Initiation to 3D Printing – Practical exercises 3

Sylvain Lefebvre and Camille Schreck, ENSEM 2020-2021

January 25th 2022

1 Important information

- The code can be written in either C, C++, Python, or JAVA language, your choice.
- At the end of the session, send the **code and GCode of Exercise 3, 4 and 5 (optional bonus)** packed into a single ZIP file to:
  - sylvain.lefebvre@inria.fr
  - camille.schreck@inria.fr

  with the mail subject **ENSEM: TP3 [nom][prenom]**.
- **Name your zip file: TP3_[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 `http://shapeforge.loria.fr/vrprinter`
- Another GCode viewer `http://gcode.ws`
- List of GCode instructions `http://marlinfw.org/meta/gcode/`

3 Slicing algorithm

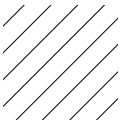
Complete the slicing algorithm written during class (slicing.cc) but computing the correct value of E.

4 Sparse filling a cube

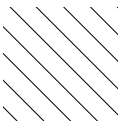
4.1 Three sets of parallel lines in a square

We are going to fill a square of 36×36 mm with three sets of parallel lines:

1. The first set is at a 45 degree angle with a spacing of 4mm.



2. The second set is at a -45 degrees angle with a spacing of 4mm.



3. The last set is at an angle of 0 degree angle with a spacing of 4mm.

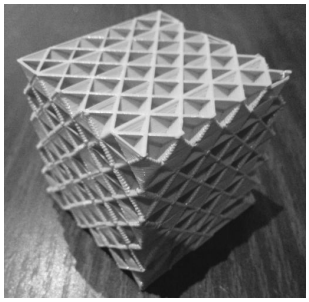
Make sure the 4mm spacing can be adjusted from a variable.

4.2 Progressive offset in the cube

We are going to fill a cube of dimensions $36 \times 36 \times 36$ mm

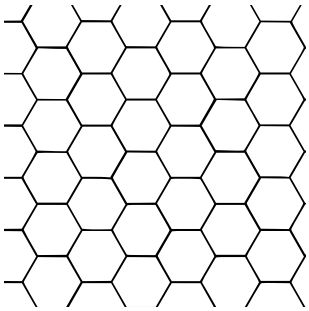
Progressively offset the lines at each layer, moving them sideways (to their right) by half a nozzle (typically 0.2mm). Generate GCode. You should now see three sets of angled walls forming closed 3D cells. For an illustration and more information refer to this URL:

`http://sylefeb.blogspot.com/2015/07/3dprint-3d-infilling-faster-stronger.html`



5 Bonus: honeycomb infill

Fill an square of 36×36 mm with a honeycomb infill:



which is a periodic tiling of regular hexagonal polygons with circumradius 3mm.

6 Miscellaneous: sample code C++

```
#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
```