“3D Turtle Graphics” by 3D Printers

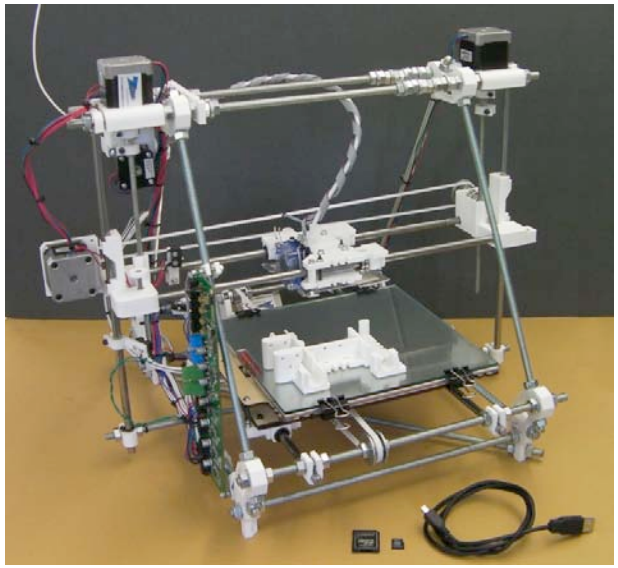
Yasusi Kanada
Dasyn.com

Introduction: Conventional 3D Object Design and Printing

► FDM-type 3D printers

- There are many types of 3D printers, but a popular cheap one is called fused deposition modeling (FDM) type.
- FDM-type printers extrude melted plastic from the nozzle and solidify it.

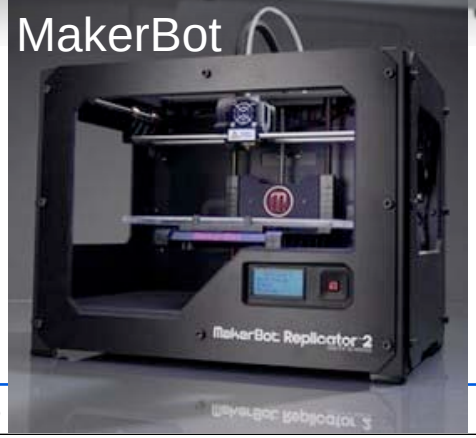
Reprap



http://www.3dfuture.com.au/2012/04/reprap-full-kit-available-for-780/

Stratasys FDM®

MakerBot



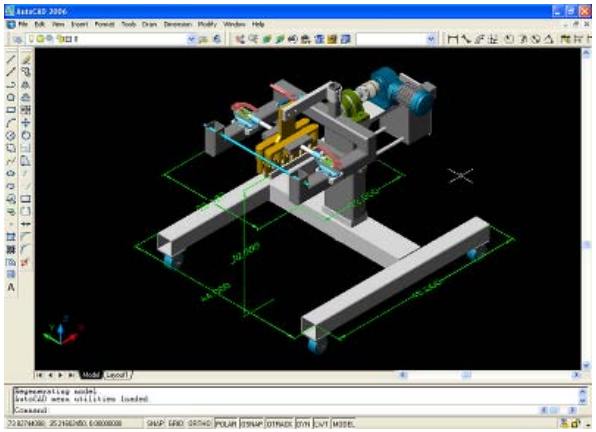
Rostock MAX

Introduction: Conventional 3D Object Design and Printing (cont'd)

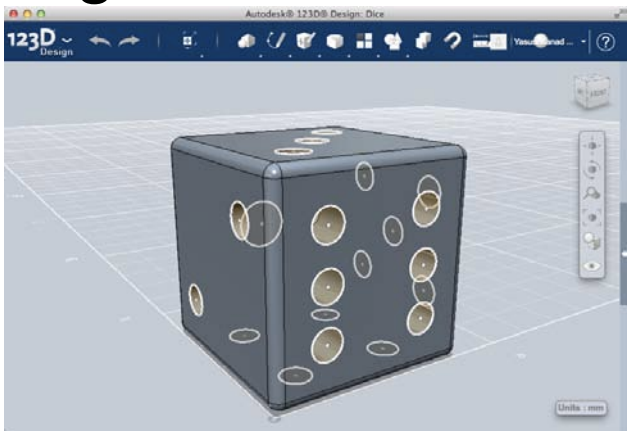
- **Outline of process for 3D object design and printing**
  - Step 1: Design by a 3D CAD tool and generation of (conversion to) a STL file
    - STL means Standard Triangulation Language or Stereo-Lithography.
  - Step 2: Decomposition into layers by using a slicer (software)
  - Step 3: Printing layer by layer by a 3D printer
- **This process is explained in detail below.**

[Step 1] Design by 3D CAD tools and output as STL files

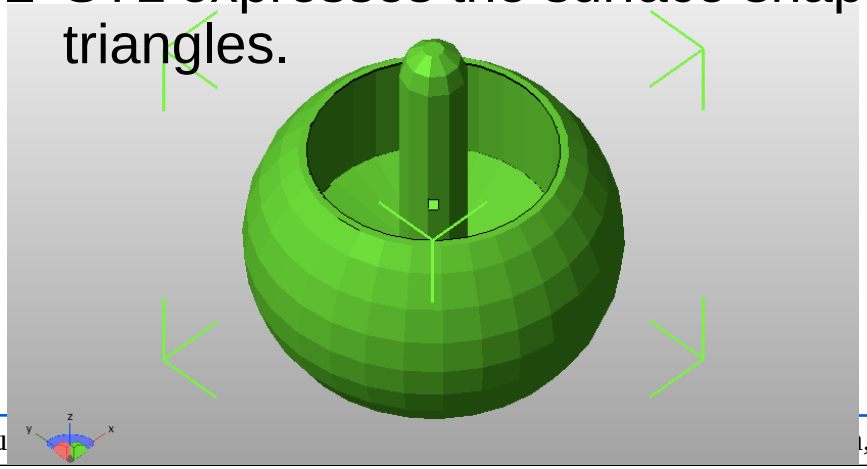
- **Conventionally designed by 3D CAD and expressed in a declarative language for describing 3D models.**



Autodesk
(CAD software)



- **3D models converted to STL, a standard declarative language.**
  - STL expresses the surface shape of objects by a collection of triangles.



```
solid tippetop_1piece_v05-repaired
 facet normal -0.988014 0.103855 -0.114202
  outer loop
   vertex -16.500000 0.000000 -7.935401
   vertex -16.139400 3.430540 -7.935400
   vertex -15.786800 3.355581 -11.054070
  endloop
 endfacet
 facet normal -0.988010 0.103854 0.114240
  outer loop
   vertex -16.500000 0.000000 -7.935401
   vertex -16.139400 0.000000 -4.816731
   vertex -16.139400 3.430540 -7.935400
  endloop
 endfacet
 ...
endsolid tippetop_1piece_v05-repaired
```

[Step 2] Slices and G-Code

- **Sliced objects are usually represented by a procedural language for CAM, which is called G-Code.**
  - G-Code is originally designed for programming the behavior of machine tools (bites etc.).
  - G-Code specifies the motion of a print head, the volocity of extruding plastic, and so on.
- **Command examples**
  - G0: simple tool motion
    - G0 X0 Y0 Z0 F3600: move to coordinate (0, 0, 0) at 3600 mm/min
  - G1: tool motion with milling (printing)
    - G0 X0 Y0 Z0 F3600 E100: move with extruding filament specified by E100

[Step 3] Printing 3D Objects

- **A print head only move horizontally unless transiting to the next layer.**
  - Because it normally prints horizontally-sliced layers one by one.
- **By using G-Code, however, the head can move toward any direction.**

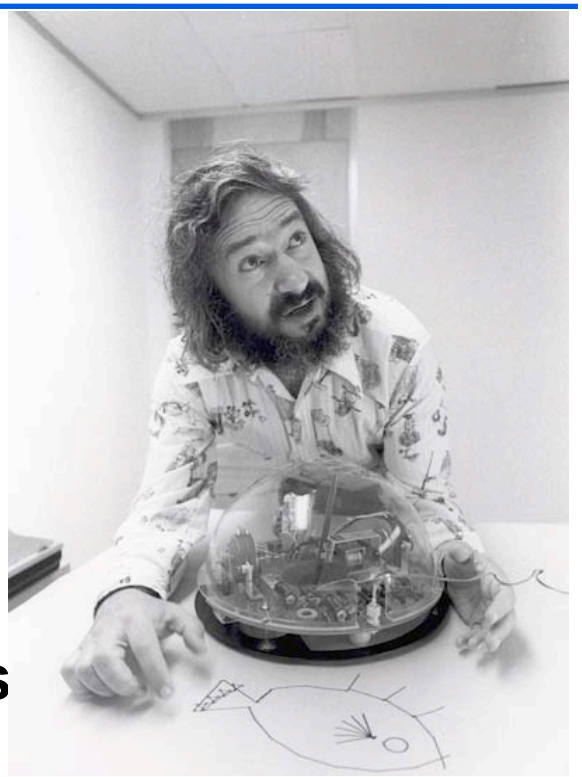
G0 X1 Y1 **Z1**

(X0, Y0, **Z0**) → (X1, Y1, **Z1**)

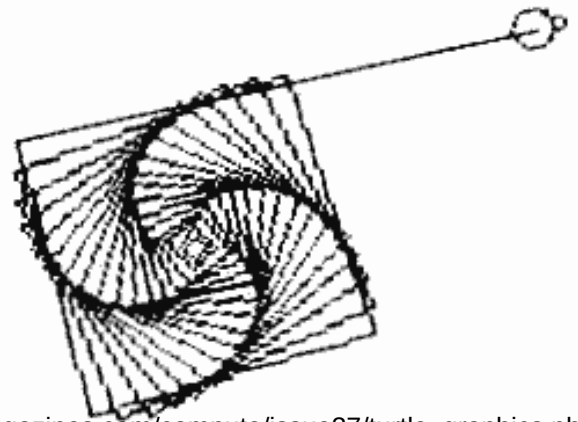
  - Although many 3D printers are not good in vertical motion.

Turtle Graphics

- **Seymore Papert and Logo**
  - Papert designed Logo programming language for children.
  - By using Logo, 2D line art can be drawn by a “turtle” — (2D) turtle graphics



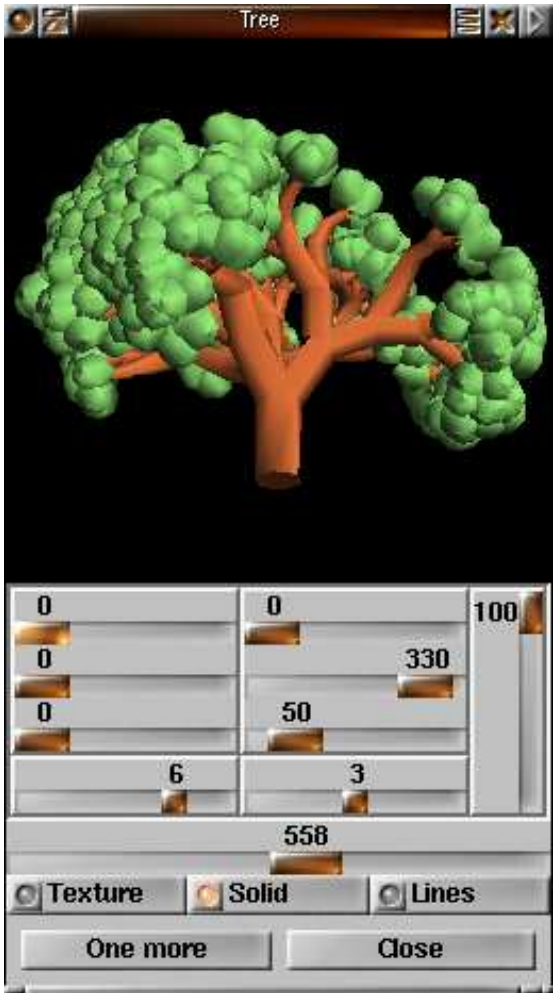
- **Drawing commands for turtle graphics**
  - Forward d: move forward by d.
  - Left a: turn left by a (degrees).
  - Right a: turn right by a (degrees).



http://www.atarimagazines.com/compute/issue37/turtle_graphics.php

3D Turtle Graphics

- **Basics**
  - forward, left, right
  - **up**, **down**
- **Extended 3D turtle graphics**
  - Bernd Paysan, “Dragon Graphics”



Paysan, B., ““Dragon Graphics” Forth, OpenGL and 3D-Turtle-Graphics”, 2009, http://bigforth.sourceforge.net/dragongraphics-eng.pdf

“Turtle Graphics” by 3D Printing

- **Drawing commands are translated into G-Code**
  - Forward → G1 (moving while printing)
- **Turtle coordinates are converted to Descartes coordinates**
  - The direction of turtle is memorized by the G-Code generating program.
  - Forward → The next coordinates are calculated from the current coordinates and direction.
  - Left, Right → The direction is updated.
- **Selection of a turtle coordinate system**
  - Polar coordinates
    - Turtle direction is arbitrary in 3D.
    - Coordinates of flight simulators are polar.
    - Difficult to guarantee printability because difficult to know the gravity direction.
  - Cylindrical coordinates
    - Turtle direction is fixed to be horizontal.
    - Vertical displacement is recorded.
    - Probably easier to be designed.

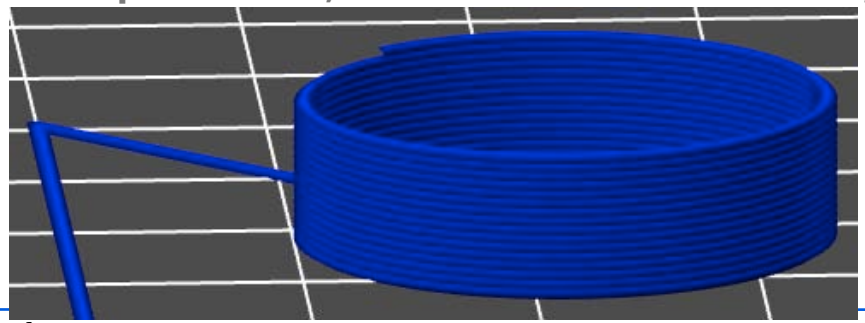
Difference between 3D Printing and 3D Graphics

- **It is difficult to print on the fly by 3D printing.**
  - Lines can be drawn freely at any location in the 3D space by 3D graphics.
  - While printing on earth, printing on the fly is difficult because extruded filament must be supported.
- **Printing velocity and amount of filament extrusion must be controlled (are controllable).**
  - Thickness of lines can be controlled in 3D graphics.
  - Specialized techniques, which are unnecessary in 3D graphics, are required for pretty 3D printing.
    - E.g., the printing velocity must be regulated.

Alternatives and Library Development

- **Alternatives**
  - To design a language similar to Logo
  - To develop libraries for conventional languages
    — this is easier and has more potential to be popular
- **Selection: A library for Python, turtle.py, was developed.**
  - Cylindrical coordinates are used.
  - Moving forward with moving up or down by forward(r, z).
  - Provided as open-source software: It is for Rostock MAX, so a slight modification is required for other printers.
  - Example: helix

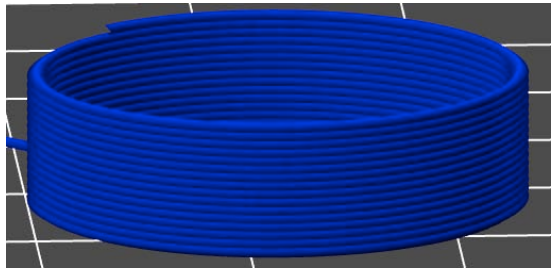
```
turtle.init(FilamentDiameter,
            HeadTemperature, BedTemperature, defaultCrossSection,
            x0, y0, 0.4);

dz = 0.4 / 72;
for j in range(0, 16):
    for i in range(0, 72):
        forward(1, dz);
        left(5);
```

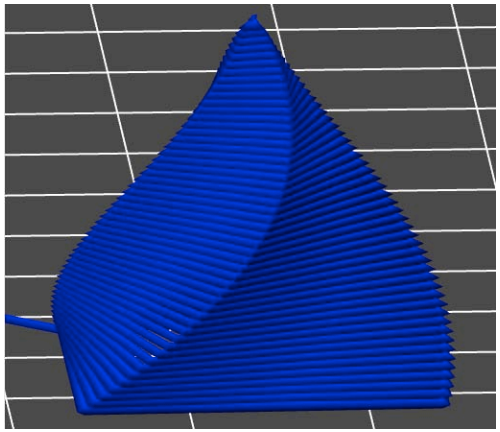


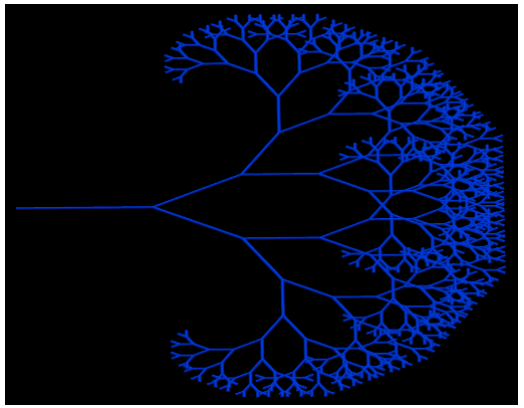
Experiments: Method

► Following examples were tested.

- Cylinder (helix)
- Skewed square piramid
- Flat fractal





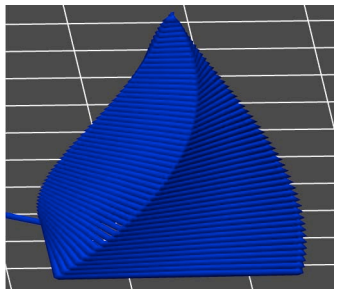


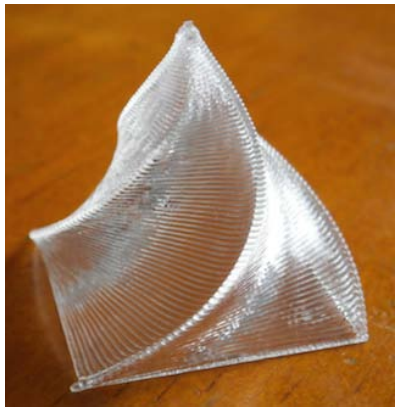
3 次元フラクタルは困難

► Repeated the following steps until succeeded.

- Program description
- Confirmation by graphics
- 3D printing

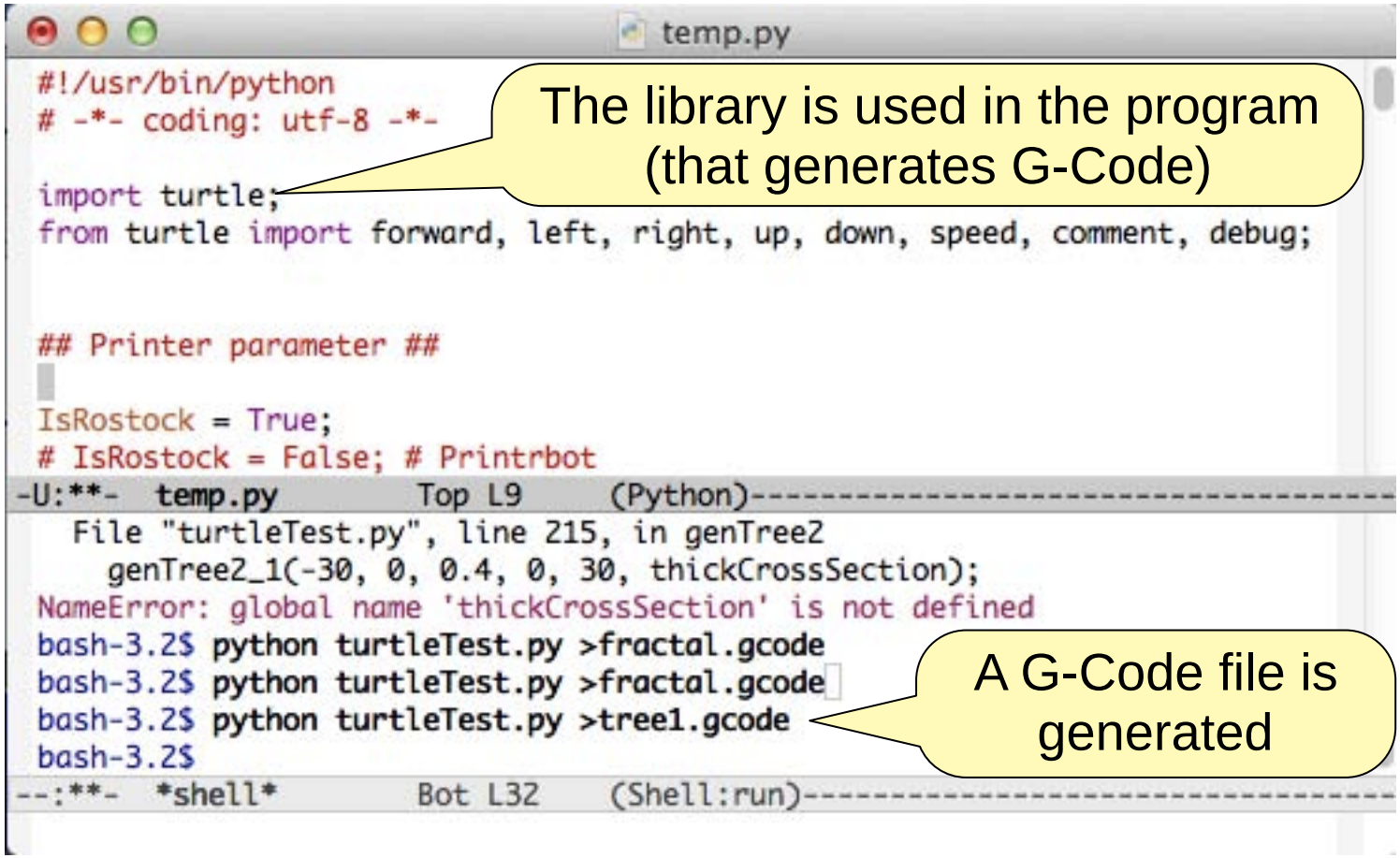
```
temp.py
#!/usr/bin/python
# -*- coding: utf-8 -*-

import turtle;
from turtle import forward, left, right, up, down, speed, comment, debug;


## Printer parameter ##

IsRostock = True;
# IsRostock = False; # Printrbot
-U:**-  temp.py        Top L9     (Python)----------------------------------
   File "turtleTest.py", line 215, in genTree2
     genTree2_1(-30, 0, 0.4, 0, 30, thickCrossSection);
NameError: global name 'thickCrossSection' is not defined
bash-3.2$ python turtleTest.py >fractal.gcode
bash-3.2$ python turtleTest.py >fractal.gcode
bash-3.2$ python turtleTest.py >tree1.gcode
bash-3.2$
--:**-  *shell*        Bot L32    (Shell:run)----------------------------------
```





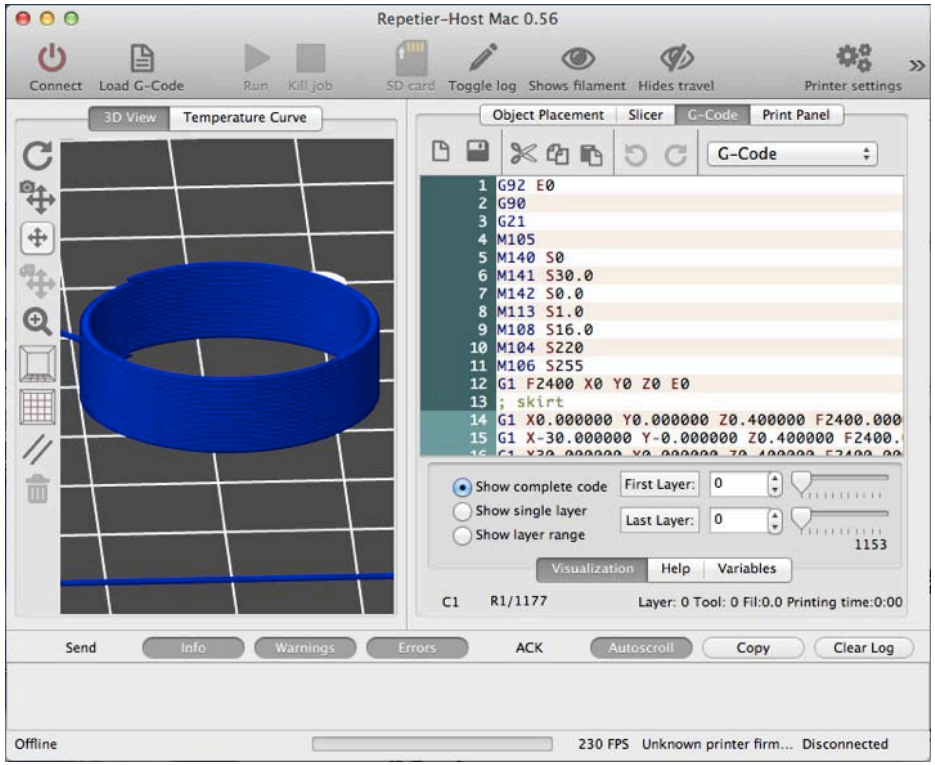
Program Description

► Programming and debugging using editors / IDE



Confirming the Production by Graphics

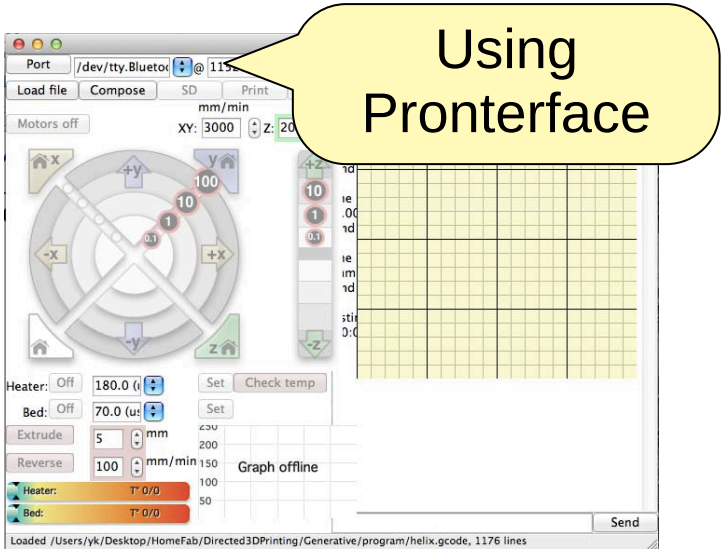
- **The G-Code file is graphically displayed by a 3D printing tool called Repetier-Host.**

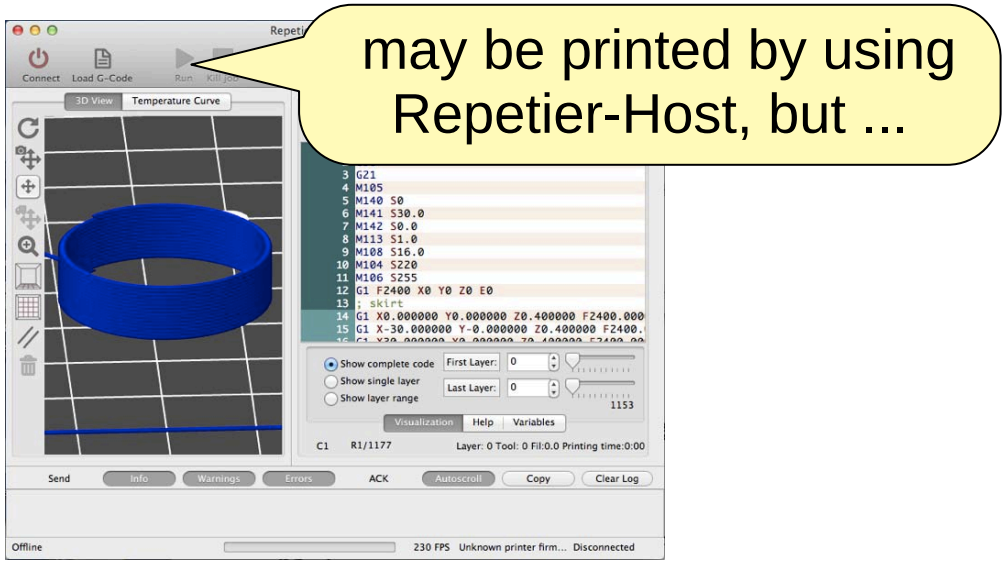


- **Visually confirm whether it will be printed well.**
  - The tool checks only weak conditions (such as syntactic correctness of commands)
  - Human beings may however misjudge the printability.
    - Because printing process is dynamic but graphics is static.

Printing Productions

- **By using a 3D printing software, specify a G-Code file and print it.**



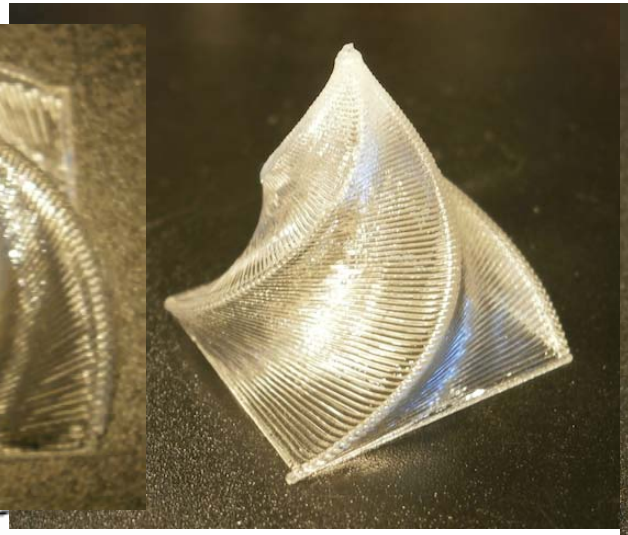


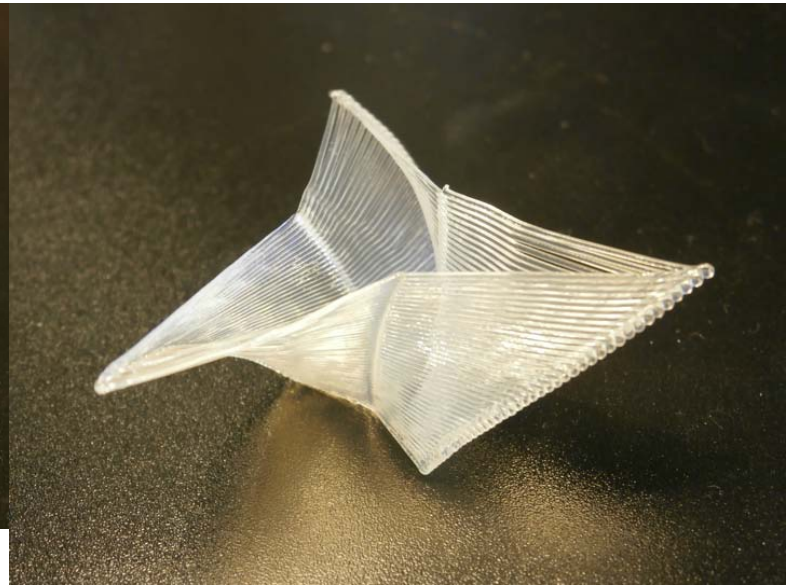
Printing Process Example

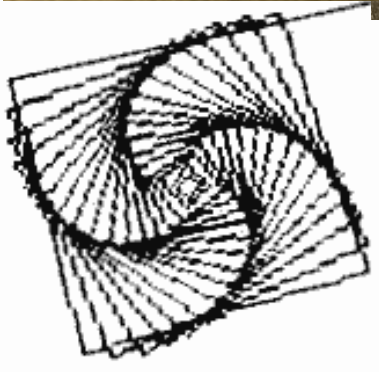
youtu.be/7H5-acxQ_RE

Production (Printing Results)

- **Rotation and enlarging/reducing**



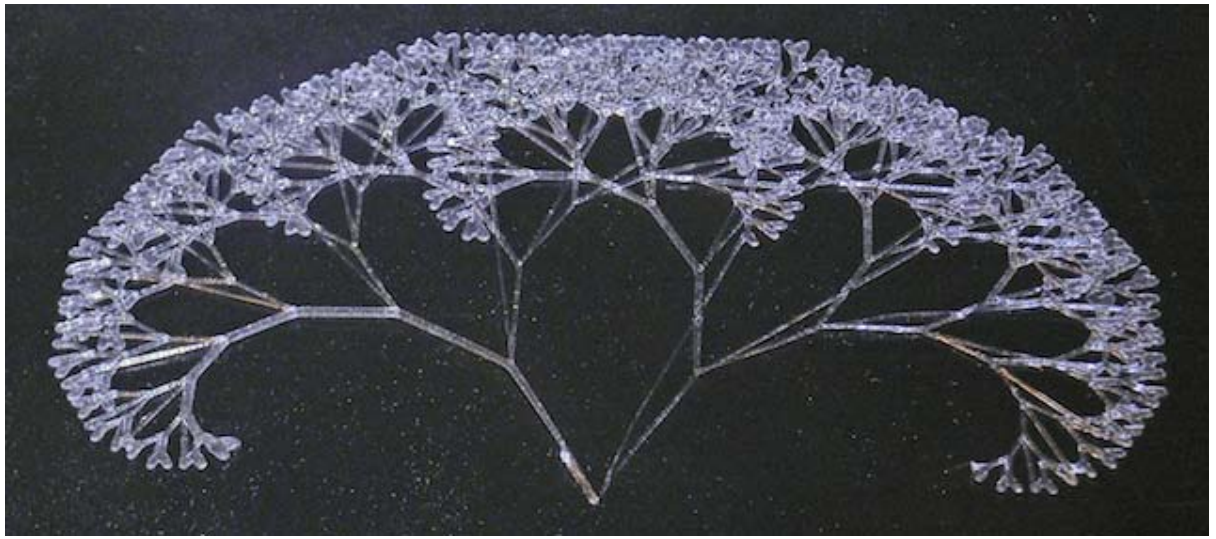




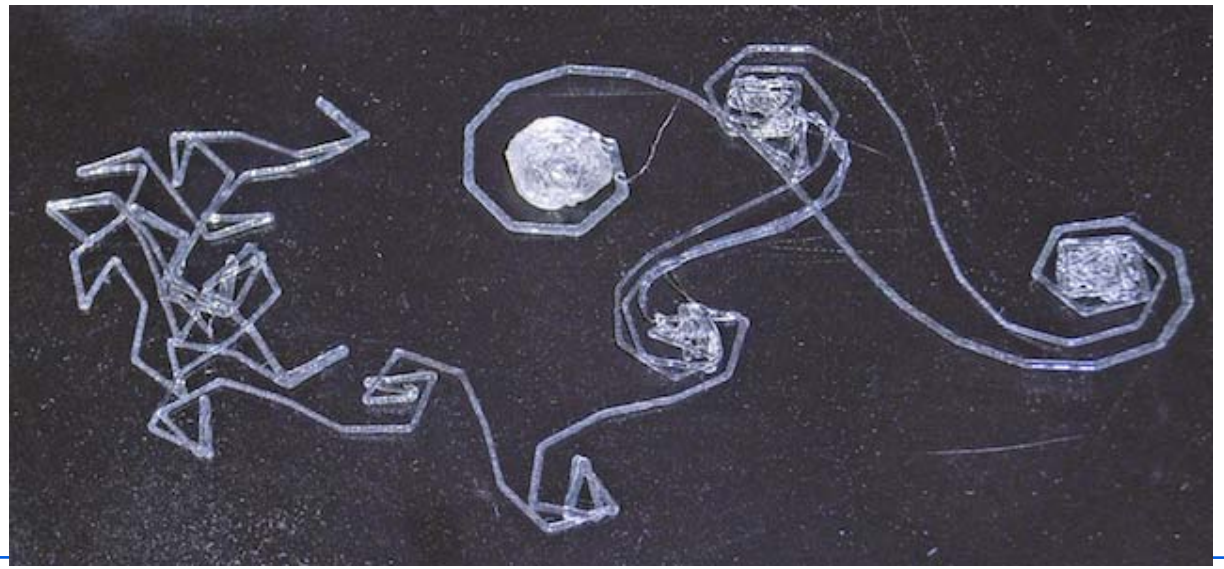
Failure

Production (Printing Results) (cont'd)

► 2D fractal figure

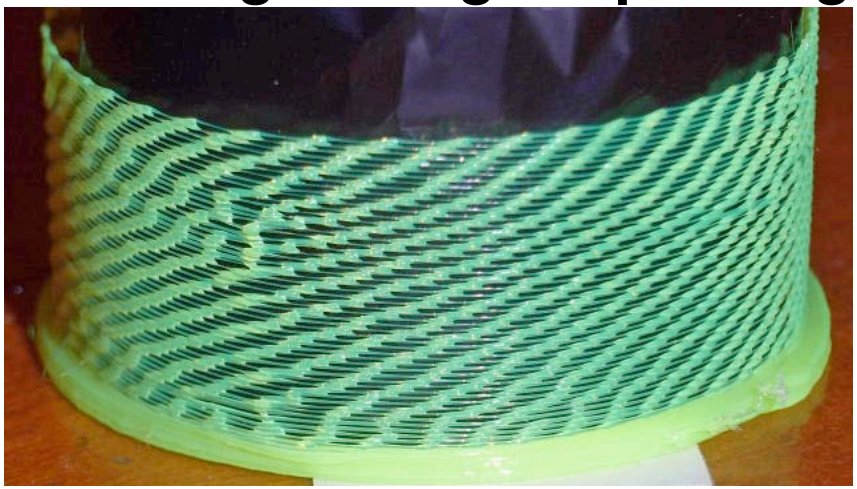


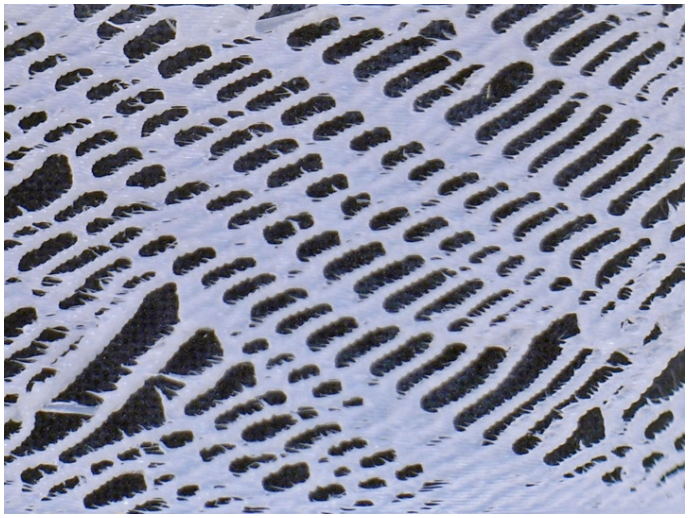
► Other 2D figure



Bonus: Productions of Methods Other Than Turtle Graphics

► Self-organizing 3D printing





Kanada, Y., "3D Printing and Simulation of Naturally-Randomized Cellular-Automata ", *19th International Symposium on Artificial Life and Robotics* (*AROB 2014*), 2014-1

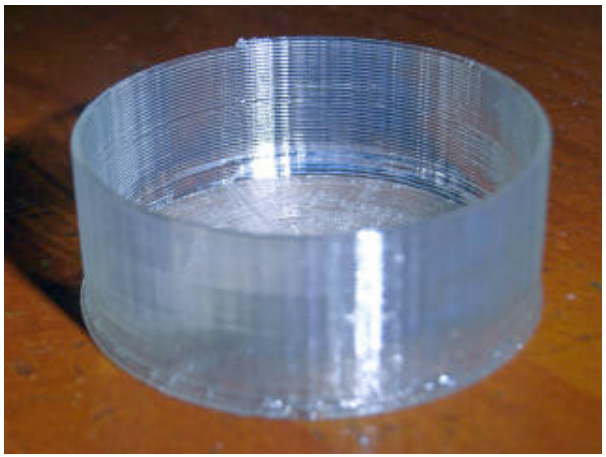
Kanada, "FDM 3D-printing as Asynchronous Cellular Automata", Y., *8th International Workshop on Natural Computing* (*IWNC8*), 2014-3.

Bonus: Productions of Methods Other Than Turtle Graphics (cont'd)

- **Direction-specified printing**

Kanada, Y., "Method of Designing, Partitioning, and Printing 3D Objects with Specified Printing Direction", *2014 International Symposium on Flexible Automation (ISFA 2014)*, 2014-7.

- **Shape generation by parts assembly and deformation**



Conclusion

► Conclusion

- A Python library for 3D turtle graphics was developed.
- An environment for 3D turtle graphics can be built by combining this library and G-Code tools, 3D printers, and so on.

► Future work

- To try various shapes and support methods.
  - Devices with the current library.
  - Extensions of the library.
- To try polar coordinates.

- The URL for this slides and the paper: http://bit.ly/1q7OWe6 (http://www.kanadas.com/papers-e/2014/08/3d_turtle_graphics_by_3d_print.html)