, Piotr Pianowski, and Piotr Strzelczyk. TEX Gyre text fonts revisited. *MAPS*, 2018. https://www.ntg.nl/maps/48/05.pdf.

[32] Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski. Progress in the TEX Gyre Math Font project. Slides, 2013. Bachotek, Poland. https://www.gust.org.pl/projects/e-foundry/math/tgm-state02.pdf.

[33] Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski. Gust e-foundry font projects. *Zpravodaj Československého sdružení uživatelů TeXu*, 26(1):7–46, 2016. https://dml.cz/dmlcz/150240, doi:10.5300/2016-1-4/7.

[34] Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski. Parametric math symbol fonts. *TUGboat*, 38(2):208–211, 2017. https://www.tug.org/TUGboat/tb38-2/tb119jackowski.pdf.

[35] Richard J. Kinch. MetaFog: Converting MetaFont shapes to contours. *TUGboat*,

16(3):233–243, 1995. Proceedings of the 1995 Annual TUG Meeting. https://www.tug.org/tugboat/tb16-3/tb48kinc.pdf.

[36] Jörg Knappen. Fonts for Africa: The fc fonts. *TUGboat*, 14(2):104–106, 1993. https://www.tug.org/TUGboat/tb14-2/tb39knappen.pdf.

[37] Jörg Knappen. Release 1.2 of the dc-fonts: Improvements to the European letters and first release of the text companion symbols. *TUGboat*, 16(4):381–387, 1995. https://www.tug.org/TUGboat/tb16-4/tb49knap.pdf.

[38] Jörg Knappen. The dc fonts 1.3: Move towards stability and completeness. *TUGboat*, 17(2):99–387, 1996. TUG 1996 Proceedings, Dubna, Russia. https://www.tug.org/TUGboat/tb17-2/tb51knap.pdf.

[39] Jörg Knappen and Norbert Schwarz. European Computer Modern fonts and Text Companion fonts, 1997. README file. http://mirrors.ctan.org/fonts/ec/src/00readme.txt.

[40] Donald E. Knuth. *The TEXbook*. Addison-Wesley, 1986.

[41] Jerzy Ludwichowski and Karl Berry. The GUST font license: An application of the LaTEX Project Public License. *TUGboat*, 29(1):66–67, 2008. Proceedings of the 17th European TEX Conference 2007, Bachotek, Poland.

[42] Jerzy B. Ludwichowski. Licensing of the TEX Gyre family of fonts, 2009. EuroTEX 2009, The Hague, The Netherlands. https://www.ntg.nl/bijeen/44/slides/jerzy-slides.pdf.

[43] Jerzy B. Ludwichowski, Bogusław Jackowski, and Janusz M. Nowacki. Five years after: Report on international TEX font projects. *TUGboat*, 29(1):25–26, 2008. Proceedings of the 17th European TEX Conference 2007, Bachotek, Poland. https://www.tug.org/TUGboat/tb29-1/tb91ludwichowski-fonts.pdf.

[44] Frank Mittelbach, Robin Fairbairns, and Werner Lemberg. *LaTEX font encodings*. LaTEX Project Team, 2016. https://www.latex-project.org/help/documentation/encguide.pdf.

[45] Frank Mittelbach and Ulrike Fischer. *The LaTeX Companion, Third Edition*. Addison-Wesley, 2023.

[46] Janusz M. Nowacki. Polish computer fonts. Web page. The TEX Gyre fonts in their beauty. https://jmn.pl/en/tex-gyre/.

[47] Karel Píška. A conversion of public indic fonts from MetaFont into Type 1 format with TEXtrace. *TUGboat*, 23(1):70–73, 2002. Proceedings of the 2002 Annual TUG Meeting. https://www.tug.org/TUGboat/tb23-1/piska.pdf.

[48] Karel Píška. Creating Type 1 fonts from MetaFont sources: Comparison of tools, techniques and results. In *Preprints for the 2004 TUG Annual Meeting, Xanthi, Greece*, pages 54–64, 2004. https://www.tug.org/TUGboat/tb25-0/piska.pdf.

[49] Karel Píška. Comments and suggestions about the Latin Modern fonts. *TUGboat*, 29(1):57–65, 2008. https://www.tug.org/TUGboat/tb29-1/tb91piska-lm.pdf.

[50] Will Robertson. An exploration of the Latin Modern fonts. *The PracTEX Journal*, 2006(1), 2006. https://www.tug.org/pracjourn/2006-1/robertson/robertson.pdf.

[51] Will Robertson. An exploration of the Latin Modern fonts. *TUGboat*, 28(2):177–180, 2007. This is a slightly updated reprint of [50]. https://www.tug.org/TUGboat/Articles/tb28-2/tb89robertson.pdf.

[52] Peter Selinger. potrace, 2019. http://potrace.sourceforge.net.

[53] Péter Szabó. Conversion of TEX fonts into Type 1 format. *MAPS*, 26:192–206, 2001. Proceedings of the EuroTEX 2001 conference, Kerkrade, the Netherlands. https://www.ntg.nl/maps/26/26.pdf.

[54] Péter Szabó. TEXtrace, 2002. TEXtrace is a collection of scripts for UNIX that convert any TeX font into a Type1 .pfb outline font immediately suitable for use with dvips, pdftex, acroread (and any many other programs). https://sourceforge.net/projects/textrace/.

[55] Conrad Taylor. Typesetting african languages. Technical report, Conradiator, 2000. A personal investigation by Conrad Taylor into the problems of typesetting African languages, with a focus on those which use an extended latin character set beyond the scope of normal fonts. This was written in 2000 and does not reflect the current situation where Unicode and OpenType offer solutions, but is contributed for historical reference. https://ia801305.us.archive.org/12/items/TypesettingAfricanLanguages/Afrolingua_full.pdf.

[56] Hàn Thế Thành. Making type1 fonts for Vietnamese. In *Preprints for the 24th Annual TUG Meeting, Waikoloa, Hawaii*, pages 1001–1007. TUG, 2003. https://tug.org/tug2003/preprints/Han/han.pdf.

[57] Vladimir Volovich. CM-Super: Automatic creation of efficient Type 1 fonts from MetaFont fonts. *TUGBoat*, 24(1):75–78, 2003. https://www.tug.org/TUGboat/Articles/tb24-1/volovich.pdf.

[58] Martin Weber. Autotrace, 2002. https://autotrace.sourceforge.net/.

Notes

1. Later versions of *Ghostscript* have a dual license, one for commercial use and one under a GPL-like license. These are slightly different versions.
2. Now part of Monotype.
3. `https://www.gust.org.pl/projects/e-foundry`
4. To be more precise: EPS files.
5. https://ctan.org/tex-archive/fonts/utilities/metatype1
6. `MetaFog` is part of Richard J. Kinch's commercial TEX distribution TrueTEX.
7. `https://www.gust.org.pl/bachotex/2024-pl/presentations/rkubiak-2-2024.pdf`
8. For the exact definition see The TEXbook[40], Appendix H.
9. Note that this must be specified before any non-ASCII code is used in the file.
10. *texdoc* is a 'command line' program to access the TEX documentation. If you are not familiar with using the command line, your TEX IDE probably has a way to access this documentation. For example in TeXShop, there is the menu item `Help > Show Help for Package`. And if you can't find a way to access the documentation on you local system, or if your local system does not have the documentation installed, you can use `https://texdoc.org`.
11. `https://www.gust.org.pl/projects/e-foundry/latin-modern/index_html`
12. `https://www.gust.org.pl/projects/e-foundry/licenses`
13. At least on MacOS. I don't know for sure if this is also the case on Linux and Windows.
14. Not in TEX's font directories
15. This can be found with the command
`kpsewhich -expand-var='$TEXMFHOME/tex/latex'`
Another appropriate choice would be the path returned by the command
`kpsewhich -expand-var='$TEXMFLOCAL/tex/latex'`
16. In ConTEXt also the original Postscript font name (avantgarde, bookman, courier, schoolbook, helvetica, palatino, times, chancery) can be used and will be replaced by the TEX Gyre name.
17. `https://www.gust.org.pl/projects/e-foundry/tex-gyre-dejavu-math`
18. `https://ctan.org/tex-archive/macros/context/base/fonts`
19. `https://github.com/contextgarden/context-distribution-fonts/tree/main/fonts/data/cms/companion`

Pieter van Oostrum
Leidsche Rijn
Utrecht
pieter@vanoostrum.org