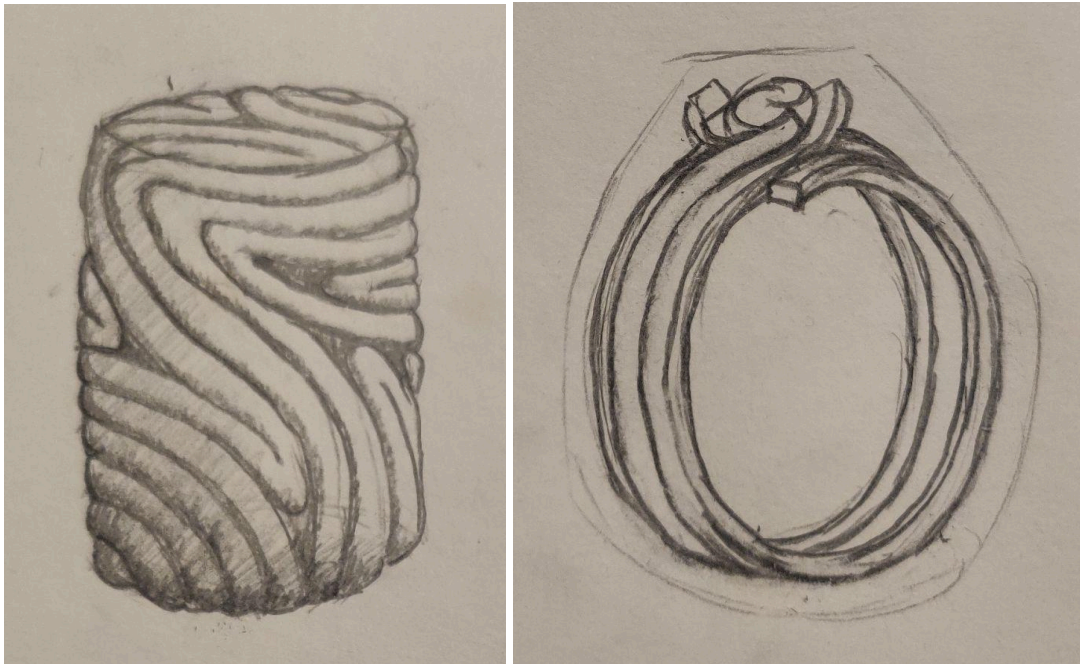


This program randomly generates infills for any object. Given the shell of any object, infills will be generated that follow a smooth swirling pattern that looks organic and aesthetically appealing. These objects can then be fabricated via 3d printing to create both decorative and functional objects with unique designs.

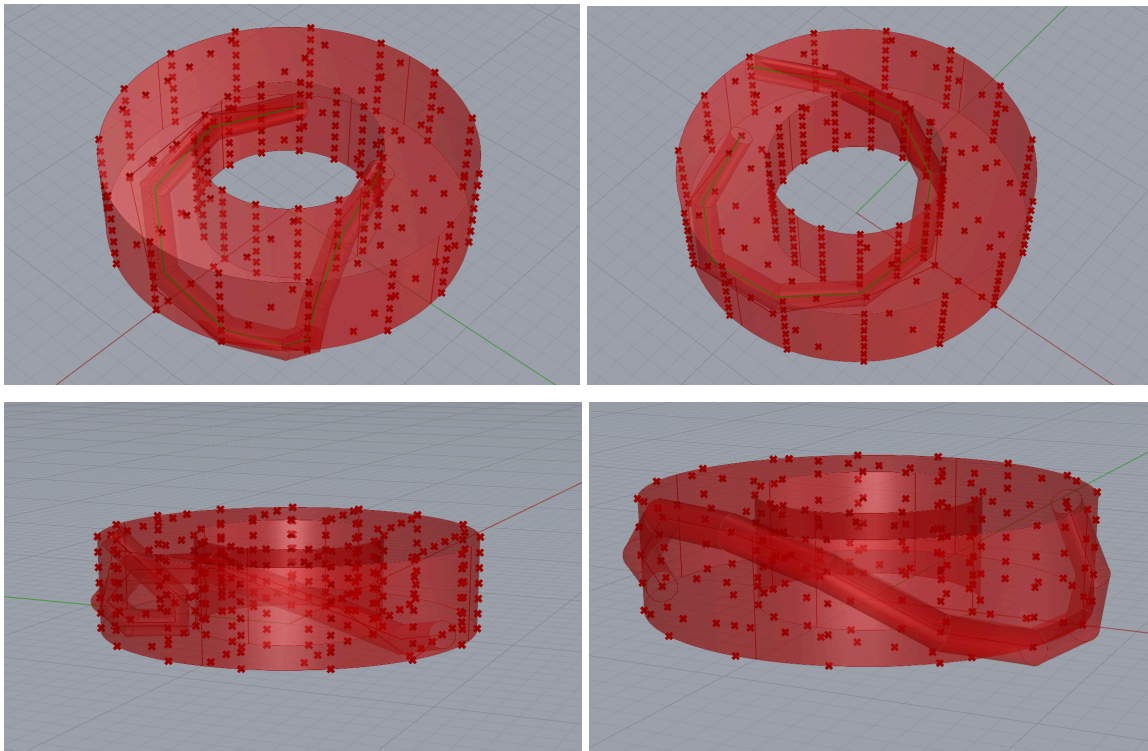
Some initial concept sketches:



The project mainly consists of a single python file meant for use in Grasshopper. It takes 5 inputs: points, max\_len, max\_dist, alpha, and regen. The first five are parameters used for the algorithm within the python file, while regen is simply an input to regenerate the output. 'Points' are the points that will be used to generate the infill, max\_len is the maximum number of points allowed per line segment, max\_dist is the maximum distance between each point in the line. The alpha parameter determines whether the lines follow the direction of velocity or acceleration more. That is, if alpha is high, the lines will favour moving straight, while alpha being low means the line is likely to continue any curve it is currently making.

The concept behind the code is fairly simple. Given a number of points (presumably on the surface of an object, though not necessary), select a random point to start from and a second nearby ( $< \text{max\_dist}$ ) point as our second point. Each additional point is determined by getting the vector between the current point and the last point ('velocity' in code), and the difference in velocity for the last two transitions ('acceleration'). The velocity and acceleration are then combined with the formula  $\alpha \cdot \text{velocity} + (1 - \alpha) \cdot \text{acceleration}$ , and that determines the ideal direction for the next point. We then find the nearest point to that direction within max\_dist using cosine distance. This process is repeated until the line segment hits max\_len length.

A few example outputs are as follows:



It is intended that multiple line segments will be used to fill the space, but currently there are issues with overlapping segments, which will be solved later.

