

Why G-code is an advantage for an open machine

What is G-Code

G-code is a standard language born for defining paths executed by CNC.

It became the de-facto standard for 3d filament printer (FFF) and it is composed of three classes of commands, movements, tool selection (filament extrusion tool in FFF) and special commands.

Movements are the commands starting with G0 or G1 and they define where to move the referenced axes. The tool command is a 'T' letter followed by a number and the number can be greater or equal to 0, example of valid commands are 'T0', 'T1', and 'T5'.

Examples are:

T0 ; Select tool 0

G0 X10 ; Move X axes 10mm from the origin

G0 X10 Y15 ; Move X axes 10mm and Y axes 15mm from the origin

G1 X20 Y20 E10 ; Move X axes 20mm and Y axes 20mm from the origin extruding 10mm

G1 Z0.1 ; Move the Z axes means layer change

In our laser machines the G-code is used to define the path of the laser on the powder bed and when the E axes is specified with a value different from zero the laser is turned on while doing the movement.

The tool is used to define the setting of the laser, speed and power, that has to be used when the laser is on while moving. So the commands:

T0

G1 X15 Y15 E1

have the effect to select tool 0, the laser parameters identified by tool 0, and move the laser from the current position to position (15,15) with the laser turned on.

The following example draws a square with 1cm side on the powder bed using T0 settings for the horizontal lines and T1 settings for the vertical lines.

T0

G0 X10 Y10

G1 X20 Y10 E1

T1

G1 X20 Y20 E2

T0

G1 X10 Y20 E2

T1

G1 X10 Y10 E4

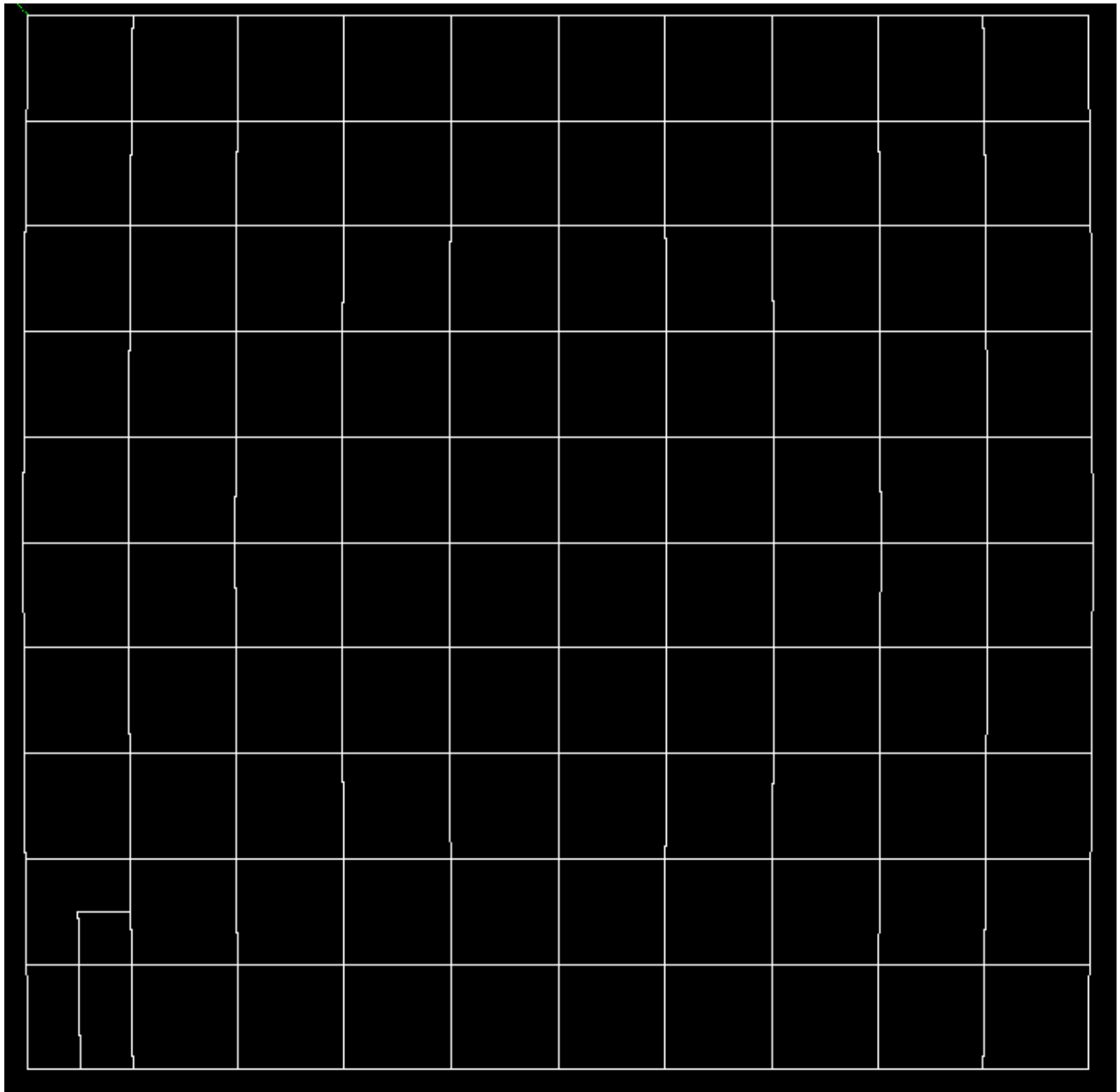
You can find more information with a simple search, some useful links are:

<https://en.wikipedia.org/wiki/G-code>

<https://marlinfw.org/meta/gcode/>

Write your G-code

The first possibility is to directly write a single layer as we have done when we wrote one of the first calibration tests of the SnowWhite, the “test_dist” a simple grid used to verify the correctness of the distances.



Doing it you have complete control over the path.

Software generation of objects

Another option is to use a software that generates the G-code based on parameters or another kind of language as in the case of Full Control Gcode Design, <https://fullcontrolgcode.com/>

Other slicers

The first thing to note is that you can use any slicer that generates a standard gcode; you have only to pay attention to a few things:

- the origin should be in lower left of the building area;
- the movements needs to be in millimeters and absolute.

Example of slicing software that can be use are:

Slic3r	https://slic3r.org/
Cura	https://ultimaker.com/en/software/ultimaker-cura
Simplify3D	https://www.simplify3d.com/
IceSL	https://icesl.loria.fr/

From our experience the use of FFF slicers is only a matter of turning off all the options related to the retractions and handling of speed/temperatures then understanding how to change the relevant parameters of laser machines (hatching distance, shell distance, number of shells and so on).

What you can do modifying G-code?

Another possibility is to repeat or substitute the gcode to obtain a particular effect, examples are remelting and infill with a pattern.

Remelting

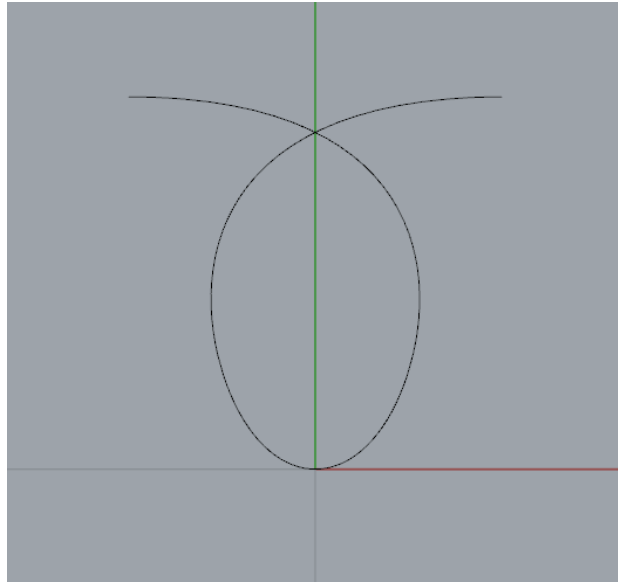
The remelting is a method of improving the surface quality and consists in a second pass on the external surface of the object done al last part of each layer. Usually with an FDM slicer you can define both the number of skins (perimeter of the object in the XY plan) and the tool to be used when building; in this case the remelting can be implemented simply writing at the end of each layer, i.e. just before a positive increment in Z, all the movements defined between the skin tool change and the following non skin tool change of the same layer.

Infill with a pattern

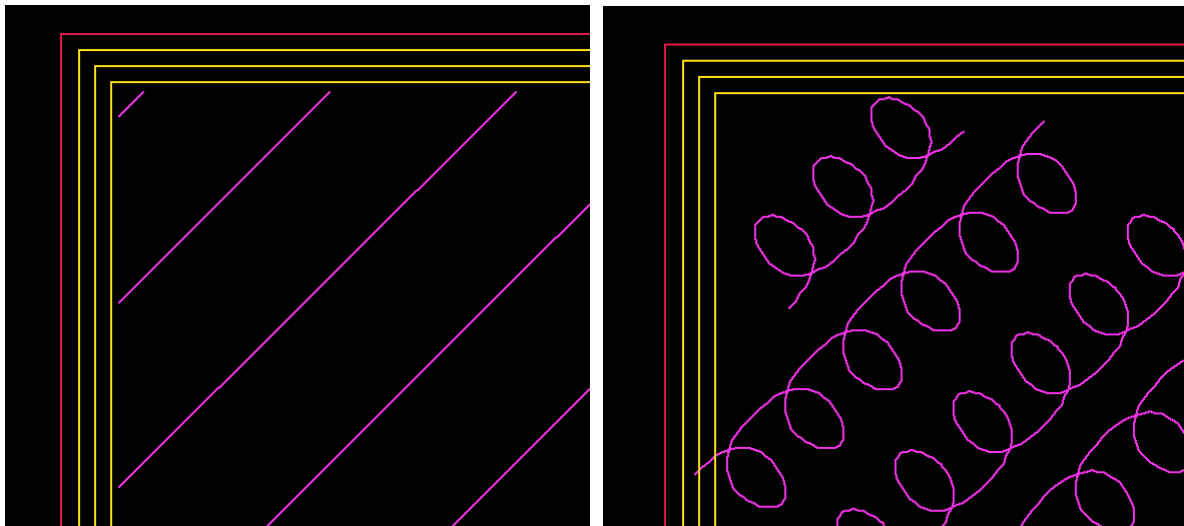
As the G-code is a sequence of lines another idea is to change the shape of the infill pattern without modifying the slicer. The first thing is to correctly identify infill lines, in our Cura setup for metalONE the infill tool is T4 and is used for infill only.

Then the method used is to define a pattern with a defined width and height, knowing that the line defined by the G-code defines also the rotation on the XY plane we can rotate the patter in the direction of the laser movement.

To test our idea we designed this pattern:



then generated the corresponding G-code and wrote a python script that substituted every occurrence of the movements of the T4 tool with a repetition of our pattern expressed via the correctly rotated G-code.



After some trial and errors we got a working prototype, on the left a detail of the starting paths (only straight lines) and on the right the resulting paths with the pattern repeated instead of the infill lines.

Sharebot r&d Team

If you want to discuss, comment or deepen please mail to metalone.support@sharebot.it