Initiation to 3D Printing – Practical exercises

Sylvain Lefebvre and Camille Schreck, ENSEM 2021

Friday 20th January, 2023

1 Important information

- I recommand to write the code in C++ (the templates and corrections will be in C++). But you can also use C, Python, or JAVA.
- At the end of the session, send the **code and GCode of exercises 4, 5, 6 and 7**. The files should be in a single folder called **TP1_[nom][prenom]** and compressed into a ZIP (or tar.gz..) file to:
  - sylvain.lefebvre@inria.fr
  - camille.schreck@inria.fr

  with the mail subject **ENSEM: TP 1 [nom][prenom]**

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: A square

Write a GCode that prints a square in vrprinter (see links). For now we will not worry about printer setup (nozzle and bed temperature) but we will simply move the print head and push filament.

Recall that the main instruction for motion is: `G1 X10.0 Y20.0 Z0.2 E0.543 F1200` where X,Y,Z and E are respectively the X,Y,Z axis and E the filament axis. F is the speed in *millimeters per minutes.* A typical value of F is 1200 (20 mm/sec) when printing and 3000 (50 mm/sec) when traveling. The numbers indicate which value to give to each axis. If not specified the axis remains where it was before. All values are in millimeter.[1]

Recall that the E axis is in absolute value,[2] that is the printer unrolls the filament in absolute coordinate:

```
G1 E1.0 ; unrolls 1 mm
G1 E2.0 ; unrolls another 1 mm
G1 E2.0 ; does nothing since we were already at 2 mm
G1 E4.0 ; unrolls 2 mm
```

Finally, it can be useful to reset the value of an axis with `G92`. In particular `G92 E0` resets the E axis, setting 0 as the current value. Now it becomes possible to do:

```
G1 E1.0 ; unrolls 1 mm
G1 E2.0 ; unrolls another 1 mm
G92 E0.0 ; reset the E axis, restarting from 0
```

[1] GCodes G20/G21 switch to respectively inches and millimeters, it is safer to call G21 once at the beginning to ensure the printer expects millimeters.

[2] This can be changed, but for these exercises we consider only absolute coordinates.

```
G1 E1.0 ; unrolls 1 mm
G1 E2.0 ; unrolls another 1 mm
```

Usually we do not push filament without moving. We push filament to deposit material along a line. For instance:

```
G1 X0.0 Y0.0 Z0.2 ; move to starting point
G1 X10.0 E1.0 ; move to x=10 while pushing up to 1 mm of filament
```

This will push one millimeter of filament along the 10mm of the line segment. All axes are interpolated such that a linear amount of material is deposited along the segment[3]

How much filament should you push? We take the idealized model that considers that we manufacture perfect rectangles along a line. For a nozzle of diameter $nw = 0.4$mm (which means a track width of nw), a layer thickness $\tau = 0.2$mm, and a segment of length L, the volume of plastic to push is $V_{track} = nw \times \tau \times L$. Now, when pushing a filament length Δ_E the pushed volume is $V_{pushed} = \Delta_E \times \pi \frac{d^2}{4}$ with $d = 1.75$mm the filament diameter. Since we want $V_{pushed} = V_{track}$, you can easily obtain Δ_E. Do the math!

4 Exercise: A square from code

1. Implement a program that outputs a file ”square.gcode” that contains the GCode producing a square.

2. Modify the program to output 25 layers of thickness 0.2mm, for an object of 5mm height in total.

5 Exercise: A cylinder from code

Write a new program that outputs a file ”cylinder.gcode”, which creates an empty cylinder of diameter 8mm and height 10mm.

6 Exercise: A regular hexagon from code

Write a new program that outputs a file ”hexagon.gcode”, which creates and empty regular hexagon with circumradius $R = 10$mm and height 15mm (see `https://en.wikipedia.org/wiki/Hexagon`).

7 Exercise: A pyramid

1. Write a new program that outputs a file ”pyramid.gcode”, which creates and empty pyramid with a squared base given a size of the base s and a height h.

2. Choose parameters such that the angle of the slope is less than 45° to avoid overhang .

8 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
```

[3] Ideally. In reality things are more complex `http://marlinfw.org/docs/features/lin_advance.html`

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

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

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