**Designing Linear Patterns for Liquid Printing**

**Step 1 - Install Python (if you haven't already)**

Download and install Anaconda:

<https://www.continuum.io/downloads>

This will give you everything you need to generate GCode and plot pictures of the patterns you’re drawing.

More experienced users will have their own Python installation – in which case they will need to make sure Numpy and Matplotlib are installed for the scripts to work.

**Step 2 - Install a text editor (if you haven't already)**

Anaconda comes with a text editor included, but I like Sublime Text (that’s where all the screenshots here come from). You can find it here:

https://www.sublimetext.com/

**Step 3 - Design Your Pattern**

You can design your pattern using any piece of software that can produce .csv files. Inkscape is a good, free one that I use in this tutorial. Adobe Illustrator works too.

First, set the document size to roughly the dimensions of the bounding box of the object you want to print. For instance, if I want to print a shape into a container of dimensions 18 x 12 mm, I'd use the following settings:

Graphical user interface, text, application, chat or text message

Description automatically generated

Then, use any of the line tools to draw your shape. The direction and order of the lines you draw are the direction and order in which the printer will print them:

Graphical user interface, text, application

Description automatically generated

Now save the shape as an .svg (like the shape "test\_squiggle.svg" in the same folder (hopefully!) as this tutorial. Any of the formats of .svg should, but you might to try different types if the convertor we use next isn't happy with your file.

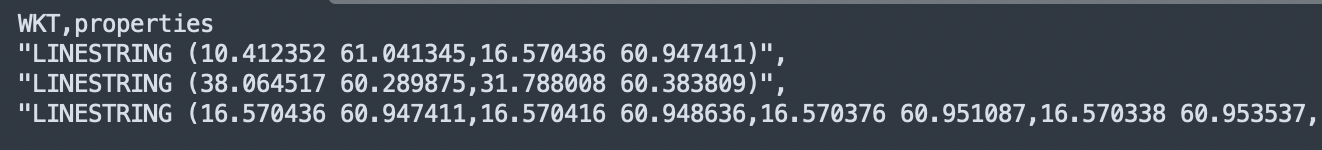
You then need to convert your .svg to a .csv. At the moment, this is a little hack-ey and involves a two-step process. First, you have to convert the .csv to a .wkt file using an online too. You can find it and log in to it here:

https://mygeodata.cloud/converter/

Username: jwforth@gmail.com

Password: printing22

Drag and drop your .csv file into the area in the browser window that says "Drag & Drop files here..." (you may need to do this in two consecutive windows) and then hit "Convert Now". Download the resulting zip file and open it up. You should get a .csv file that looks something like this:



The file consists of three lines (just like file above) comprising x- and y-coordinates. You know want to convert these coordinates into gcode. To do this, you need to run a python script, the current version of which is in the jupyter notebook "**fractal\_sliver\_v0-2.ipynb**". Open it up and let's take a look at the code. In the first box, you define some settings for your pattern:

Text

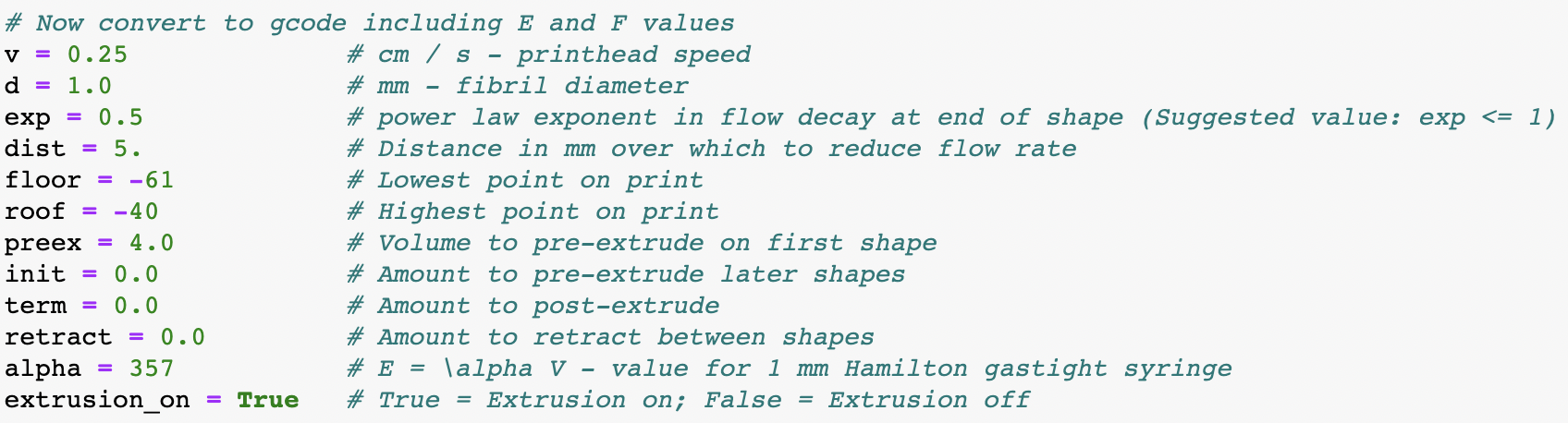
Description automatically generated

The first thing you'll want to do is **change "filename"** so that it matches the name of the pattern you've made.

x\_dim and y\_dim set the dimensions of the bounding box of the pattern you want to print in mm, and can be used to rescale the size of that pattern

inlet\_d can be set 0 for liquid-phase printing or varied accordingly if you're printing for organ-on-a-chip applications.

Once you're happy with the settings, run box 1, and let's take a look at box 2:



Again, play with these variables until you're happy - hopefully the comments are self-explanatory! Once you're happy, run this box, and you should get some output that looks like this:

Chart

Description automatically generated

If you get an error message, or you don't see a nice graph like the one above - email Joe!

You'll also see that a new file has appeared, with the suffix '.gcode'. Open it up in text editor (e.g., Sublime Text) - if you want to understand what all the different bits do, go look at Appendix 1. If you just want to get on with printing, do one quick check - scroll to the bottom of the gcode script, it should look like this:

Graphical user interface, text, application

Description automatically generated

Don't worry if the numbers aren't quite the same - what matters is that you see the line "End of Gcode". If you cant see it, just re-run the jupyter notebook script!

That's it! You're ready to start printing.

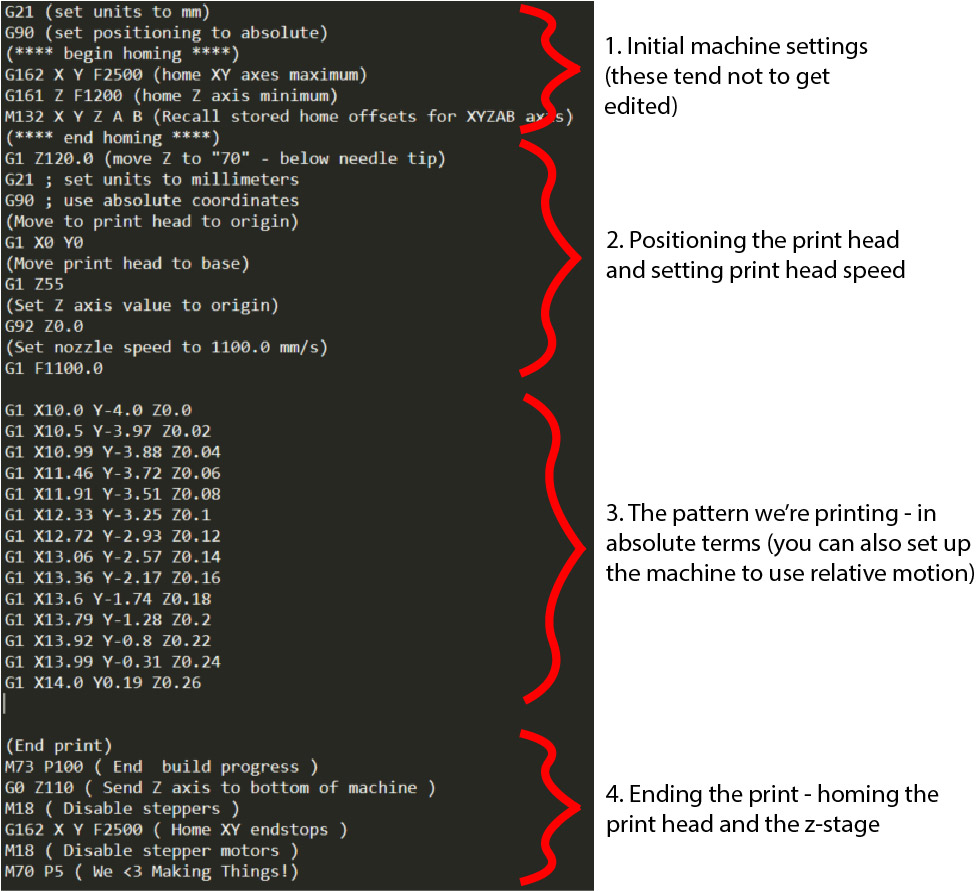
**Appendix 1 - The Anatomy of a gcode Script**

GCode is a language that contains all the information our printer needs to print out the pattern we want. It consists of commands (bits that begin with a G or an M) and comments (bits in brackets – the machine doesn’t read these – it just makes it easier for a human to figure out what the script is doing). You can find a full-ish list of these commands here:

https://reprap.org/wiki/G-code

Note that this list isn't necessarily exhaustive, not all 3D printers support all codes (in fact, most don't), and all 3D printers have their own quirks.

An annotated picture of a fairly simple GCode script is shown below:



**The Four Commands You Need to Understand:**

1. **G1 Z55**

How high (i.e., ‘z-axis value’) the print head will start printing. This tells the print head to start 55 mm above z = 0. Reduce it to lower the print head. Increase it to raise the print head. For liquid printing, it probably shouldn’t be lower than 52 if you’re using a 1.5” needle.

1. **G1 F1100**

How quickly the print head will move. This tells the print head to move at 1100 mm / s. Increase this number to make the print head move more quickly. Reduce it to make it move more slowly.

1. **G92 Z0.0**

A slightly funny one – this tells the machine that the Z position it’s at is actually what it should call ‘Z=0’ in all future commands. Generally, you shouldn’t touch this.

1. **G1 X10.0 Y-4.0 Z0.0**

This tells the print head to move from wherever it is – in this case (x, y, z) = (0, 0, 0), to x = 10 mm, y = -4 mm, z = 0 mm

**Appendix 2 - GCode: A Quick Tutorial**

This bit's still under construction - you might find you have to modify things a bit. Please update if you find something that works!

First download and install Pronterface or Octoprint - these are bits of software that will allow you to directly control your 3D printer:

http://www.pronterface.com/#download

https://octoprint.org/download/

Now all you need to do is switch on the printer, connect the printer to your laptop, open up Pronterface/Octoprint. Your computer should connect automatically to the printer - if it doesn’t, click “Connect”.

Next, copy and paste the following script into the gcode window:

G21 (set units to mm)

G90 (set positioning to absolute)

(\*\*\*\* begin homing \*\*\*\*)

G162 X Y F2500 (home XY axes maximum)

G161 Z F1200 (home Z axis minimum)

M132 X Y Z A B (Recall stored home offsets for XYZAB axis)

(\*\*\*\* end homing \*\*\*\*)

G1 Z120.0 (move Z to "70" - below needle tip)

G21 ; set units to millimeters

G90 ; use absolute coordinates

(Move to print head to origin)

G1 X0 Y0

(Move print head to base)

G1 Z54

(Set Z axis value to origin)

G92 Z0.0

(Set nozzle speed to 1000.0 mm/s)

G1 F1000.0

#########

(Enter your coordinates here)

#########

(End print)

M73 P100 ( End build progress )

G0 Z110 ( Send Z axis to bottom of machine )

M18 ( Disable steppers )

G162 X Y F2500 ( Home XY endstops )

M18 ( Disable stepper motors )

M70 P5 ( We <3 Making Things!)

In-between the rows of (‘########’), enter the co-ordinates you want the print head to move along. e.g., If you want the print head to move to the coordinates (x, y, z) = (10, -5, 7), you would enter

G1 X10 Y-5 Z7

in the empty line below where it says “(Enter your coordinates here)”. Note that these coordinates are all in mm. You should probably take some time to play around with seeing what happens when you enter more lines with more coordinates. Careful not to set Z to anything below 0, or the print head will crash into the printing stage.

**Relative Motion**

Sometimes, you might just want to tell the print head to make relative motions (i.e., move this far along x), rather than telling it which coordinates to move to (i.e., move to x = 5). To do this in a way that’s vaguely useful, first run the following script:

G21 (set units to mm)

G90 (set positioning to absolute)

(\*\*\*\* begin homing \*\*\*\*)

G162 X Y F2500 (home XY axes maximum)

G161 Z F1200 (home Z axis minimum)

M132 X Y Z A B (Recall stored home offsets for XYZAB axis)

(\*\*\*\* end homing \*\*\*\*)

G1 Z120.0 (move Z to "70" - below needle tip)

G21 ; set units to millimeters

G90 ; use absolute coordinates

(Move to print head to origin)

G1 X0 Y0

(Move print head to base)

G1 Z54

Now, delete the above script from the window, and enter:

G1 X10 Y-5 Z7

Rather than telling the print head to move to the co-ordinates (x, y, z) = (10, -5, 7), you’re telling to print head to move 10 mm along the x-axis, -5 mm along the y-axis, and 7 mm along the z-axis. Playing around with the numbers above will move the print head whatever distance you enter as the numbers. Careful not to use a large, negative value for ‘Z’, or the print head will crash into the printing stage.