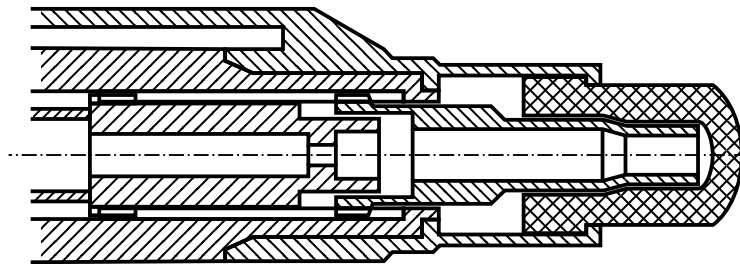
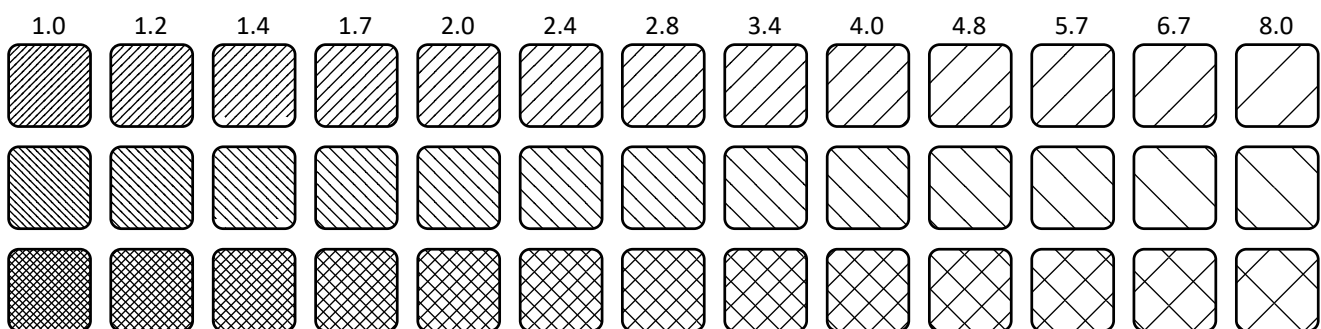


Patterns for mechanical drawings in Inkscape

Images in patents must look like classical pen drawings. For such images, old fashioned hatch patterns can be handy.

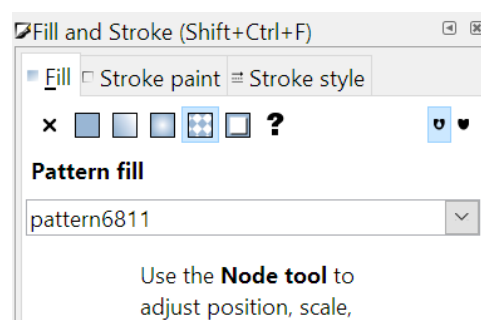


I created the following set of 3x13 patterns to be used in Inkscape. A name was assigned to each pattern, such as *Hatch 2.0 x 45°* for a 2 mm x 45° pattern.

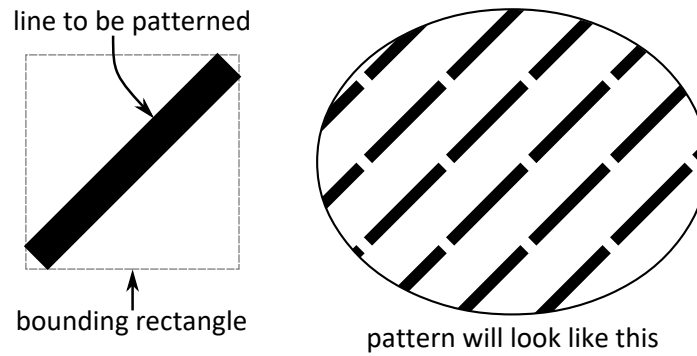


Creating a pattern in Inkscape

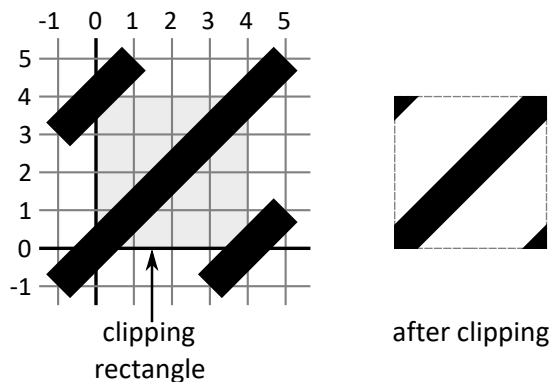
In Inkscape, a pattern can easily be created by selecting a group of objects and choosing *Object > Pattern > Objects to pattern* from the menu, or alt-I (on Windows and Linux). What happens is that Inkscape determines the bounding rectangle around the selected objects and assigns a name to the pattern, such as *pattern6811*. We can use the pattern on a selected object by choosing the pattern button from the *Fill and Stroke* tool:



If we pattern a simple 45° line, the following happens:



We can remove the white space between the lines by adding two additional lines, grouping the lines and applying a clipping rectangle:



If we use this to create a pattern, a perfect 45° hatch pattern will be the result. However, this is a significant amount of work if we want to create 39 patterns.

Using Python to create pattern definitions

Therefore, Python was used to create the text file in SVG format, containing all pattern definitions.

A simple SVG file looks like this:

```
<svg width="1350" height="220"
xmlns="http://www.w3.org/2000/svg">
<defs>
...
</defs>
...
</svg>
```

← pattern definitions go here

← shapes in the drawing go here

A pattern definition looks like this:

```
<pattern
id="..."
inkscape:stockid="..."
x="0"
y="0"
width="..."
height="..."
```

← here we must fill in the name of the pattern

← this is what appears in the Inkscape drop-down

← this is the left of the clipping rectangle

← this is the bottom of the clipping rectangle

← this is the width of the clipping rectangle

← this is the height of the clipping rectangle

```
patternUnits="userSpaceOnUse">
...
</pattern>
```

← here we must define the objects of the pattern

We don't have to clip the objects that make up the pattern: the x, y, width and height properties define the clipping rectangle. In our case, the pattern definition is only made up of three lines. A simple line in SVG is defined like this:

```
<path
style="stroke:#000000;stroke-width:0.25mm"
d="M ... , ... .. , ... .."

/>
```

← this defines line color and –width
← this defines beginning and end points

Now each pattern will be created essentially by drawing 3 lines in a clipping square. To create the 45° hatch patterns, we draw three lines with the following coordinates:

The pitch of the hatch pattern is equal to the diagonal size of the square. The width of the clipping square is equal to the pitch multiplied by $\sqrt{2}$. To create a cross hatch pattern, the long lines of both patterns must be combined, the short lines are not needed.

The Python *HatchPatterns.py* uses string templates to create all these snippets of SVG text and compile them into an SVG file.

This script creates two separate files: *HatchPatterns.svg* and *Defs.txt*. Each can be used in a different way.

How to use these patterns

There are two ways to use the files generated by *HatchPatterns.py*:

1. *HatchPatterns.svg* contains a rectangle for each pattern. You can *copy/paste* these rectangles to the drawing where you want to use them, and *copy/ctrl-paste* the object properties to the objects that must have the patterns. *Copy/ctrl-pasting* object properties is a very fast way to work.
2. The second solution is to open the `..\share\patterns\pattern.svg` file in a text editor and append the contents of *Defs.txt* to the `<defs> ... </defs>` section. With this second option, the patterns will be available in every Inkscape session from then on.

Hope this is useful!

HenkJan van der Pol - January 2, 2017

Enclosed:

1. *HatchPatterns.py*
2. *HatchPatterns.svg*
3. *Defs.txt*