Initiation to 3D Printing – Practical exercises 2

Sylvain Lefebvre and Camille Schreck, ENSEM 2020-2021

Friday 21ˢᵗ January, 2022

1 Important information

- The code can be writeen in either C, C++, Python, or JAVA language, your choice.

- At the end of the session, send the **code and GCode of exercises 3, 4, 5, and 6** packed into a single ZIP file to:

  - sylvain.lefebvre@inria.fr
  - camille.schreck@inria.fr

  with the mail subject **ENSEM: TP2 [nom][prenom]**.

- **Name your zip file: TP2_[nom]_[prenom].zip**.

- Before leaving the class, check with the professor that the mail with your ZIP was well received.

2 Useful Links

- To write and test GCode `https://icesl.loria.fr/webprinter/`
  (older version: `http://shapeforge.loria.fr/vrprinter`)

- Another GCode viewer `http://gcode.ws`

- List of GCode instructions `http://marlinfw.org/meta/gcode/`

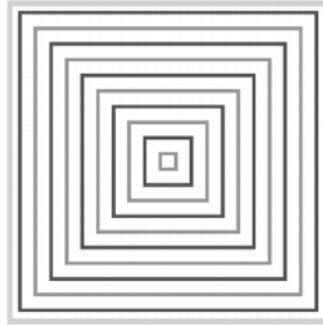## 3 Exercise: filling the square (zigzag infill)

1. Consider a square with dimensions 40×40mm. Implement the filling of the square (slice) with a zigzag infill. The zigzag main direction rotated 90 degrees at each slice (left/right, then front/back). Beware of overlaps!

2. Modify the program to output 50 layers of thickness 0.2mm, for an object of 10mm height in total.

4 Exercise: filling the square (contour parallel infill)

1. Consider again a square with dimensions 40×40mm. Implement the filling of the square (slice) with a contour parallel infill.

2. Modify the program to output 50 layers of thickness 0.2mm, for an object of 10mm height in total.

5 Exercise: mixing zig zag and contour parallel

1. Consider again square with dimensions 40×40mm. Implement the filling of the square with two parallel contours, followed by the zig-zag infill.

2. Modify the program to output 50 layers of thickness 0.2mm, for an object of 10mm height in total.

6 Exercise: filling the hemisphere

1. Implement a program outputting the GCode of of circle of 20mm radius on one layer. It will be filled with two contour parallel tracks, followed by a zigzag infill.

2. Bonus: Implement a program outputting the GCode of an hemisphere of 20mm radius with 0.2mm layers. Each layer will be filled with two contour parallel tracks, followed by a zigzag infill.

7 Miscellaneous: sample code C++

```cpp
#include <iostream>
#include <fstream>
#include <cmath>  // use constant M_PI to get the value of pi

int main () {
    std::ofstream file;
    file.open ("square.gcode");
    // header
    file << "G21" << std::endl;  // dimensions in milimeters
    file << "G90" << std::endl;  // absolute positioning
    file << "G28" << std::endl;  // homing

    // exercise code
    file.close();
    return 0;
}
```

In Linux, compile the above program (contained in a file main.cpp) with:

```
g++ main.cpp -o main
```