

Welcome

to the world of

MetaFun

# What is MetaFun

$\text{T}_{\text{E}}\text{X}$  does not have a graphical engine, and can only draw horizontal and vertical lines.

... but ...

$\text{T}_{\text{E}}\text{X}$  has a powerful extension mechanism (using specials) and can write and read files.

... and ...

$\text{T}_{\text{E}}\text{X}$  has a beautiful sister called `METAPOST`, which can be used for non artistic graphics.

... this is why ...

**MetaFun** is able to give  $\text{T}_{\text{E}}\text{X}$  a couple of interesting graphic features based on the `METAPOST` engine.

... so ...

**MetaFun** is just an interface between  $\text{T}_{\text{E}}\text{X}$  and `METAPOST`, and also extends existing `ConT_{\text{E}}\text{X}t` functionality.



# Using MetaFun

You can conveniently combine **MetaFun** graphics with ConT<sub>E</sub>Xt, and base your graphics on information that the typographic engine provides and vice versa.

You can use ConT<sub>E</sub>Xt to create stand alone graphics, that can be integrated in other T<sub>E</sub>X macro packages.

You can use some of the text related features in other macro packages than ConT<sub>E</sub>Xt.

There are many (classes) of METAPOST macros that can be used in pure METAPOST graphics.

You can use plain METAPOST but still use the **MetaFun** manual to learn a few tricks.



# What Can MetaFun Do

If you just look at METAPOST, the **MetaFun** format provides you a series of additional macros.

You can use **MetaFun** to combine graphics and text, but that feature is not yet generic (and depends on ConTEXt).

You can use **MetaFun** to create outline texts; this is an independent feature and goes under the name MPY.

Although there are some limitations, figures can be integrated into METAPOST graphics.

**MetaFun** implements a special driver that enables you to add special effects, like shading.



# More MetaFun

There will be a bit more communication between METAPOST and T<sub>E</sub>X.

More special effects will be added, using the special driver.

More parts of ConT<sub>E</sub>Xt will be capable of communicating with the graphic backend.

Libraries with predefined features and graphics are and will be part of the ConT<sub>E</sub>Xt distribution.

There will be a bit more documentation and much more examples, like like [this](#), [this](#), and [this](#).



# Macros, Tools and Manuals

There are two **manuals** on METAPOST and **MetaFun**: **screen** and **paper**.

There is a manual dedicated to **making outlines** with makempy.

You can use **T<sub>E</sub>XEXEC** to process METAPOST files.

You can use mptopdf or T<sub>E</sub>XEXEC to convert METAPOST output (multiple) figures.

