

# MATH FONTS: notes from the trenches

Den Haag, August 31<sup>st</sup>–September 4<sup>th</sup>, 2009

*Bogusław Jackowski, Jerzy B. Ludwichowski, Piotr Strzelczyk*

# Math typesetting

In order to typeset math, one needs a math font and a typesetting engine that is able to make use of the information contained in that font.

Until recently, the only engine that could be used for this purpose (in serious applications) was  $\text{T}_{\text{E}}\text{X}$  with fonts consisting of TFM metrics and glyphs supplied either as PK bitmaps or PostScript Type 1 fonts (claimed by Adobe as obsolete since a few years).

Recently, however, another fairly powerful engine was released: MS Office 2007, using a modern OpenType font with math info (Cambria).

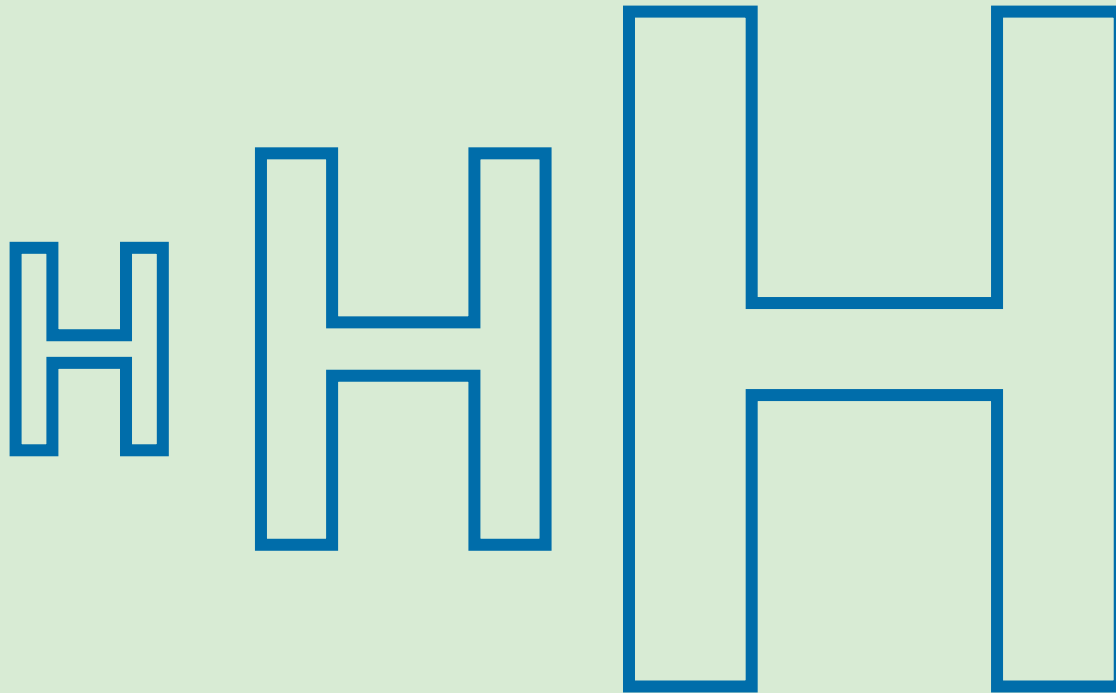
# Math typesetting: problems of today

Despite the similarities of  $\text{T}_{\text{E}}\text{X}$  and OpenType math (the latter is based on the  $\text{T}_{\text{E}}\text{X}$  concept) there are significant differences which do not allow for automated one-to-one transitions between both “worlds”.

The official documentation of math for OpenType fonts is rather poor: “confidential”, that is, unseen by many users, not many applications do understand OpenType math data (more about that later). With  $\text{T}_{\text{E}}\text{X}$  the situation is better, but far from delighting (consider, e.g., the number of math fonts for  $\text{T}_{\text{E}}\text{X}$ ). It should be emphasized, however, that thanks to the efforts of George Williams, Jonathan Kew, Ulrik Vieth, Hans Hagen, Taco Hoekwater and others, the situation recently has significantly improved.

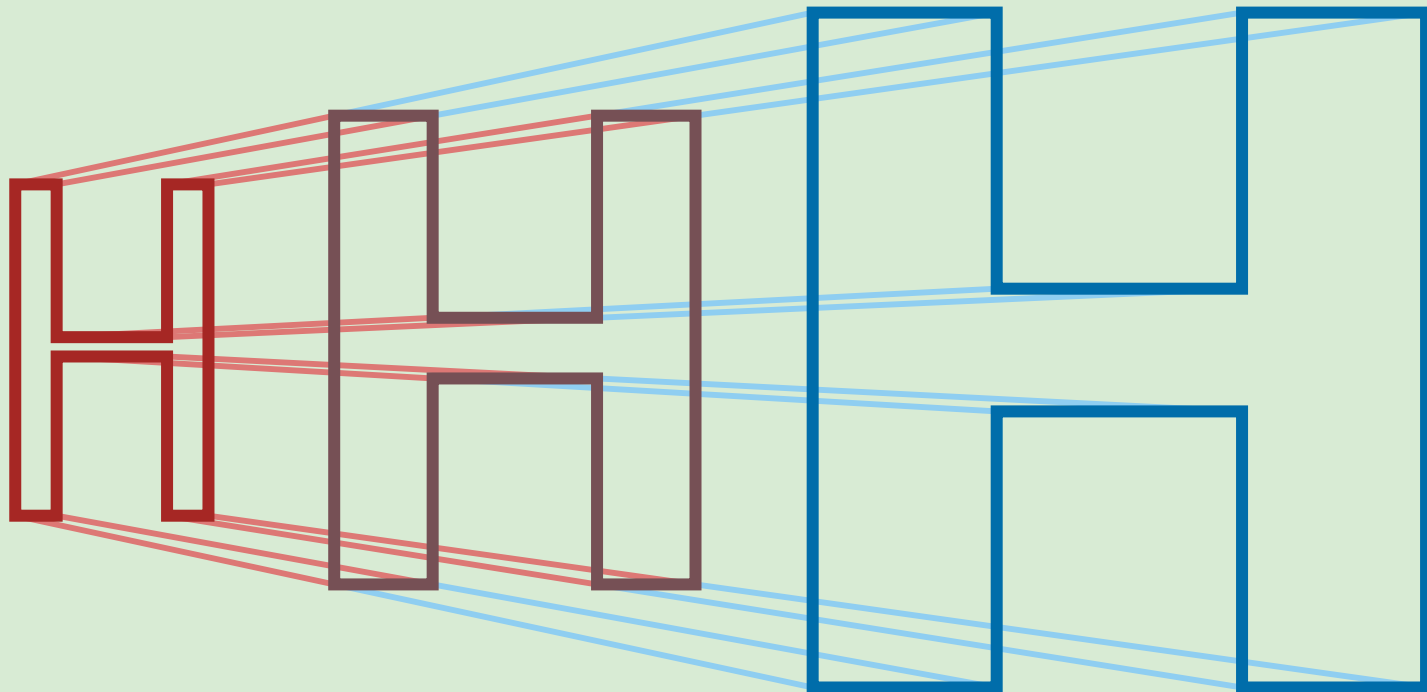
# Math typesetting: problems of today – a digression

A simple operation, **scaling**,  
commonly supported by existing font rasterizers



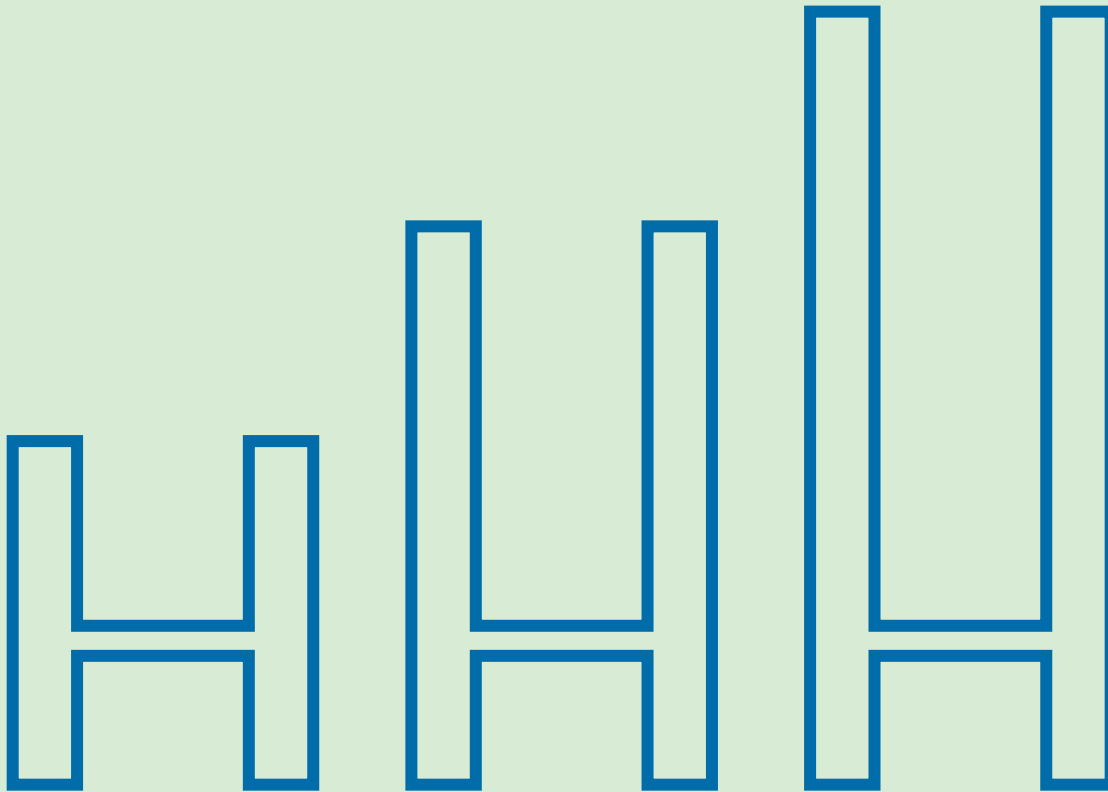
# Math typesetting: problems of today – a digression

Another relatively simple operation, **interpolation**, used to be supported by Adobe Multiple Master engines



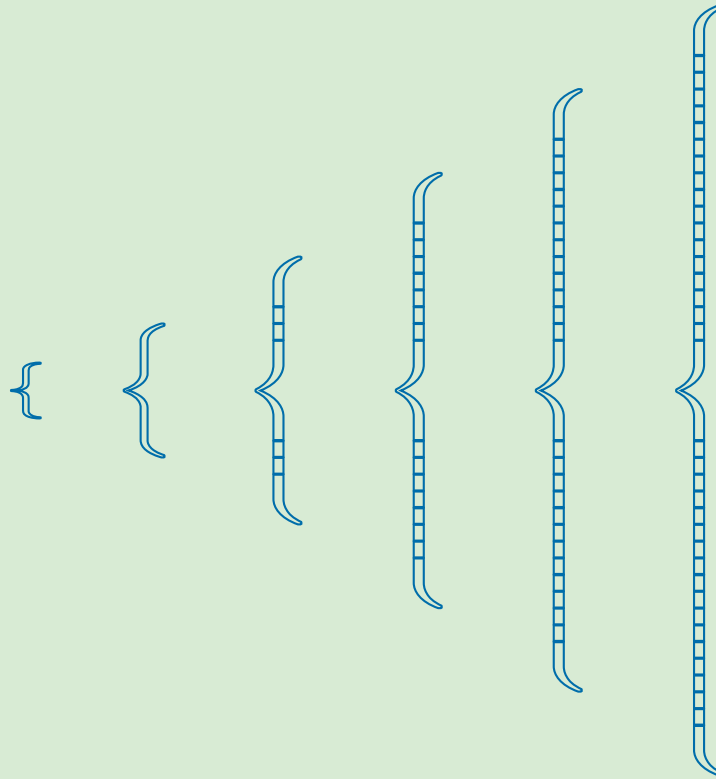
# Math typesetting: problems of today – a digression

Yet another simple operation, **translation**  
of **selected nodes**, unavailable in existing rasterizers



# Math typesetting: problems of today – a digression

Fairly complex operations, **glyph replacement** and **glyph assembling**, stemming from Gutenberg's ideas, available in  $\text{T}_\text{E}\text{X}$ , recently also in rasterizers of OTF fonts



## Math typesetting: problems of today – cont.

There are only a few OpenType fonts with math, actually just two: Cambria (commercial, Microsoft) and Asana (free, Apostolos Syropoulos).

Available math testing tools ( $\text{\LaTeX}$ ,  $\text{\luaTeX}$ , MS Office 2007, FontForge) are not fully proven and that is very painful.

The same applies for math OpenType fonts creation. In principle, only FontForge is useable. The commercial FontLab and free (but not open source) Adobe Font Development Kit for OpenType do not allow for adding math to OpenType fonts.



## Math typesetting: problems of today – cont.

An absolute must: assembling/disassembling – until recently not available.

Many tools seemingly suitable for those tasks cannot, in practice, be used; e.g., Microsoft's `ttfdump`, `ttoasm`, `ttodasm` (work “partly” – do not “understand” math, are cumbersome with respect to handling, unmaintained since 2002) or the free TTX (also works partly, does not “understand” math, not maintained regularly – last update in May 2008).

# Math typesetting: problems of today – résumé

What's left is FontForge (last update – June, 2009):

- it can be used in batch mode or even as a Python module,
- it outputs and inputs fonts as text files (SFD – Spline Font Database),
- it understands the AFDKO FEA format (i.e., it accepts the AFDKO files containing the descriptions of OpenType font features),
- there exists sfddiff, a program for semantic comparison of SFD files (thus really OpenType font files).

## Preparing the “attack”

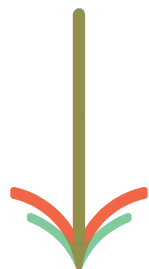
In order to concentrate fully on the math aspects, however, we have to freeze for some time (a few years?) the work on the Latin Modern and T<sub>E</sub>X Gyre collections. Hence copious changes introduced during this year.

One of the very important changes in LMs is the update of shapes according to a series of D. E. Knuth recent corrections; as concerns TGs, one cannot underestimate the importance of the fact that they are now legally available under the LPPL (GFL) licence.

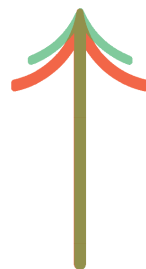
We expect to release the versions 2.xxx of the Latin Modern and T<sub>E</sub>X Gyre fonts soon after the EuroT<sub>E</sub>X 2009 meeting.

CM BITMAP FONT

LM OUTLINE FONT (OLD)



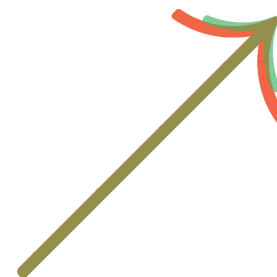
lmsy10+csc+tt  
cmsy10+csc+tt



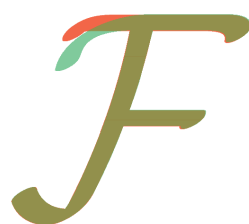
lmsy10+csc+tt  
cmsy10+csc+tt



lmsy10  
cmsy10



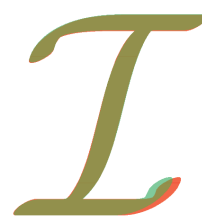
lmsy10  
cmsy10



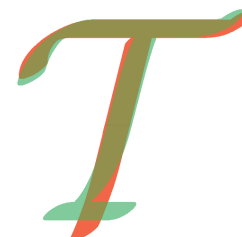
lmsy10  
cmsy10



lmsy10  
cmsy10



lmsy10  
cmsy10



lmsy10  
cmsy10



lmmi10  
cmmi10



lmmi10  
cmmi10



lmmi10  
cmmi10



lmss10  
cmss10



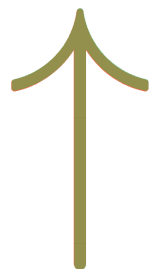
lmss10  
cmss10

CM BITMAP FONT

LM OUTLINE FONT (NEW)



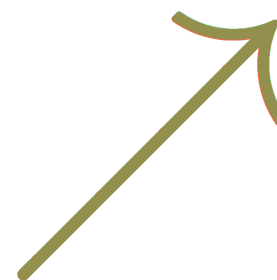
lmsy10+csc+tt  
cmsy10+csc+tt



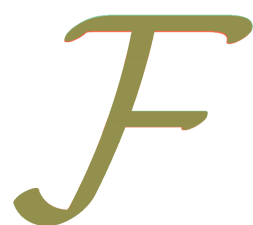
lmsy10+csc+tt  
cmsy10+csc+tt



lmsy10  
cmsy10



lmsy10  
cmsy10



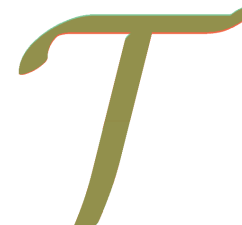
lmsy10  
cmsy10



lmsy10  
cmsy10



lmsy10  
cmsy10



lmsy10  
cmsy10



lmmi10  
cmmi10



lmmi10  
cmmi10



lmmi10  
cmmi10



lmss10  
cmss10



lmss10  
cmss10

EC BITMAP FONT

LM OUTLINE FONT (OLD)



ec-lmr10



ec-lmri10



ec-lmbx10



ec-lmss10

EC BITMAP FONT

LM OUTLINE FONT (OLD)



ec-lmr10  
ecrm1000



ec-lmri10  
ecti1000



ec-lmbx10  
ecbx1000



ec-lmss10  
ecss1000

EC BITMAP FONT



ecrm1000



ecti1000



ecbx1000



ecss1000



ecbx0500



ecbx0600



eclq8



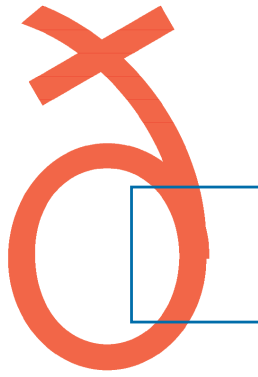
eclq8



eclb8



ecss1200



ecss1728

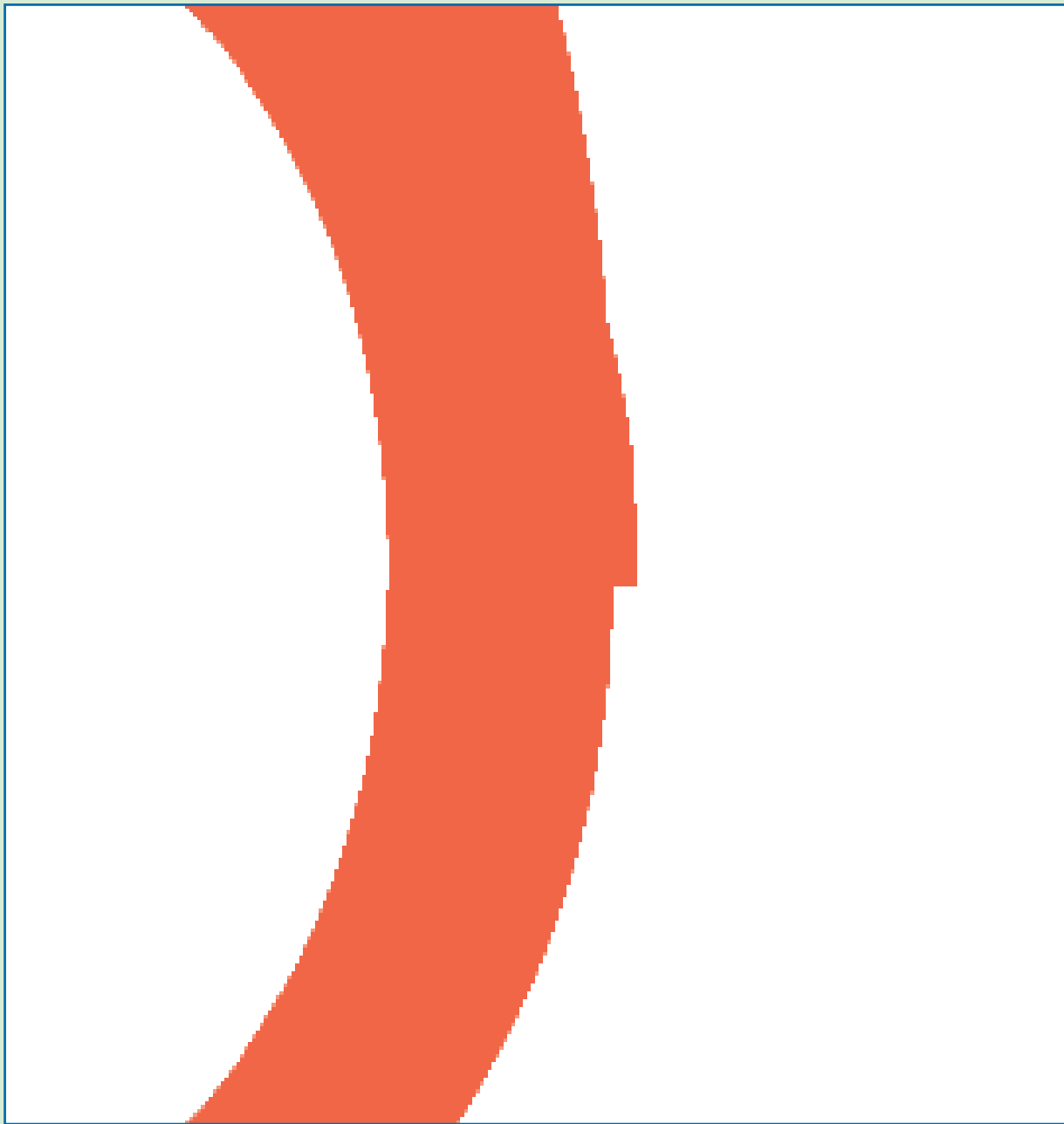


ecsx1000



ecdh1000





EC BITMAP FONT

LM OUTLINE FONT (OLD)



ec-lmr10  
ecrm1000



ec-lmri10  
ecti1000



ec-lmbx10  
ecbx1000



ec-lmss10  
ecss1000



ec-lmbx5  
ecbx0500



ec-lmbx6  
ecbx0600



ec-lmssq8  
eclq8



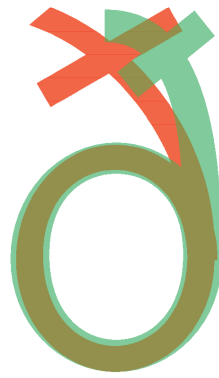
ec-lmssq8  
eclq8



ec-lmssqbx8  
eclb8



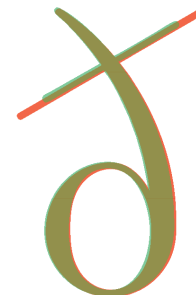
ec-lmss12  
ecss1200



ec-lmss17  
ecss1728



ec-lmssbx10  
ecsx1000



ec-lmdunh10  
ecdh1000

EC BITMAP FONT

LM OUTLINE FONT (NEW)



ec-lmr10  
ecrm1000



ec-lmri10  
ecti1000



ec-lmbx10  
ecbx1000



ec-lmss10  
ecss1000



ec-lmbx5  
ecbx0500



ec-lmbx6  
ecbx0600



ec-lmssq8  
eclq8



ec-lmssq8  
eclq8



ec-lmssqbx8  
eclb8



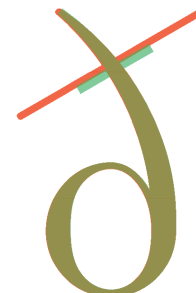
ec-lmss12  
ecss1200



ec-lmss17  
ecss1728



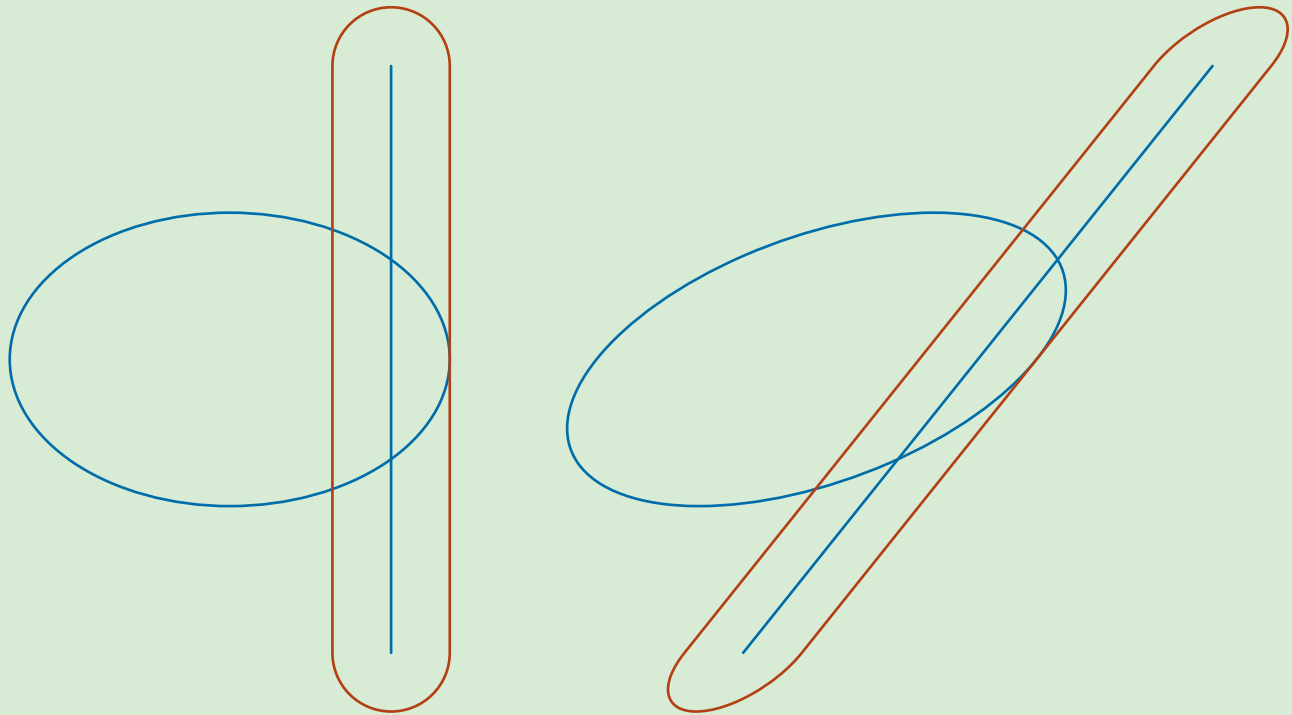
ec-lmssbx10  
ecsx1000



ec-lmdunh10  
ecdh1000

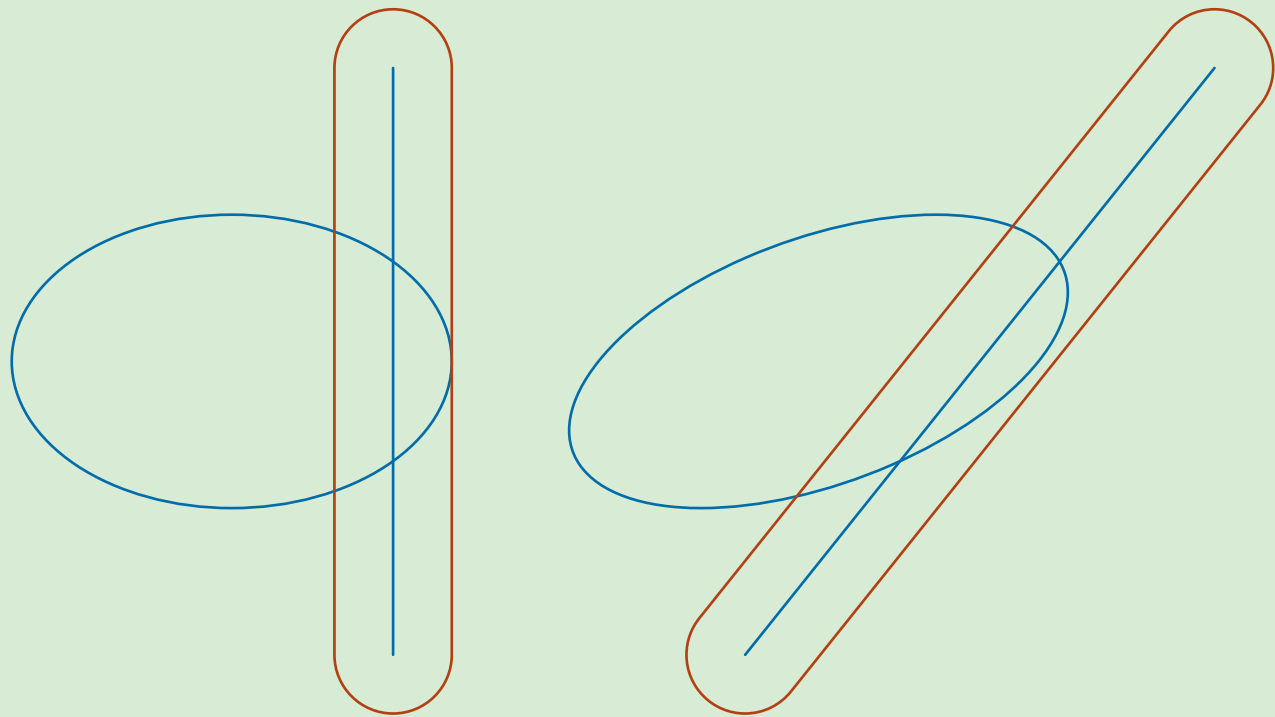
# Does slanting preserve tangency?

Answer 1: yes, of course



# Does slanting preserve tangency?

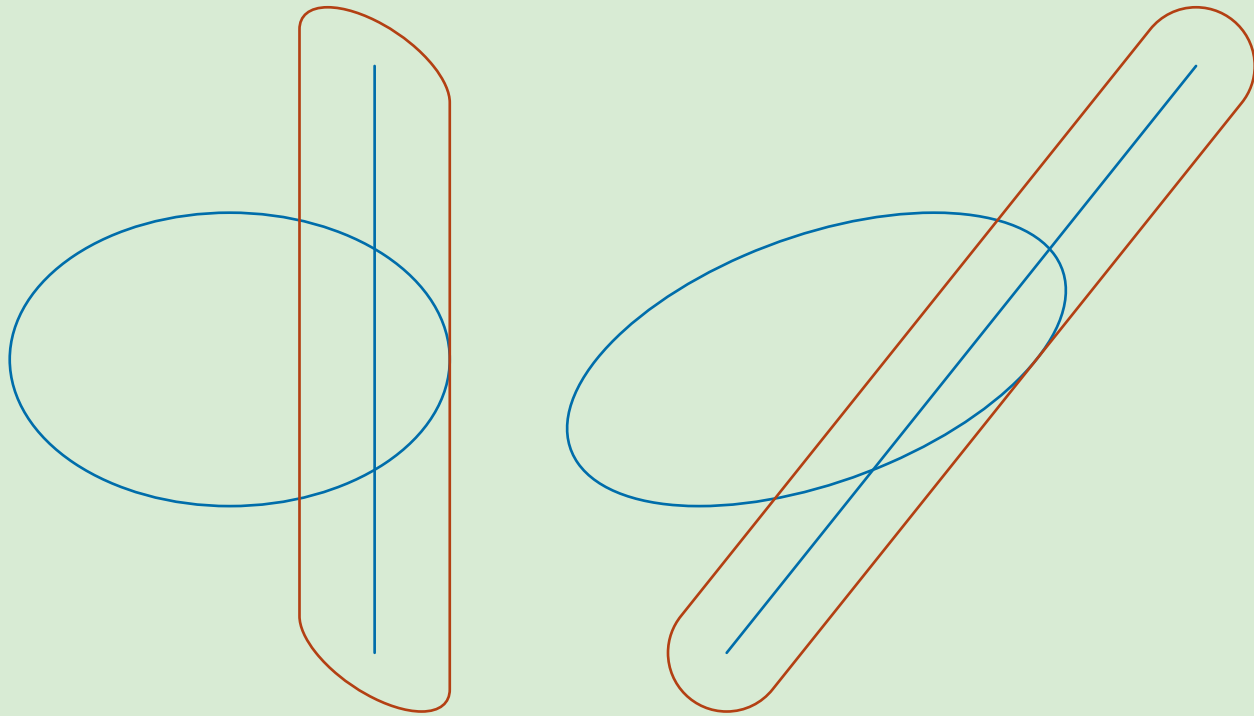
Answer 2: of course not



In Metafont and Metapost, the pen is not slanted (automatically) if the path is slanted.

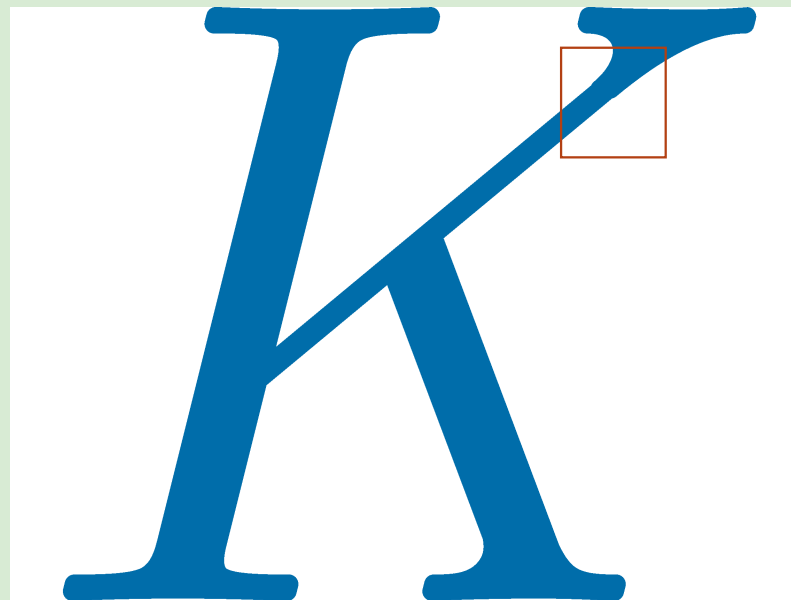
# Does slanting preserve tangency?

Answer 3: yes, but not of course



## Does slanting preserve tangency?

A trifle? Perhaps... Yet a relatively newly found bug in the Computer Modern fonts is related just to this effect:



## Does slanting preserve tangency?

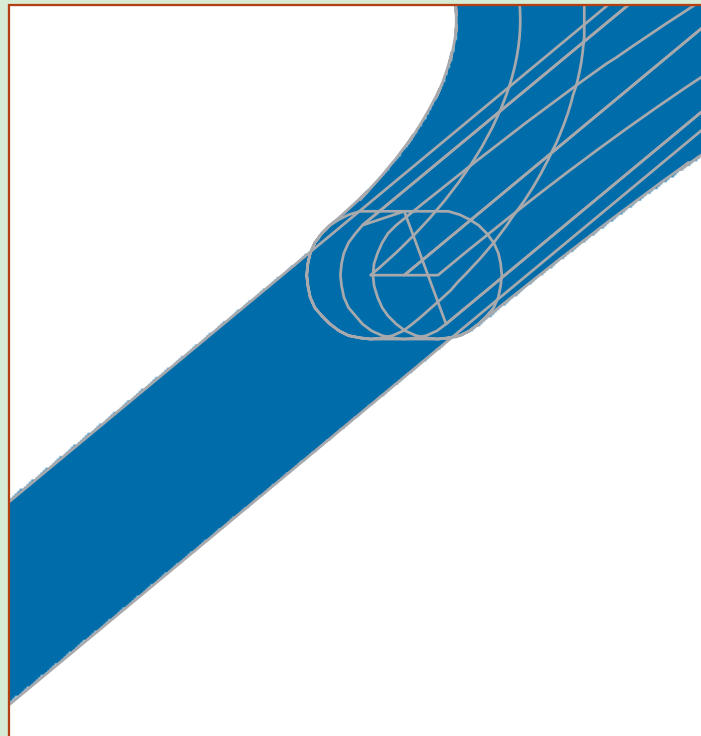
A trifle? Perhaps... Yet a relatively newly found bug in the Computer Modern fonts is related just to this effect:





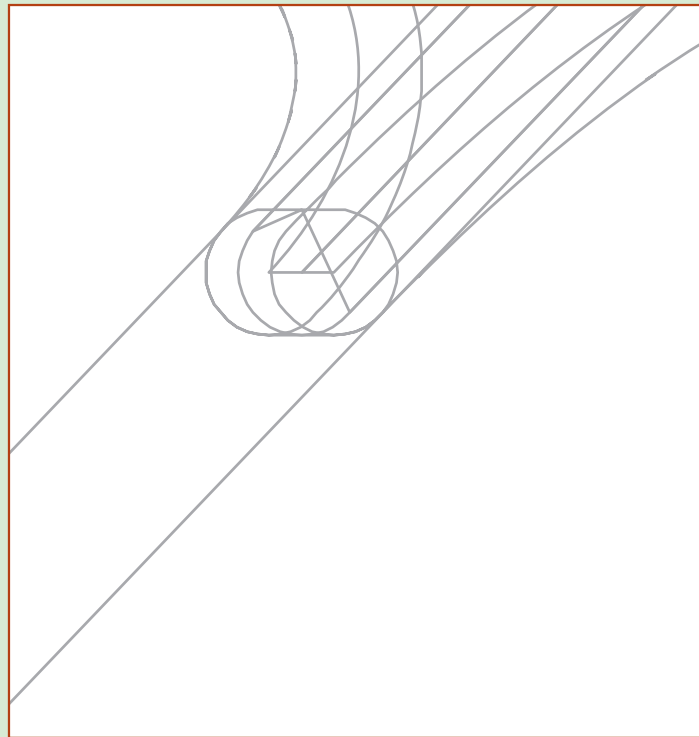
## Does slanting preserve tangency?

A trifle? Perhaps... Yet a relatively newly found bug in the Computer Modern fonts is related just to this effect:



## Does slanting preserve tangency?

A trifle? Perhaps... Yet a relatively newly found bug in the Computer Modern fonts is related just to this effect:



slant set to zero

## The directions to “attack”

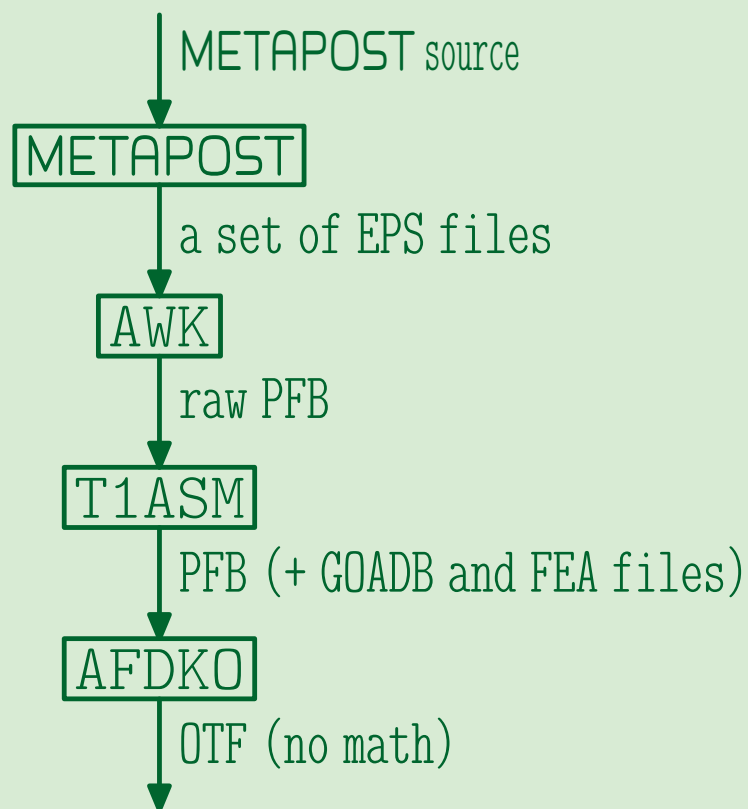
The task lies not in the creation of a single font (this can be done in a more or less manual way) but in creating of a technology providing convenient means of adding mathematics to already existing fonts, e.g., to the T<sub>E</sub>X Gyre family; therefore, GUI only tools are of no interest.

## The directions to “attack”

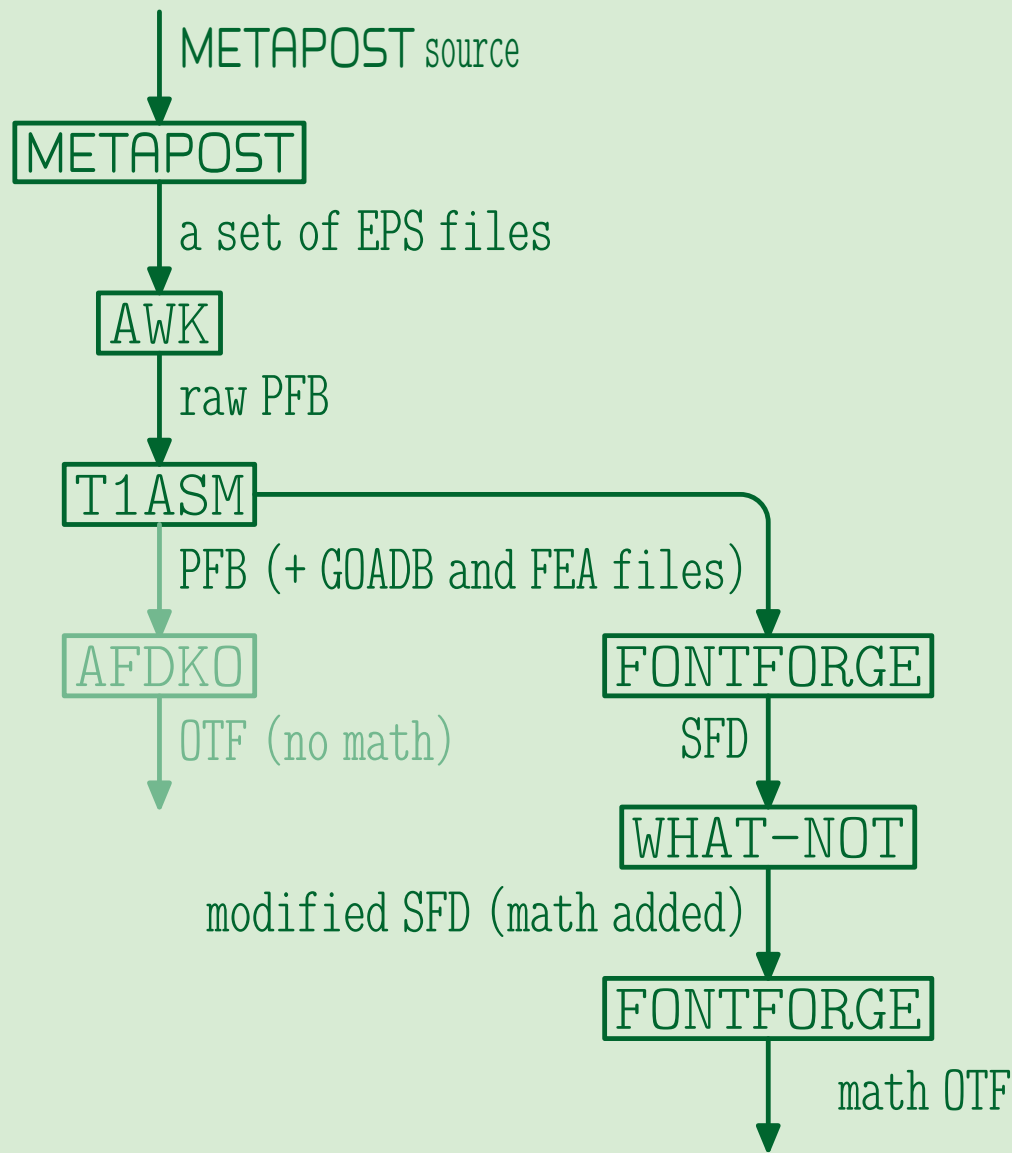
The task lies not in the creation of a single font (this can be done in a more or less manual way) but in creating of a technology providing convenient means of adding mathematics to already existing fonts, e.g., to the T<sub>E</sub>X Gyre family; therefore, GUI only tools are of no interest.

Therefore, the main direction to attack: remaking of MetaType1 into MetaTypeO, i.e., Metapost + Python scripts to generate OpenType fonts directly using the ‘fontforge’ Python module.

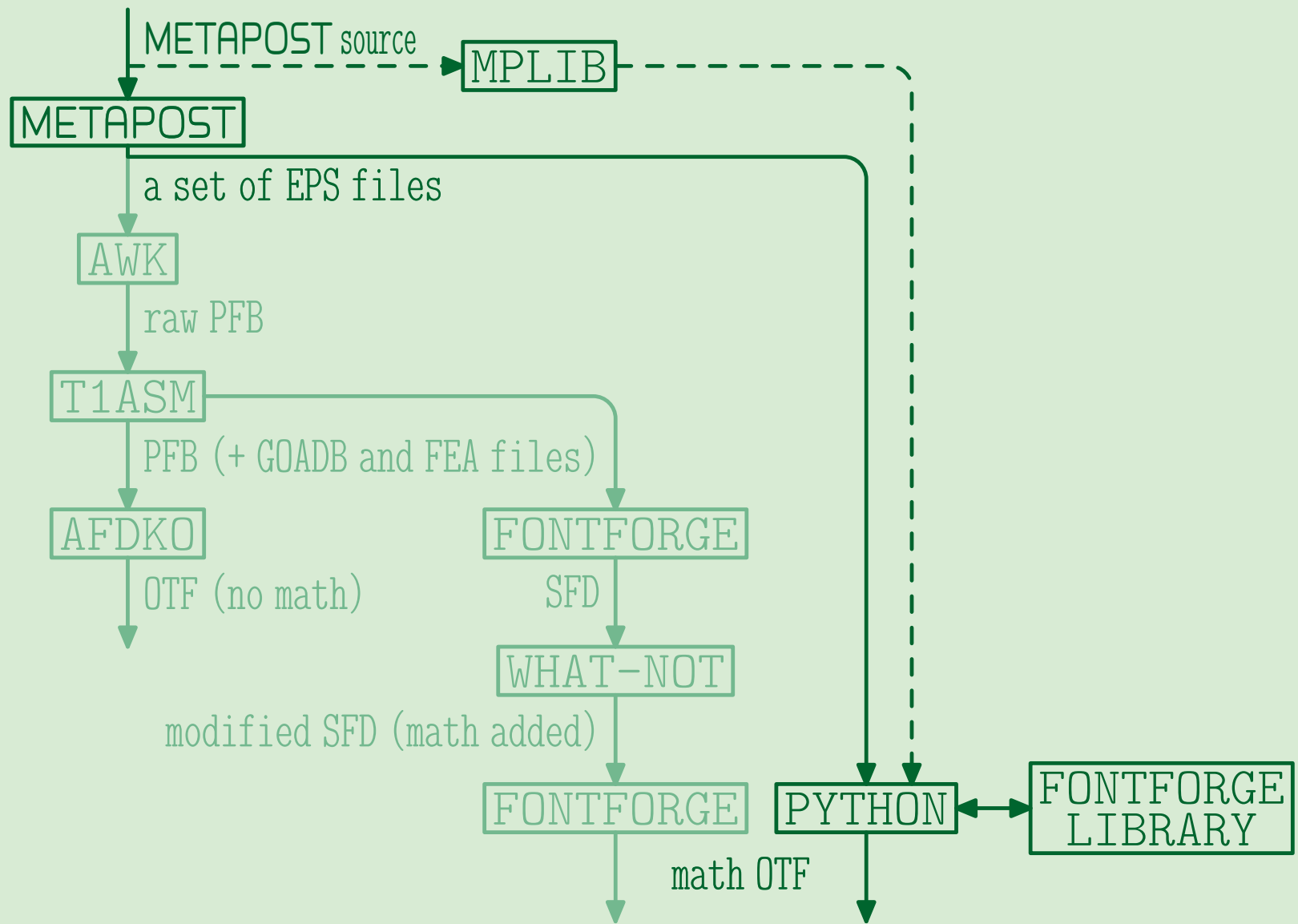
# The directions to “attack”



# The directions to “attack”



# The directions to “attack”



## Near future actions

Before the end of this year, we hope to complete the following steps:

- release as soon as possible the versions 2.xxx of the T<sub>E</sub>X Gyre and Latin Modern fonts (LPPL),
- “simulate” AFDKO with a Python script using the ‘fontforge’ module,
- extend (provisionally) that tool to handle math by processing SFD files – currently, the AFDKO notation for the FEA files does not cover math,
- create, as a “warm-up”, math OpenType fonts for the Latin Modern family, first through the use of TFM’s; the TFM data will be in a semi-heuristic way converted into OpenType font math tables.



The task turned out to be much more difficult than the font enterprises we took part in so far. And, of course, as we optimistically expected. But we are still optimistic having sound help from LUGs both as groups, and as individual members, to name three of them among many: Johannes Küster, Karel Píška and Ulrik Vieth –

**THANK YOU!**

The OpenType math fonts project is supported by T<sub>E</sub>X Users Groups, in particular, by the Czechoslovak T<sub>E</sub>X Users Group CSTUG, the German-speaking T<sub>E</sub>X Users Group DANTE e.V., the Polish T<sub>E</sub>X Users Group GUST, the Dutch-speaking T<sub>E</sub>X Users Group NTG, TUG India, UK-TUG, and – last but not least – TUG.