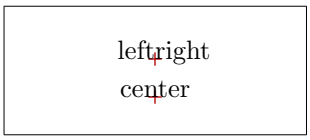
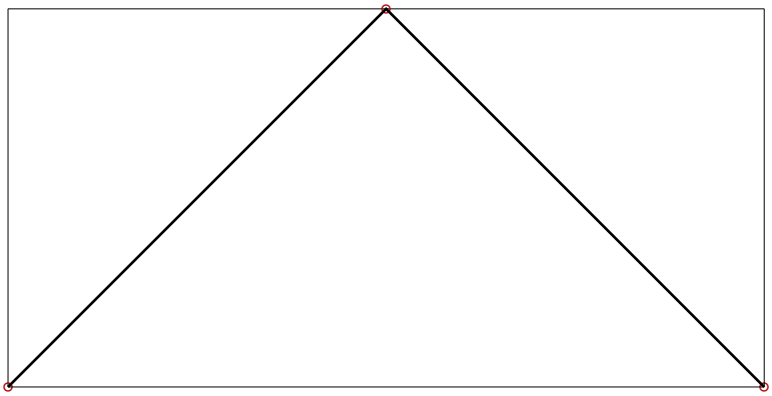
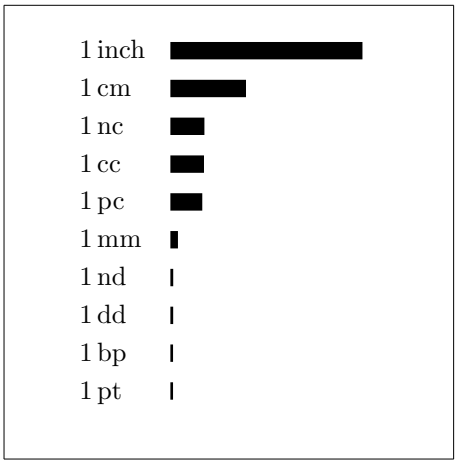
The box below is entirely typeset with TEX. Lua is not involved. We don't use the luagraphic library at all. The markers are from a special Type 1 font created with MetaType1 where all glyphs have zero width, height, and depth.



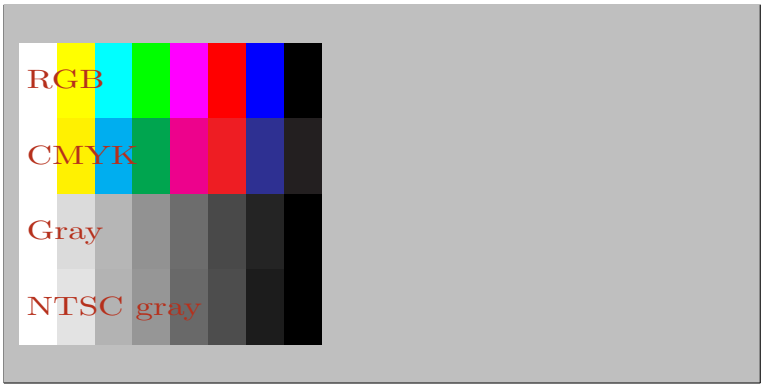
Now we combine TEX and Lua. Markers are inserted by TEX, lines by Lua.



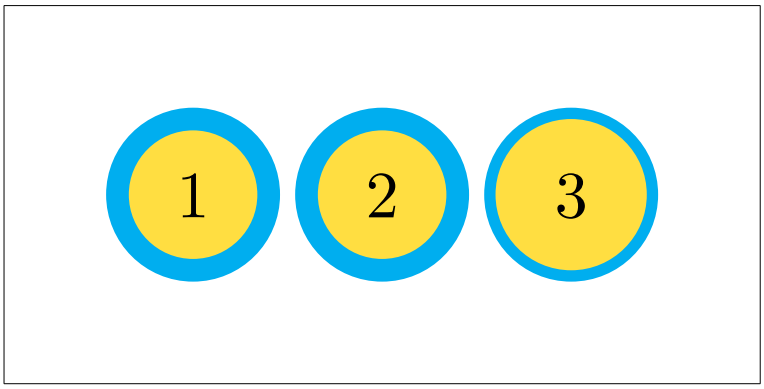
Units. Default unit is 1 millimeter. Other units can be used as well.



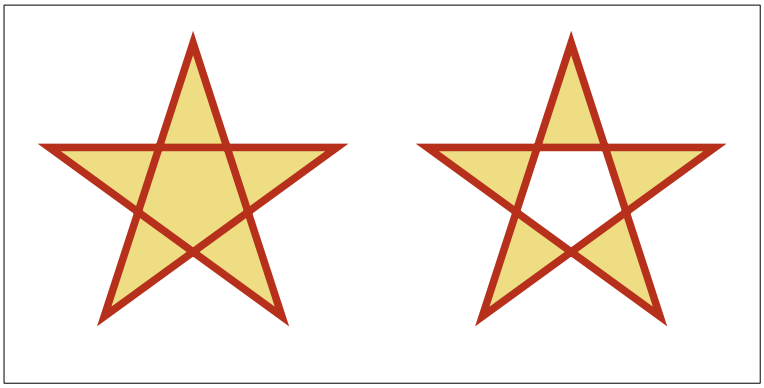
Colors can be specified by a list of components. The color model is derived from the number of list items: Gray(1), RGB(3), and CMYK(4). Instead of a list, named colors can be used as defined in `namedcolors.lua`, for instance `dvips.BrickRed`.



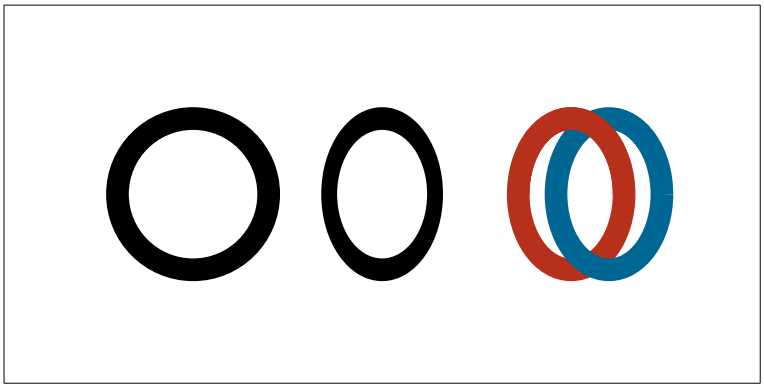
PDF supports operators to fill and stroke objects. Objects are always first filled and then stroked. We can reverse the order though. Example (1) utilizes PDF's operator for `fill` and `stroke`, (2) calls `fill` and `stroke` separately in the same and (3) in reverse order.



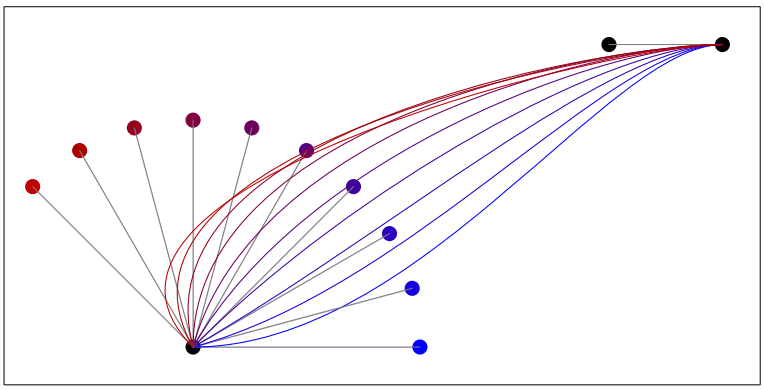
Some geometric objects can be described more conveniently in polar coordinates. The example below shows the difference between the "nonzero winding number rule" and "even-odd rule" filling.



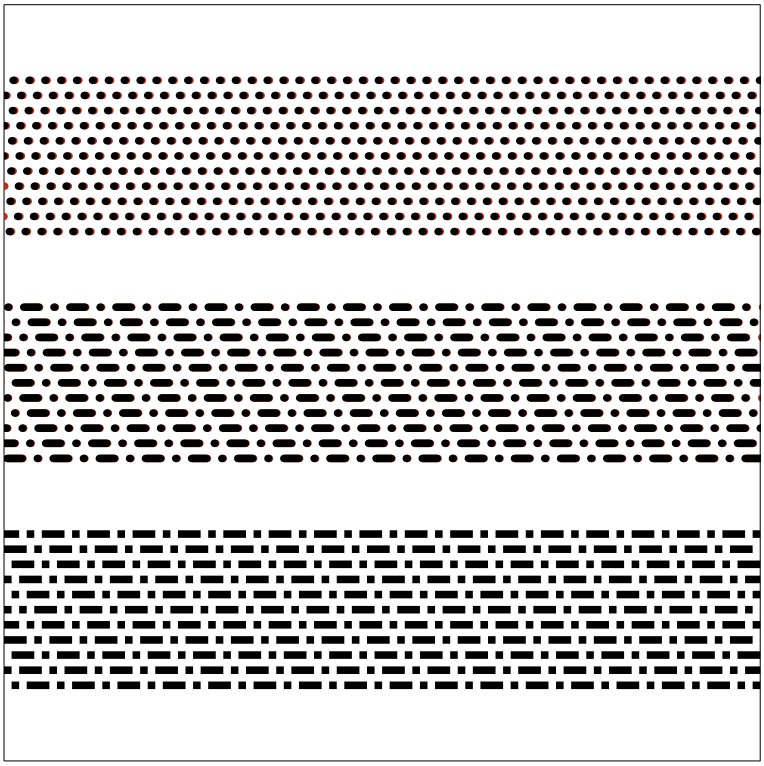
Ellipses can be created by changing the transformation matrix. This leads to non-uniform linewidths (middle). But we can also scale the path before it is painted (right).



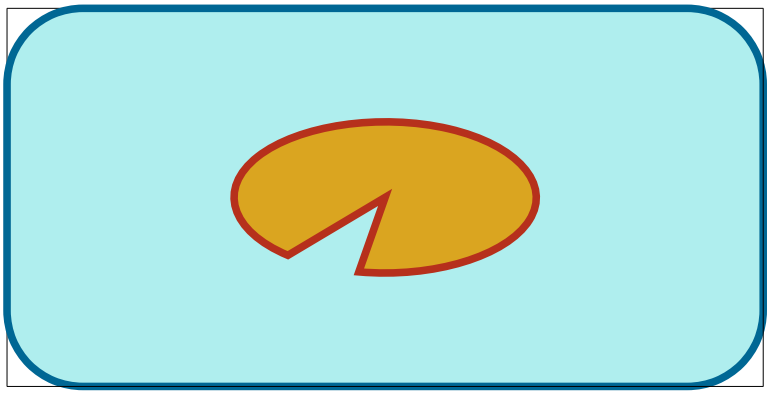
Some cubic Bézier curves. They are described by one start and one end point and two control points.



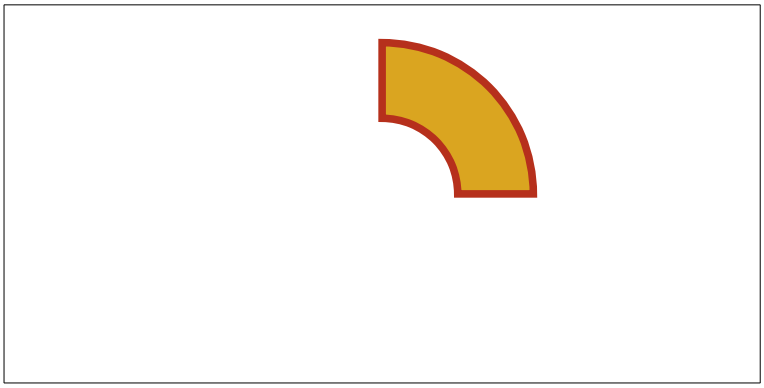
Dash patterns. No PDF viewer supports them properly. Missing elements are marked red. Ghostscript and Adobe Reader are best (though not perfect), Okular is worse, and Evince is completely unusable.



Arcs are implemented as a concatenation of four Bézier curves. Though angles in Lua are in radians we allow users to specify them in degrees. This is in conformance with PostScript. Usually paths are specified counterclockwise but...

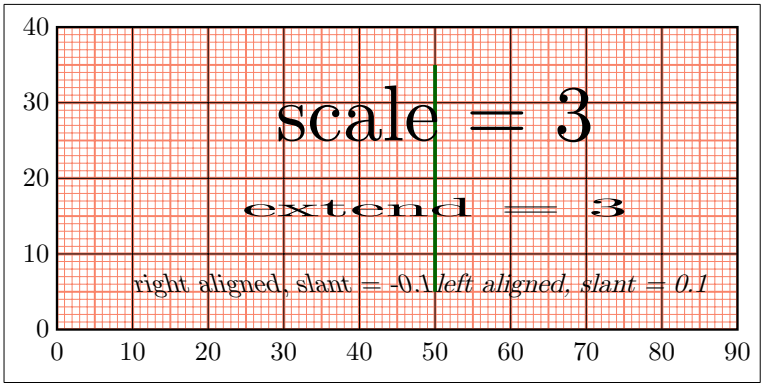


...we need "negative arcs" in some situations too. Paths supposed to be filled must be contiguous. This means that `moveto()` is only allowed once at the beginning of a path. In the example below the lower left arc is running clockwise.



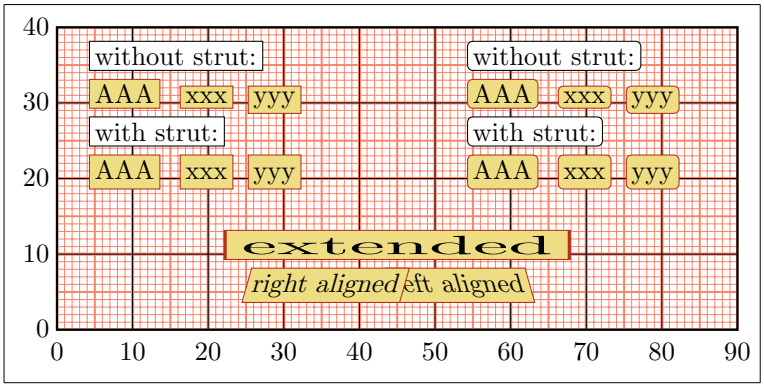
Now we add text inserted by Lua. Coordinate pairs are still in millmeters but text size is in pt. Thus text size is the same of surrounding text (if not deliberately scaled).

Text can be scaled and dvips options `slantfont` and `extendfont` are supported. Text can be left or right aligned or centered (default). Vertical alignment refers to the base line.

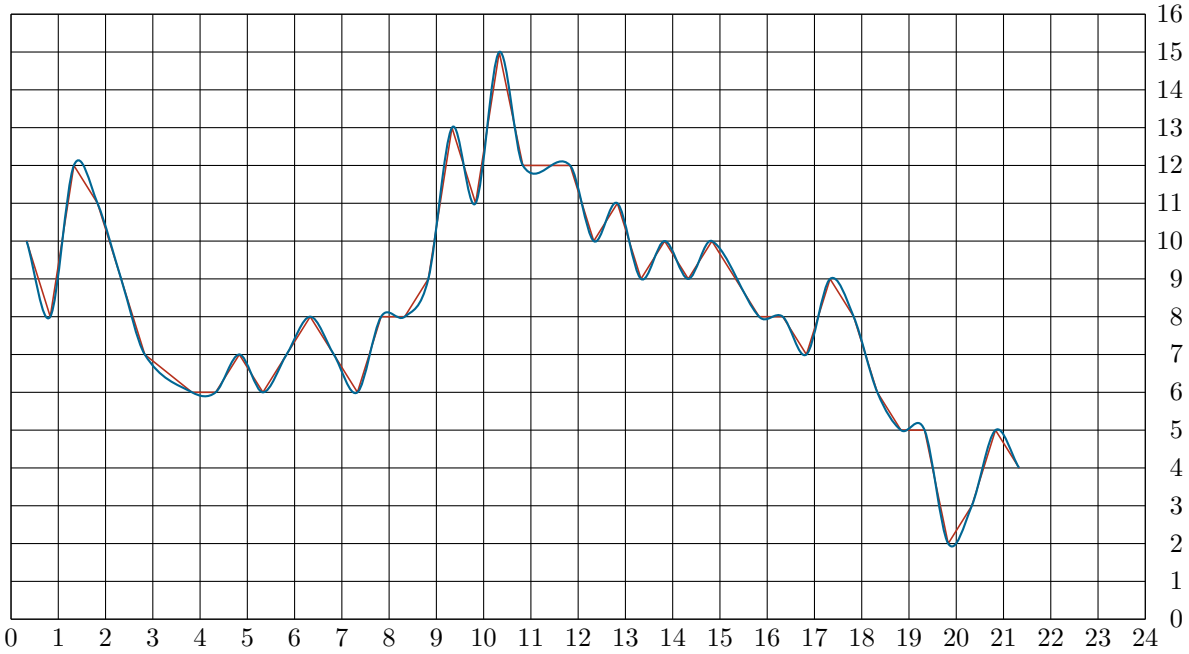


Framed text. The height of the box depends on the shape of the glyphs therein. If all boxes are supposed to have the same height and depth you have to add a strut. A strut is an empty box with zero width. The height is determined by the glyphs therein, in our case '`Ay`'.

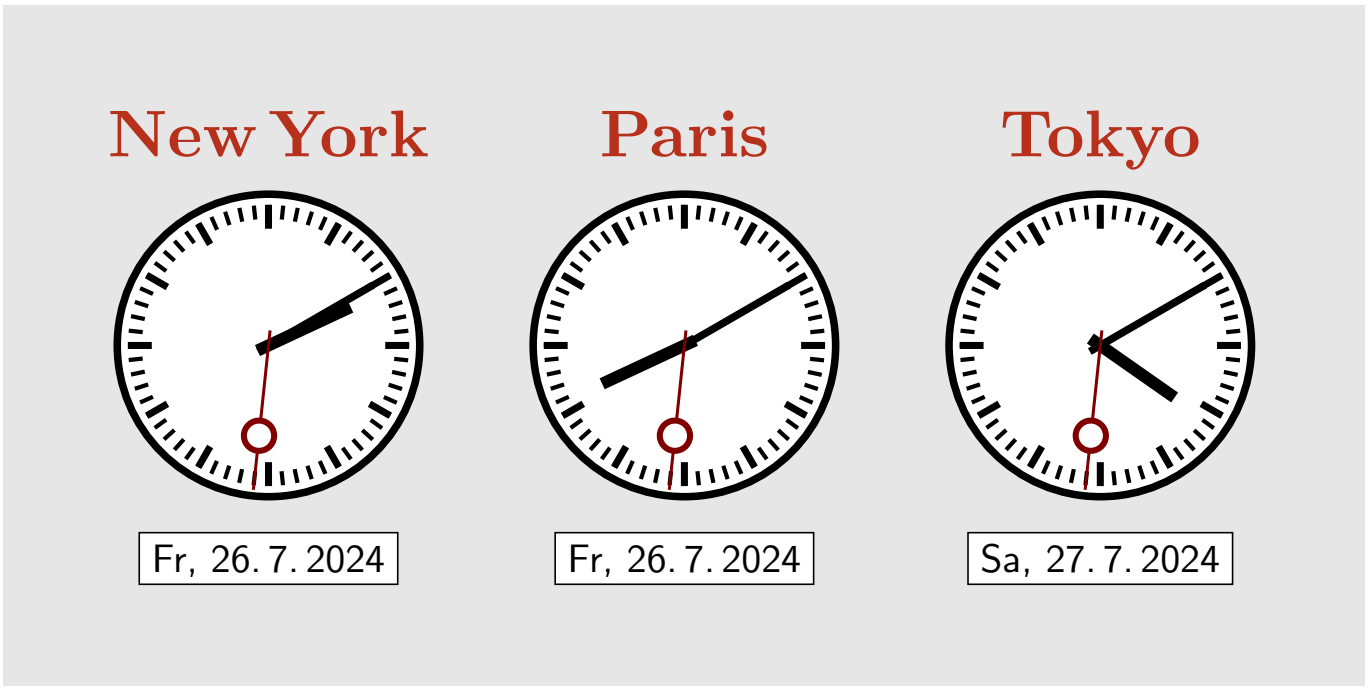
It's obvious that some matrix operations are deprecated because they cause undesired results.



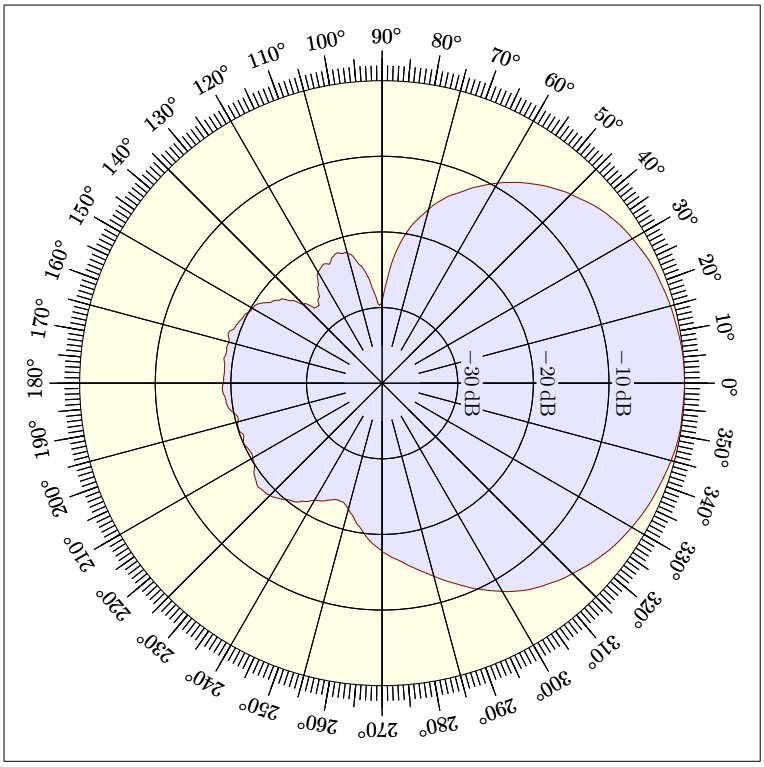
Now we read data from a file. The path is then replaced by a smooth curve.



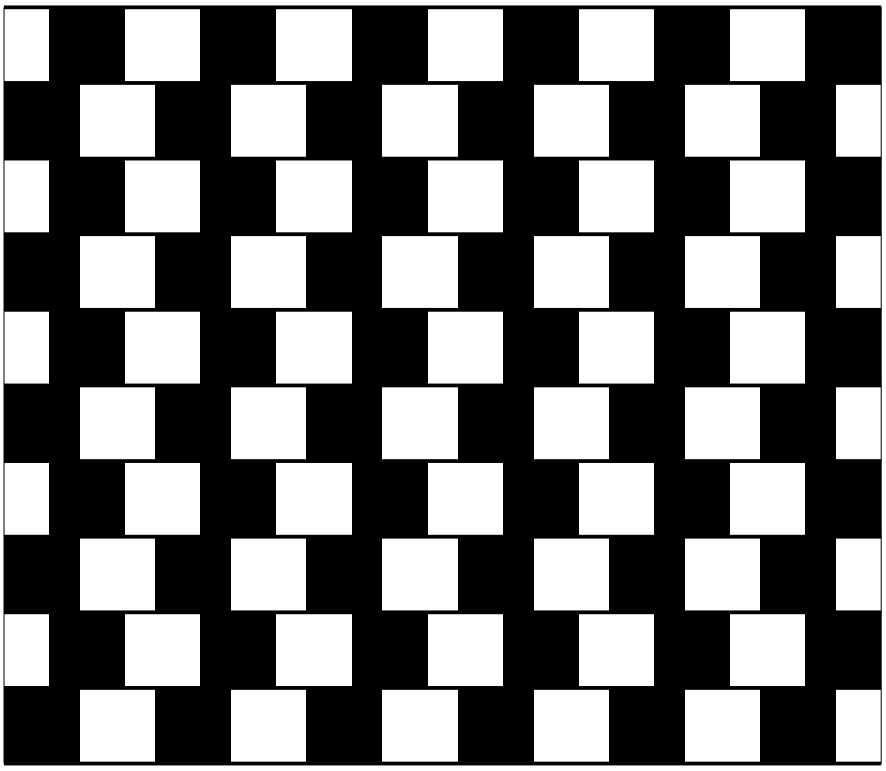
Here is a clock like those seen at German railway stations.



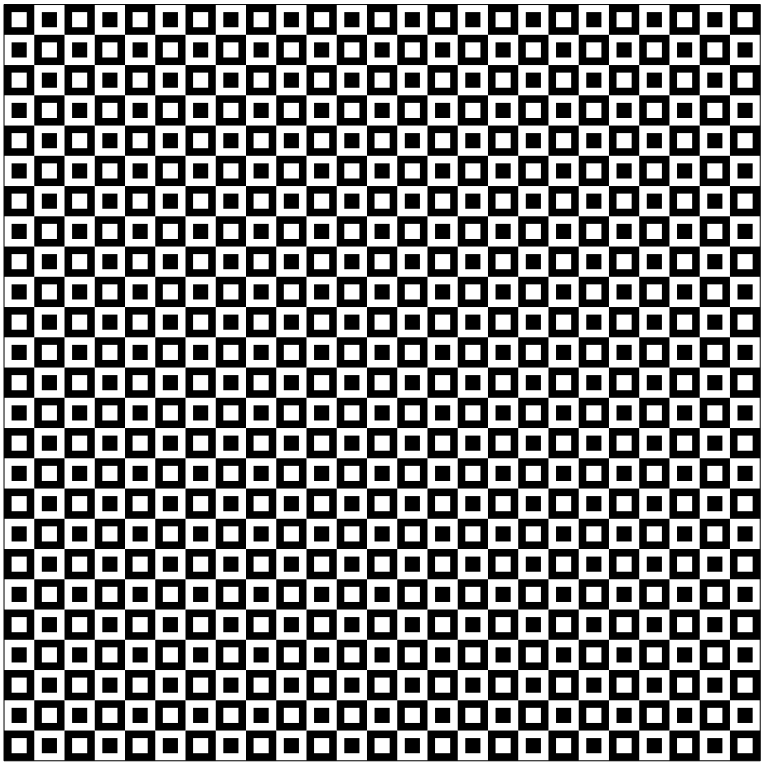
Antenna radiation patterns are usually provided in polar coordinates.



Are the horizontal lines straight?



Jiggling Squares (Kees van der Laan at EuroTEX 2012)



Arrow heads