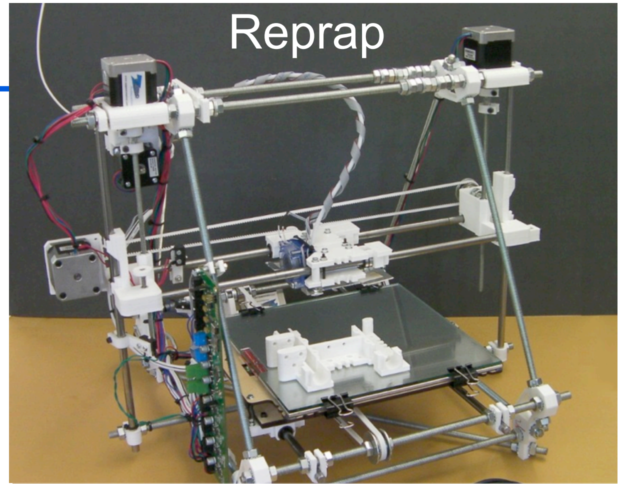
FDM 3D-printing as Asynchronous Cellular Automata

Yasusi Kanada

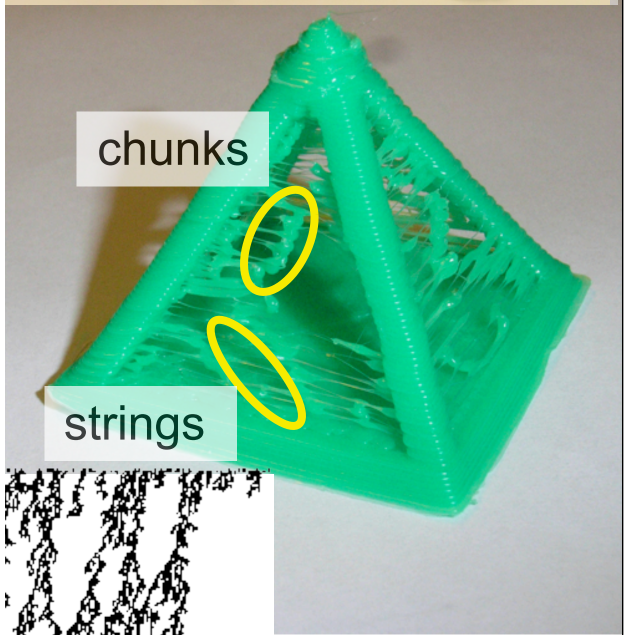
Dasyn.com

Introduction

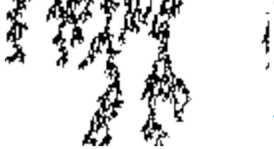


http://www.3dfuture.com.au/2012/04/reprap-full-kit-available-for-780/

- **3D printing** (or Additive manufacturing)
  - Objects are designed by using 3D CAD.
  - 3D objects are printed layer by layer.
  - Cheap FDM 3D printers are widely used. (FDM means fused deposition modeling)
- **3D printers can generate "naturally-randomized" self-organized patterns.**
  - Printing conditions and process including nozzle temperature, extrusion process, air motion, etc., are fluctuated.
  - *Chunks* and *strings* are generated.

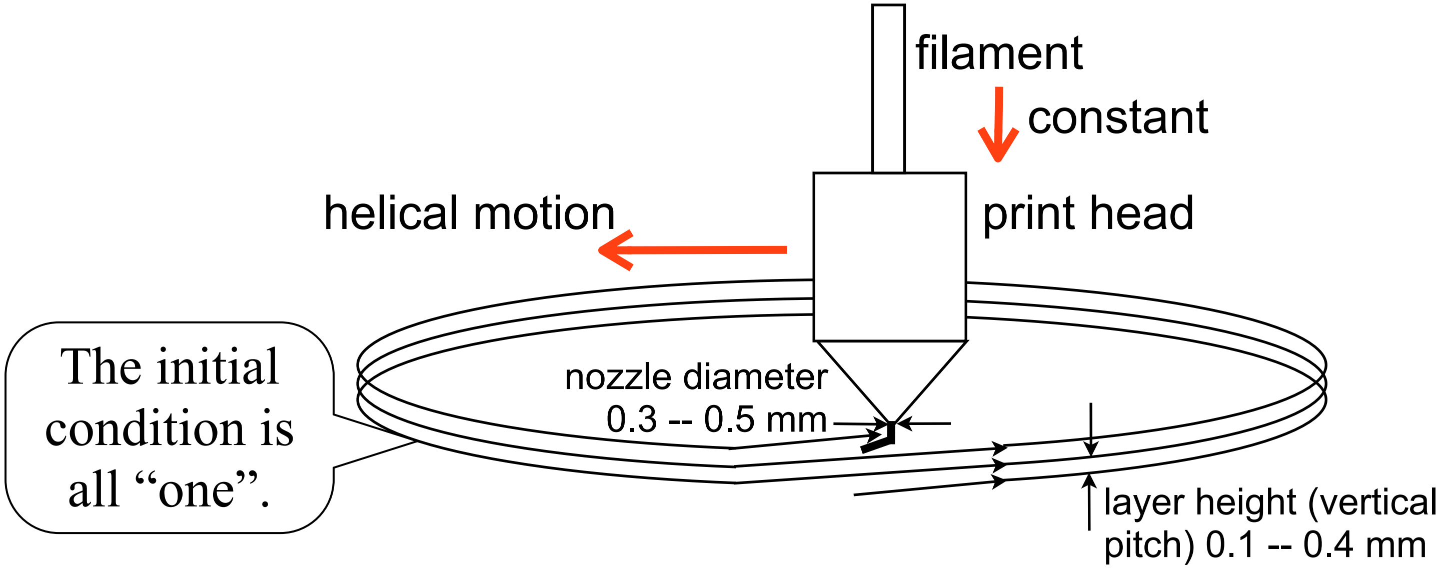


- **FDM 3D-printing can be interpreted as asynchronous CA** (cellular automata)**.**
  - A printing head generates 1D on/off patterns.
  - Fluctuated patterns are similar to patterns generated by stochastic CA.



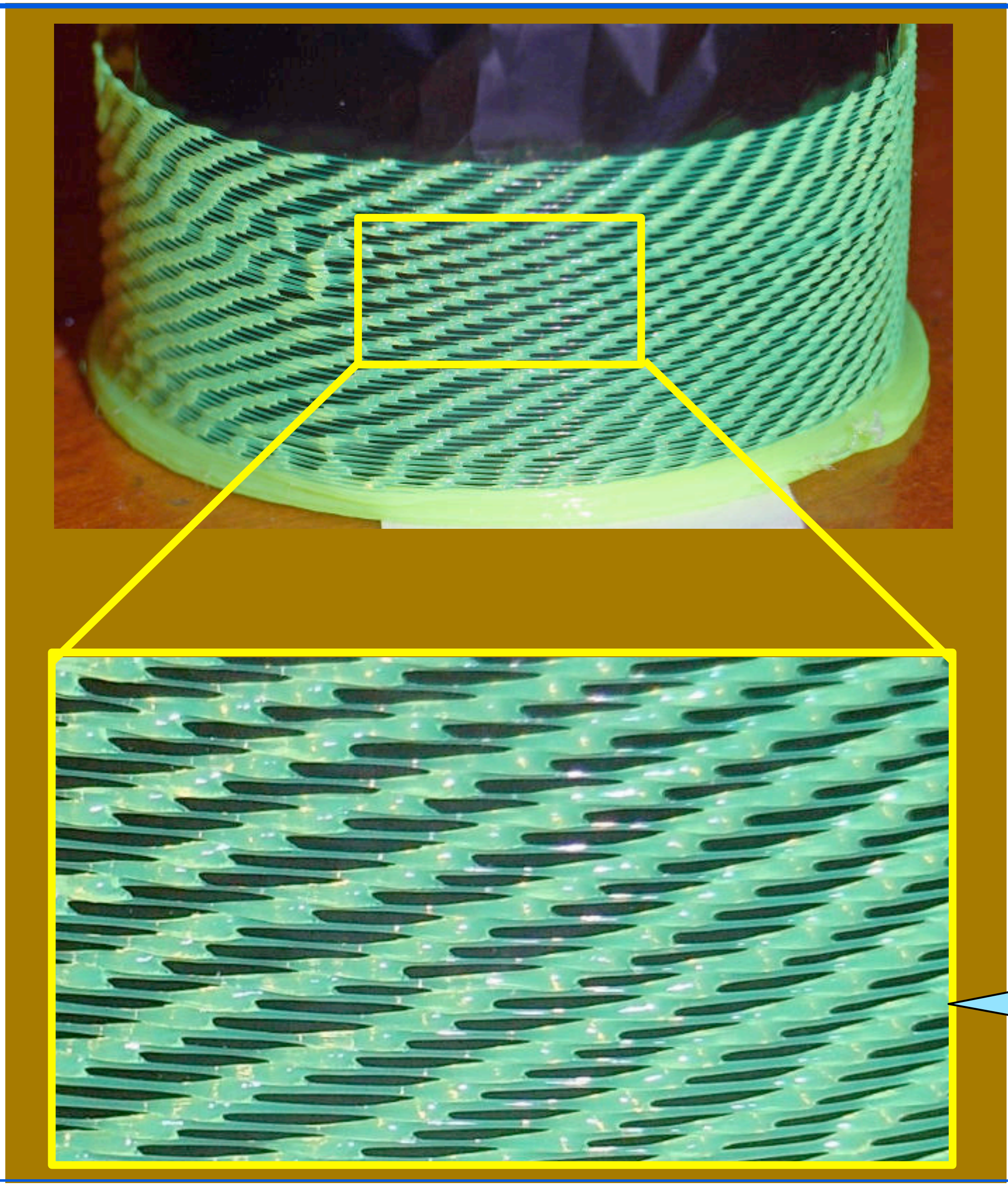
Methods for 1D CA pattern generation

- **1D CA-like patterns are generated by a helical motion.**
  - No 3D CAD/CAM program prints in this way!
- **No explicit randomization is introduced.**
  - Velocity of filament extrusion is constant and small.



- **This method was originally proposed by the previous paper.**
  - Yasusi Kanada, “3D Printing and Simulation of Naturally-Randomized Cellular-Automata”, *19th International Symposium on Artificial Life and Robotics* (*AROB 2014*), 2014-1.

Typical printed patterns -- Stripes

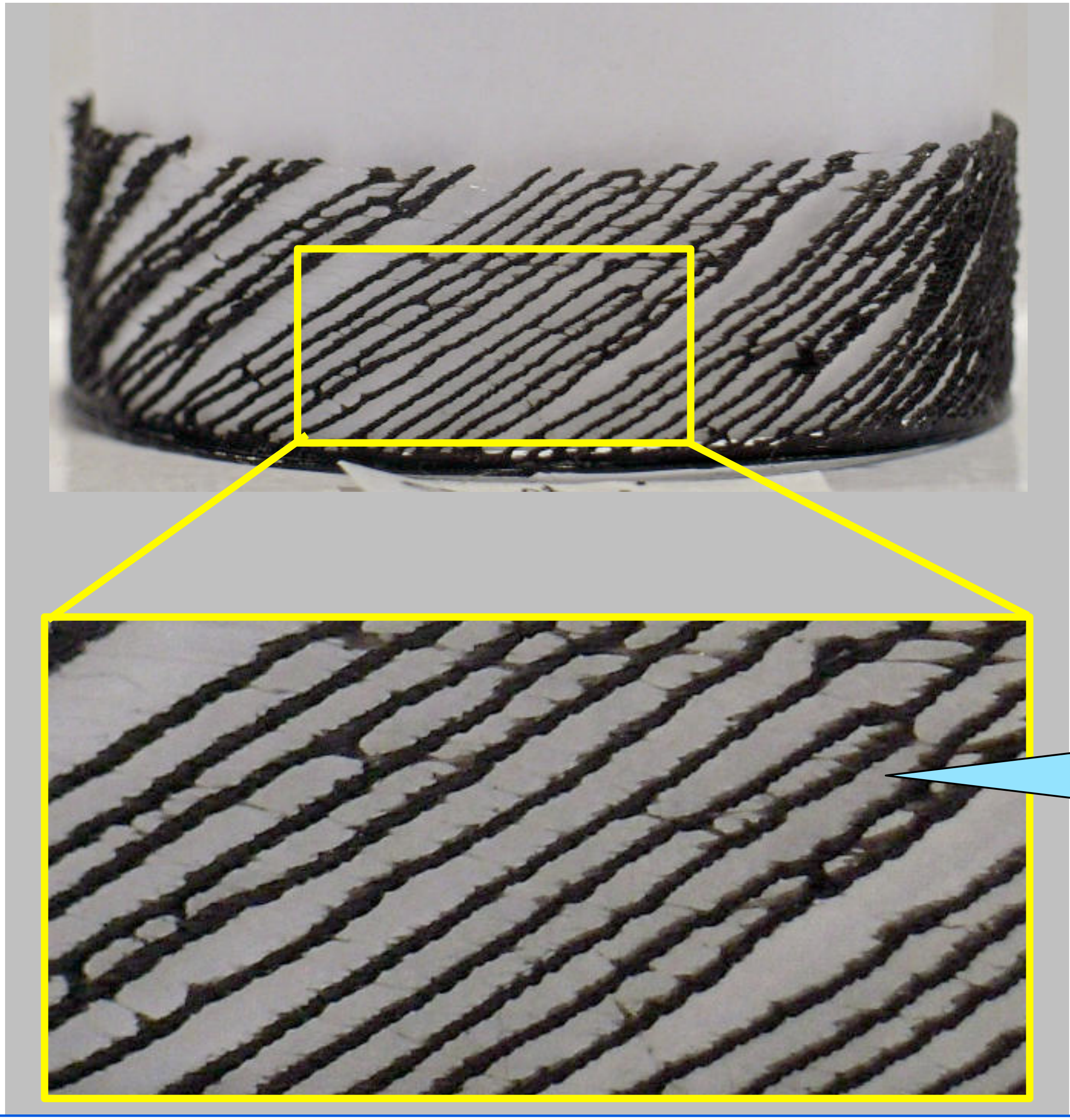


The layer is thick (0.3 mm).
The filament cross section is $0.045\ mm^2$.
PLA.

Stacked chunks and strings can be seen.

More uniform pattern

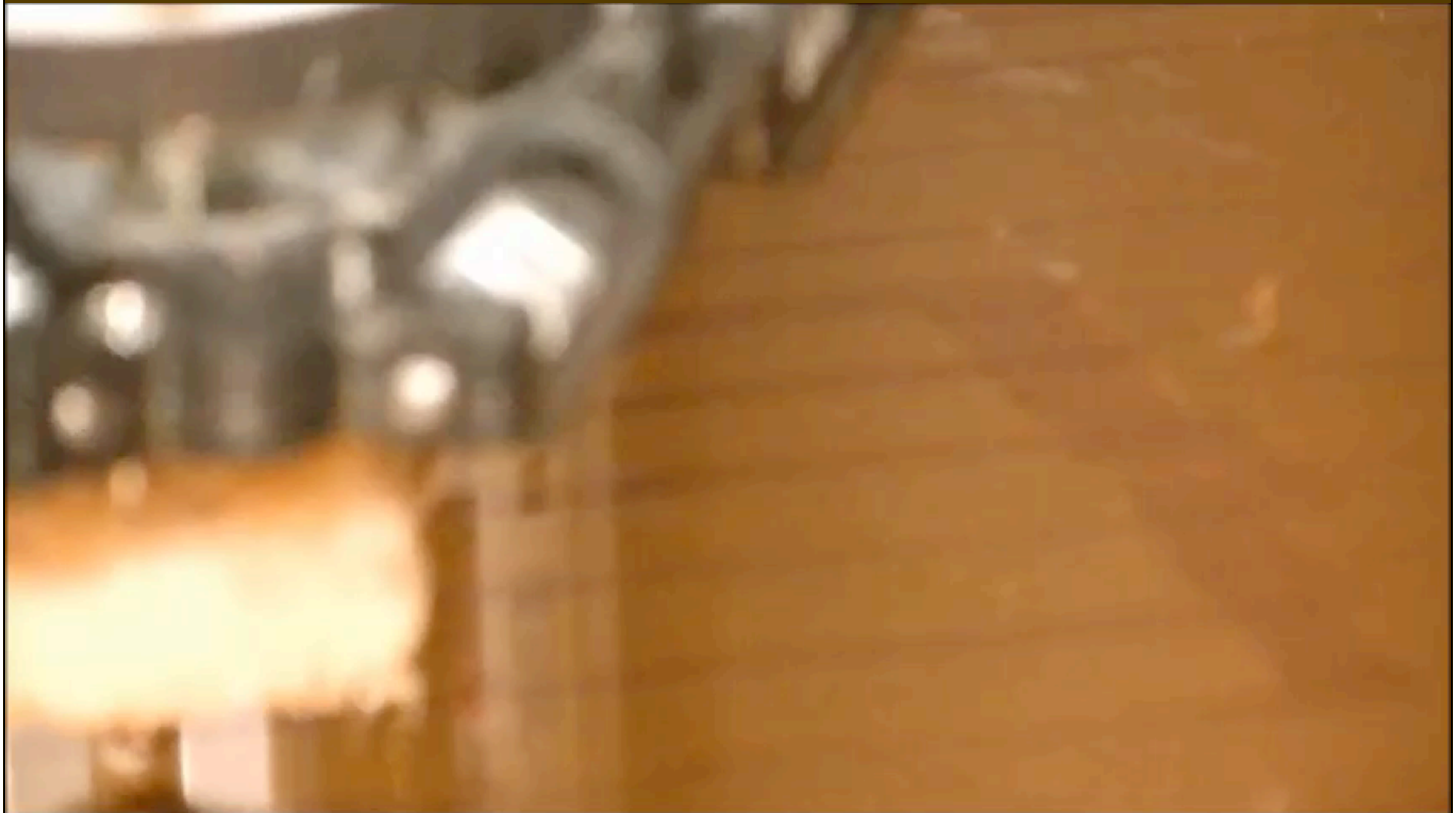
Typical printed patterns -- Stripes (cont'd)



The layer is thin (0.1 mm).
The filament cross section is $0.02\ mm^2$.
PLA.

Less uniform pattern
(Strings are torn)

Printing process using Rostock MAX 3D printer

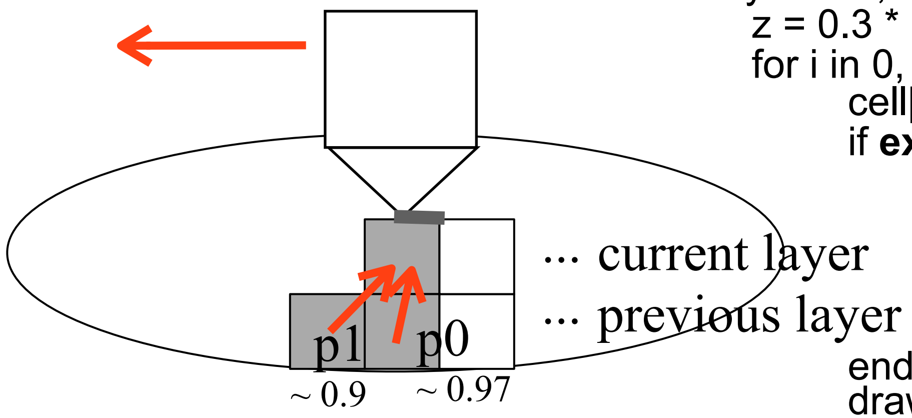


http://youtu.be/IJ15ysJR5l8

Basic computational model

- ► **A computational model that simulates printed 1D patterns (chunks only) was developed.**
  - **Explicit randomization** was (random numbers were) introduced instead of "natural randomization".
  - A pattern is generated by using a probabilistic rule:

    if extruded filament >= 1 /* certain amount */ then
        if cell[*l*–1][*i*] = 1 then cell[*l*][*i*] = 1 & filament cleared at probability p0
        else if cell[*l*–1][*i*+1] = 1 then cell[*l*][*i*] = 1 & filament cleared at probability p1
        else cell[*l*][*i*] = 0
    else cell[*l*][*i*] = 0

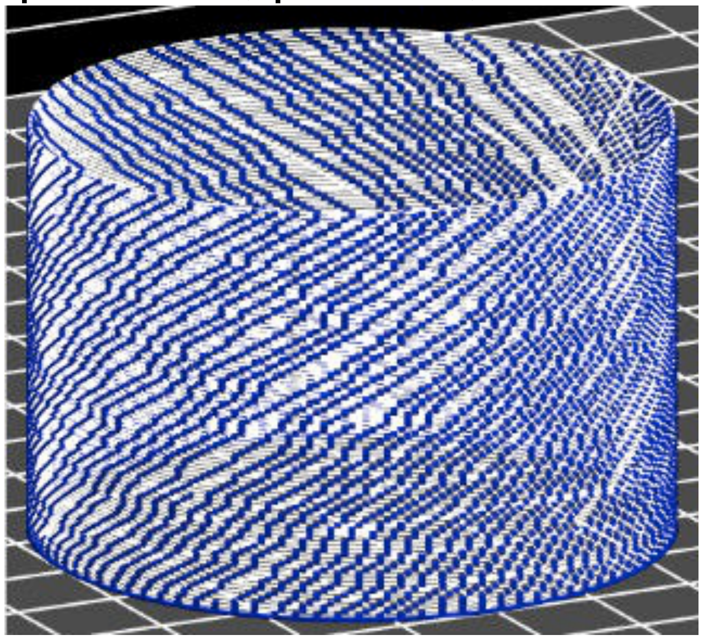



```
extrudedFilament = 0;
for layer in 1, 2, …, layers loop   // repeat for all layers
    z = 0.3 * layer;                // layer pitch is 0.3
    for i in 0, 1, …, 4 * 72 loop   // repeat for all parts of a circle
        cell[layer][i] = 0;         // clear pattern
        if extrudedFilament >= 1 then
            if cell[layer–1][i] > 0 and random() <= p0 or
               cell[layer–1][i+1] > 0 and random() <= p1 then
                cell[layer][i] = 1;             // fill the cell
                extrudedFilament = 0.0;         // clear filament
            end if
        end if
        drawNextArc(cell[layer][i], z, i);
        extrusion = extrusion + e1; // extrude
    end loop
    cell[layer][steps] = cell[layer][0];
end loop
```

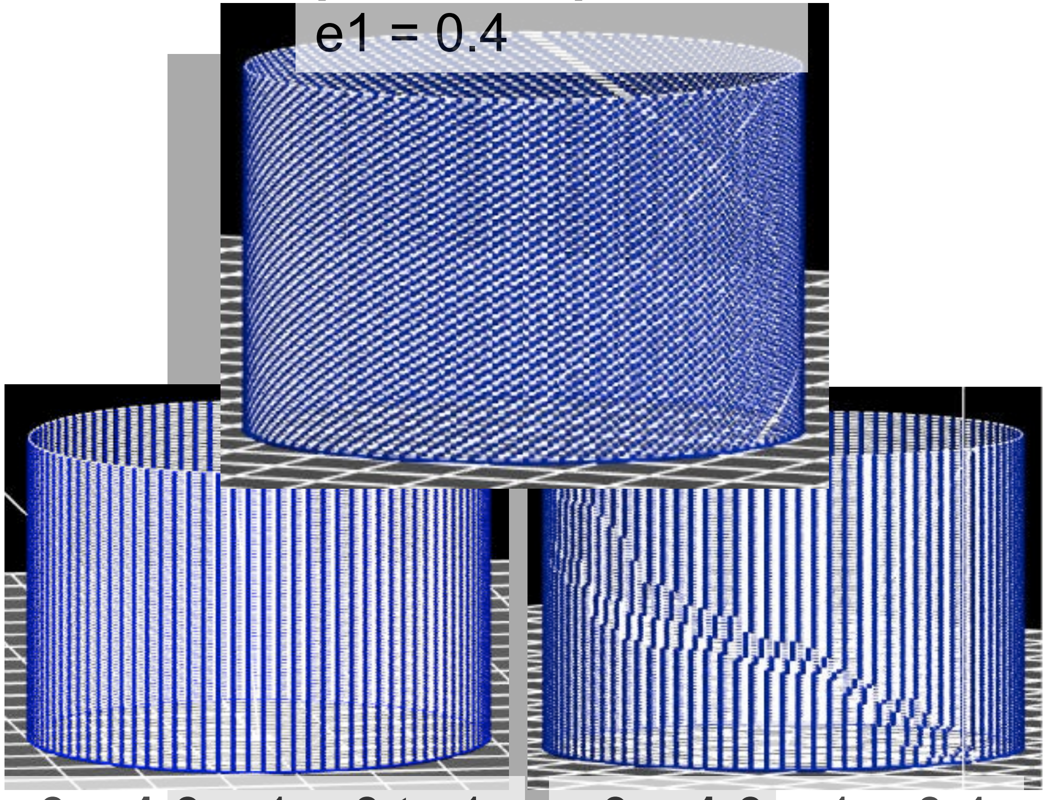
Simulation of typical patterns

- ► **A program written in Python generates G-code (CAM program) is used.**
  - ■ G-code execution results are visualized by Repetier-Host (a CAM tool for 3D printers).

p0 = 0.99, p1 = 0.9, e1 = 0.4



p0 = 0.99, **p1 = 1.0**, e1 = 0.4



p0 = 1.0, p1 = 0 to 1, e1 = 0.4

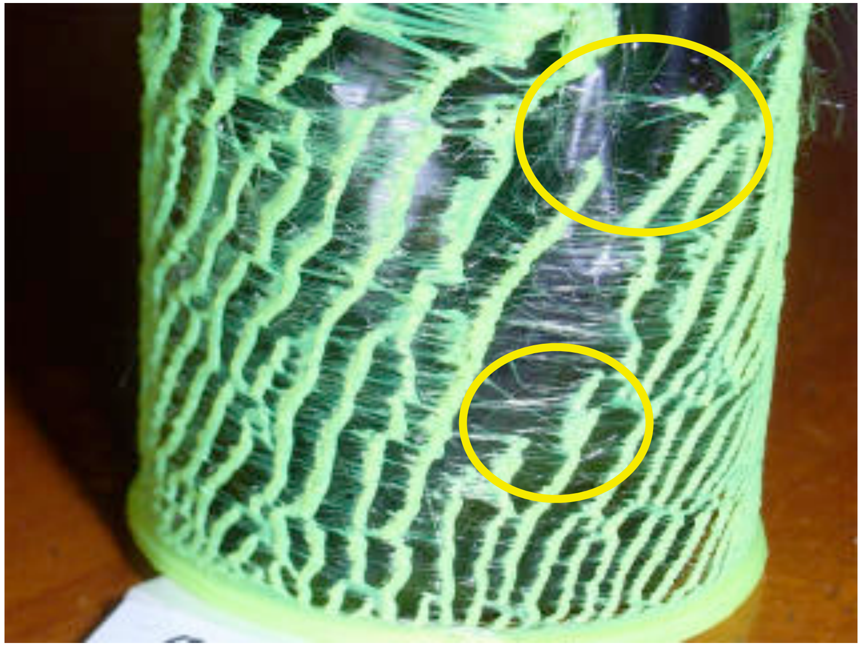
p0 = 1.0, p1 = 0.4, e1 = 0.4

Various printed patterns and Simulation results

- ► Extinction of stripes
- ► Splitting and merging of stripes
- ► Waves and meshes

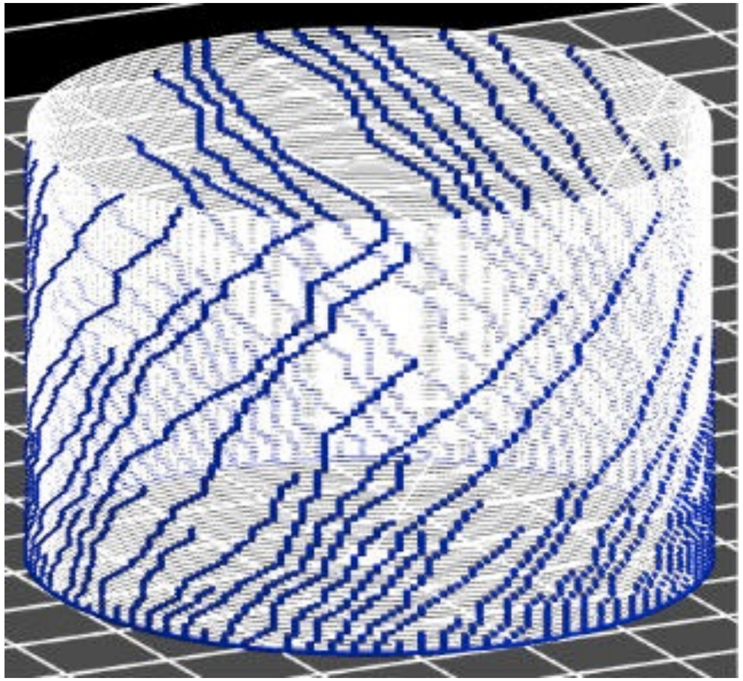
Extinction of stripes

A printed result



The layer height is 0.2 mm. PLA.

A simulation result

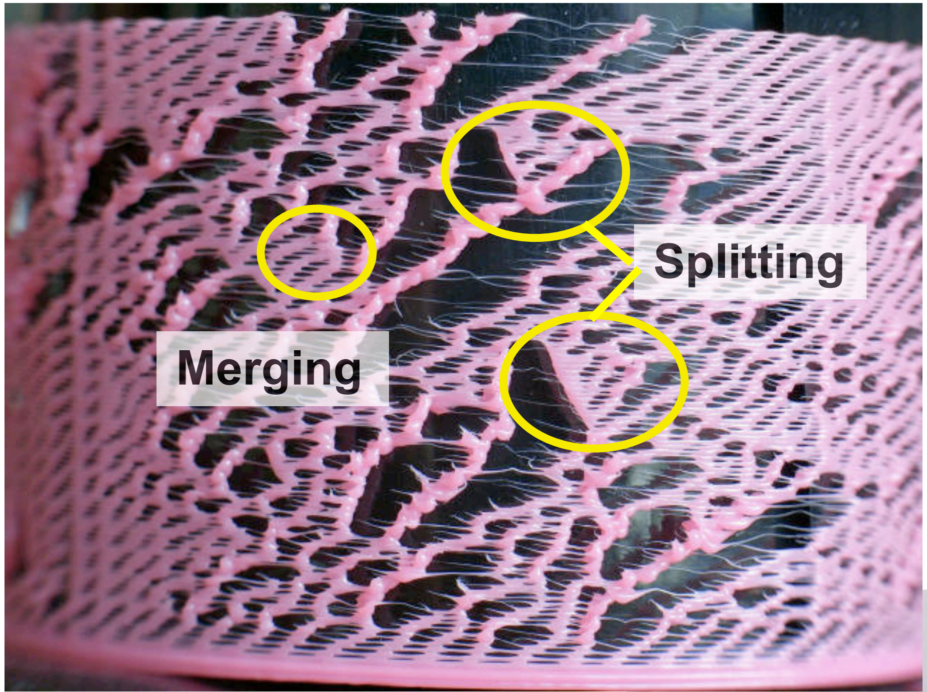


p1 = 0.9, **p0 = 0.97**, e1 = 0.6

Extinction of stripes (cont'd)

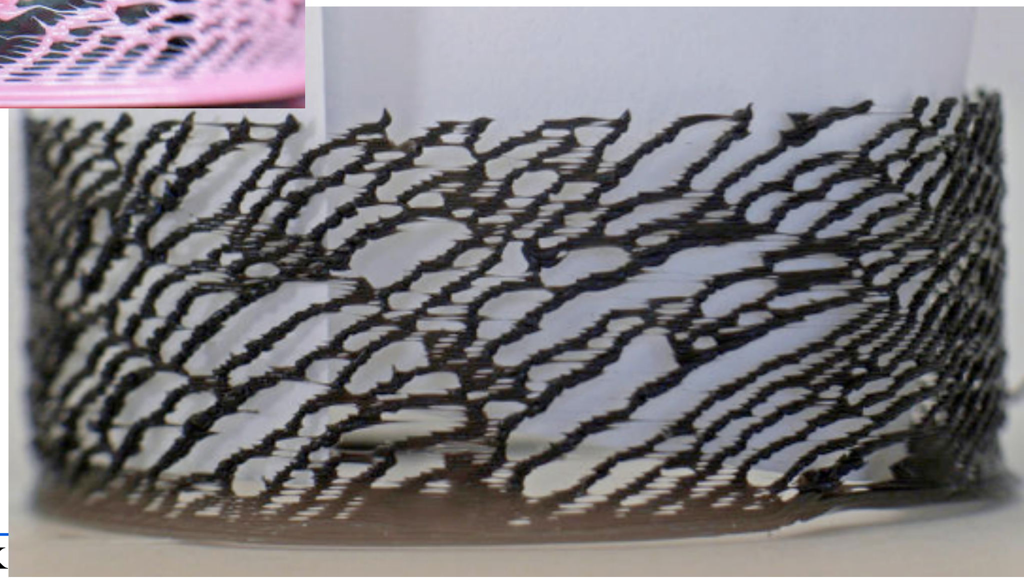
Splitting and merging stripes

► Printed results

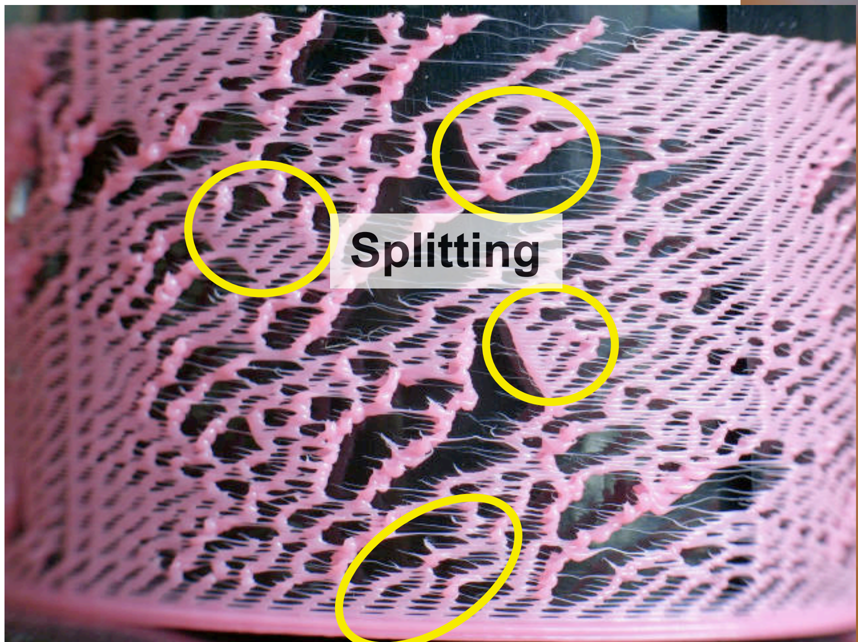


ABS

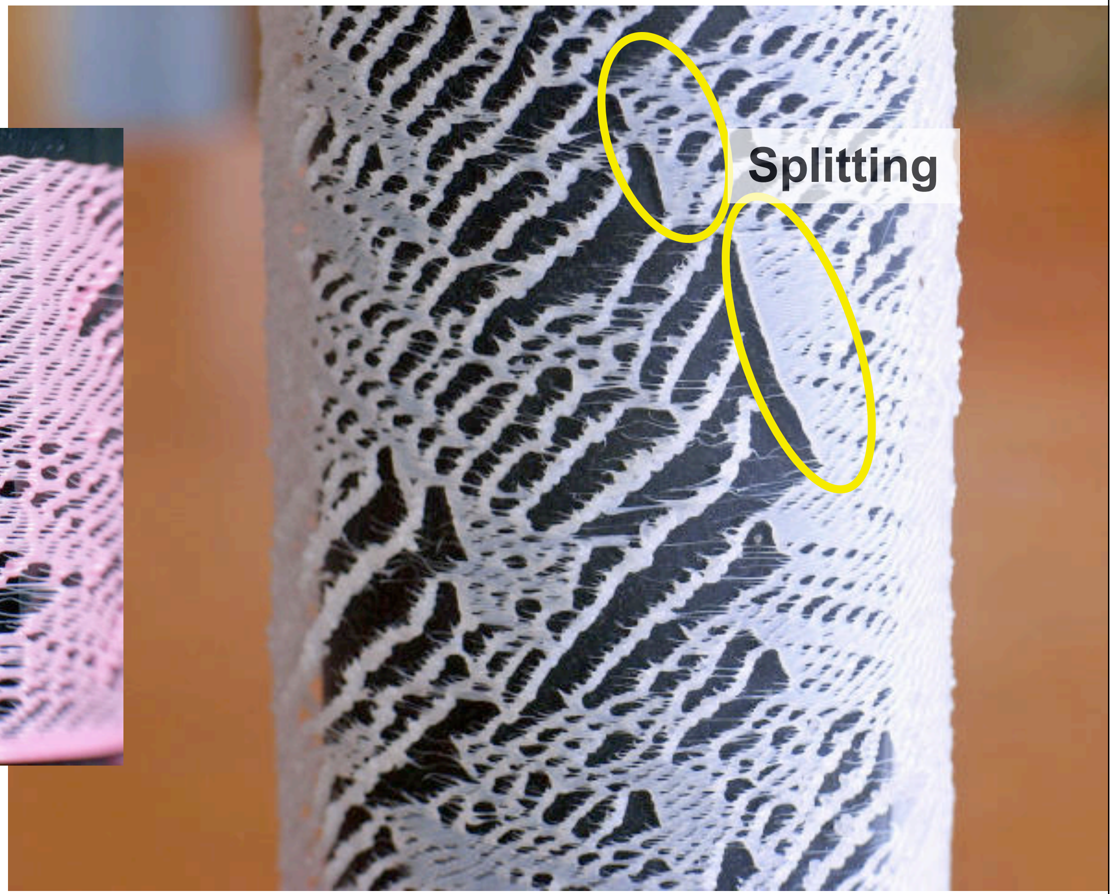
PLA



Splitting and merging stripes (cont'd)



Merging

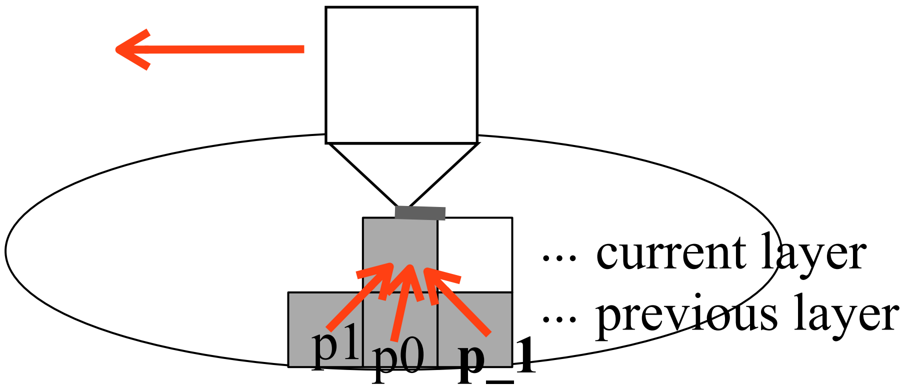


Computational model for splitting/merging stripes

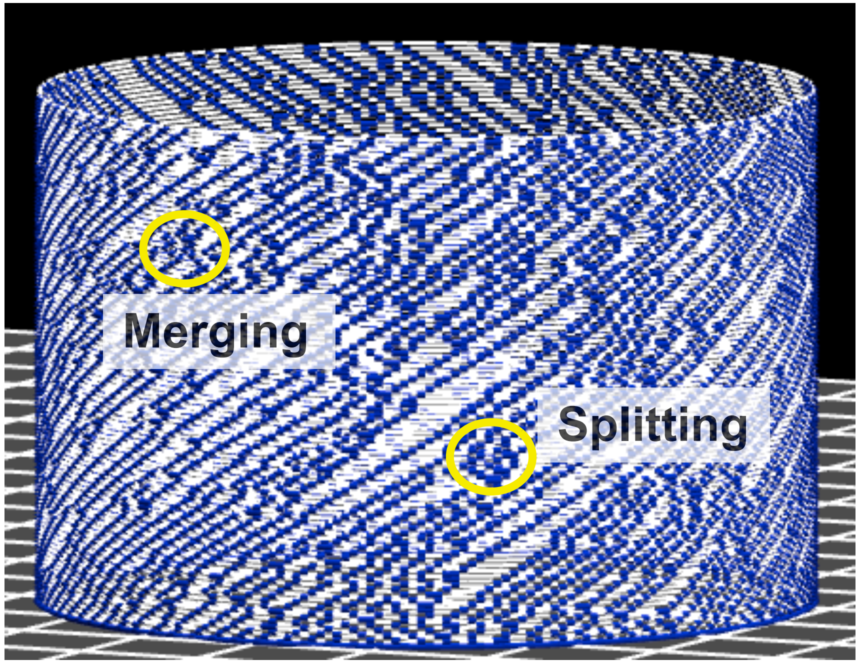
- **The computational rule must be extended to simulate splitting and merging.**
  - if extruded filament >= 1 then
    if cell[*l*–1][*i*–1] > 0 then cell[*l*][*i*] = 1 & filament cleared at probability p_1
    else if cell[*l*–1][*i*+1] > 0 then cell[*l*][*i*] = 1 & filament cleared at probability p1
    else if cell[*l*–1][*i*] = 1 then cell[*l*][*i*] = 1 & filament **reduced by C** /* C < 1 */
    at probability p0

    else cell[*l*][*i*] = 0

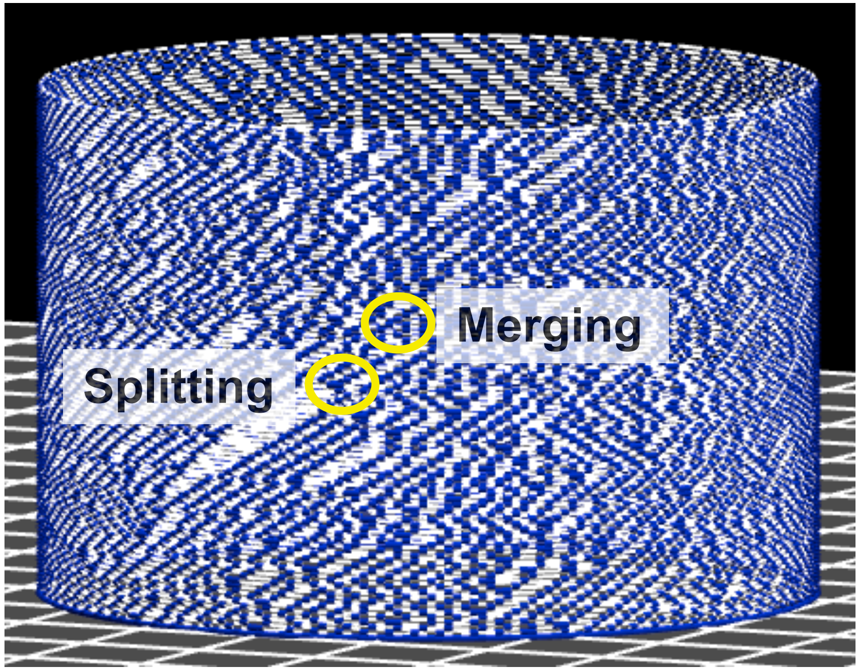
    else cell[*l*][*i*] = 0



Simulation results of splitting/merging stripes



p_1 = 0.7, C = 0.8,
p1 = 0.9, p0 = 0.995, e1 = 0.4

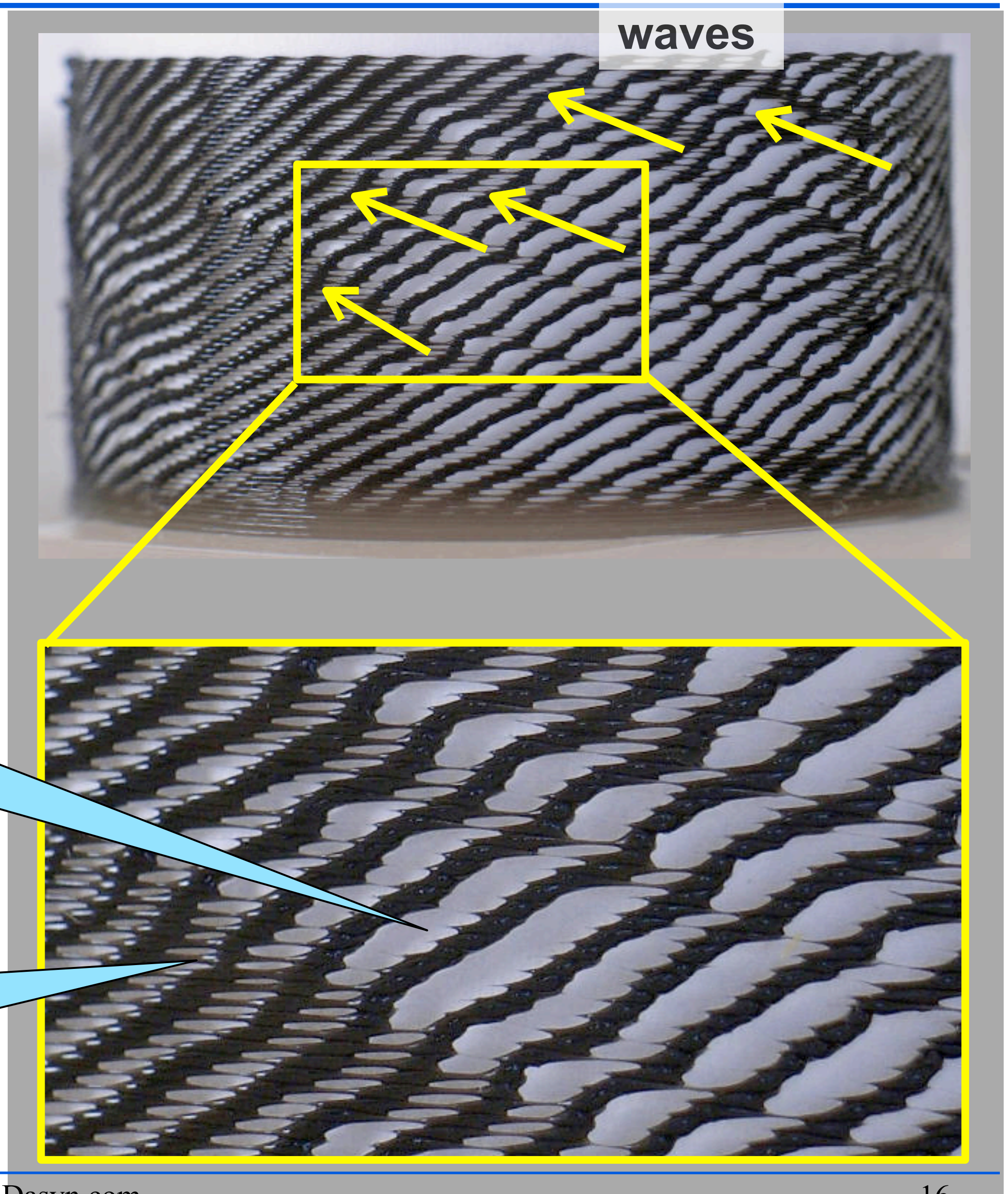


p_1 = 0.99, C = 0.9,
p1 = 0.9, p0 = 0.995, e1 = 0.4

Waves

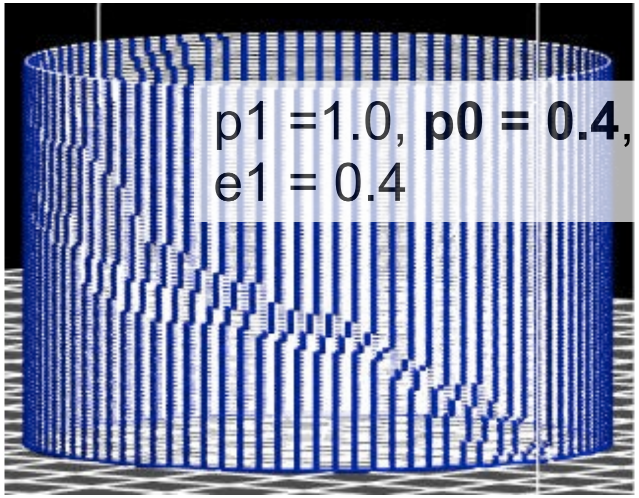
- ► **Stripes are weakly connected by thick strings.**
- ► **"Waves" moves across stripes.**

The layer height is 0.25 mm. PLA

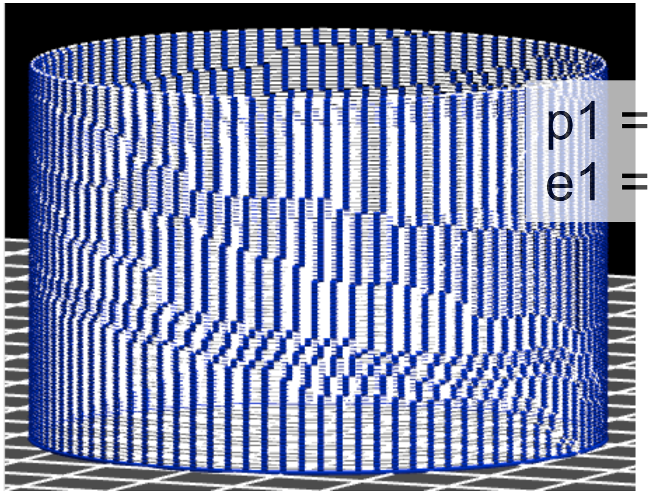


Simulation of waves

► **It is easy to see noises propagated across stripes.**



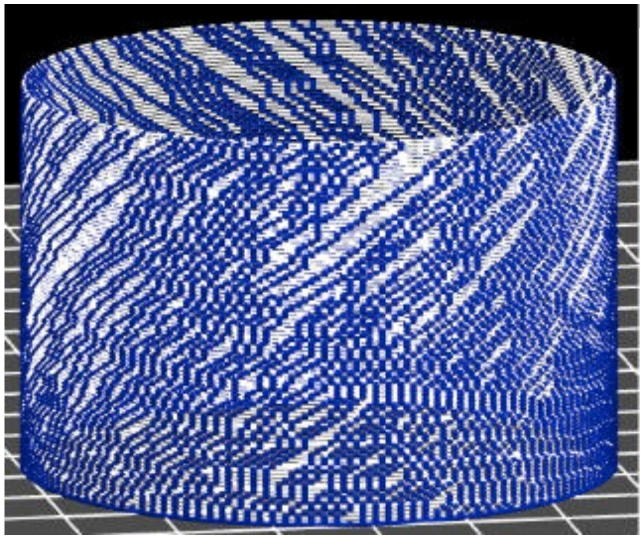
p1 =1.0, **p0 = 0.4**,
e1 = 0.4



p1 =1.0, **p0 = 0.6**,
e1 = 0.4

Noises are added to the initial condition.

► **It is less easy to see wave propagation here.**



p1 = 0.99, **p0 = 0.8**,
e1 = 0.5

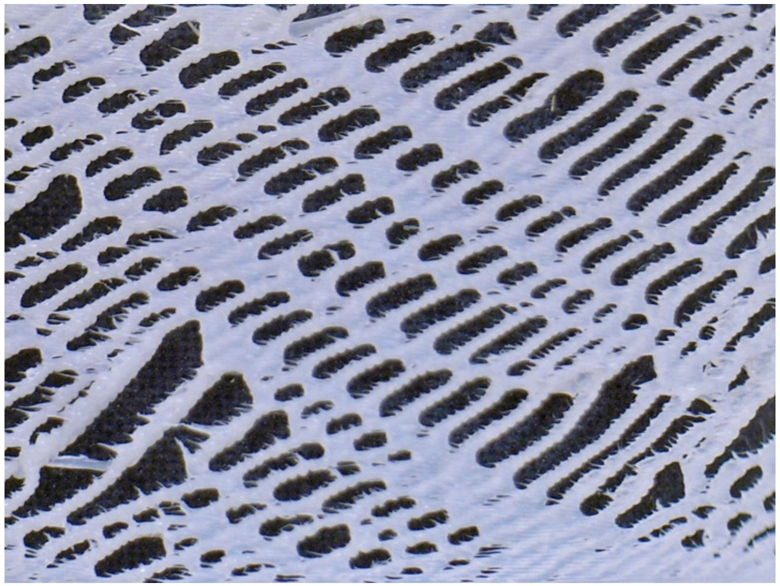
The observed waves and the simulated waves may be different because strings are not simulated.

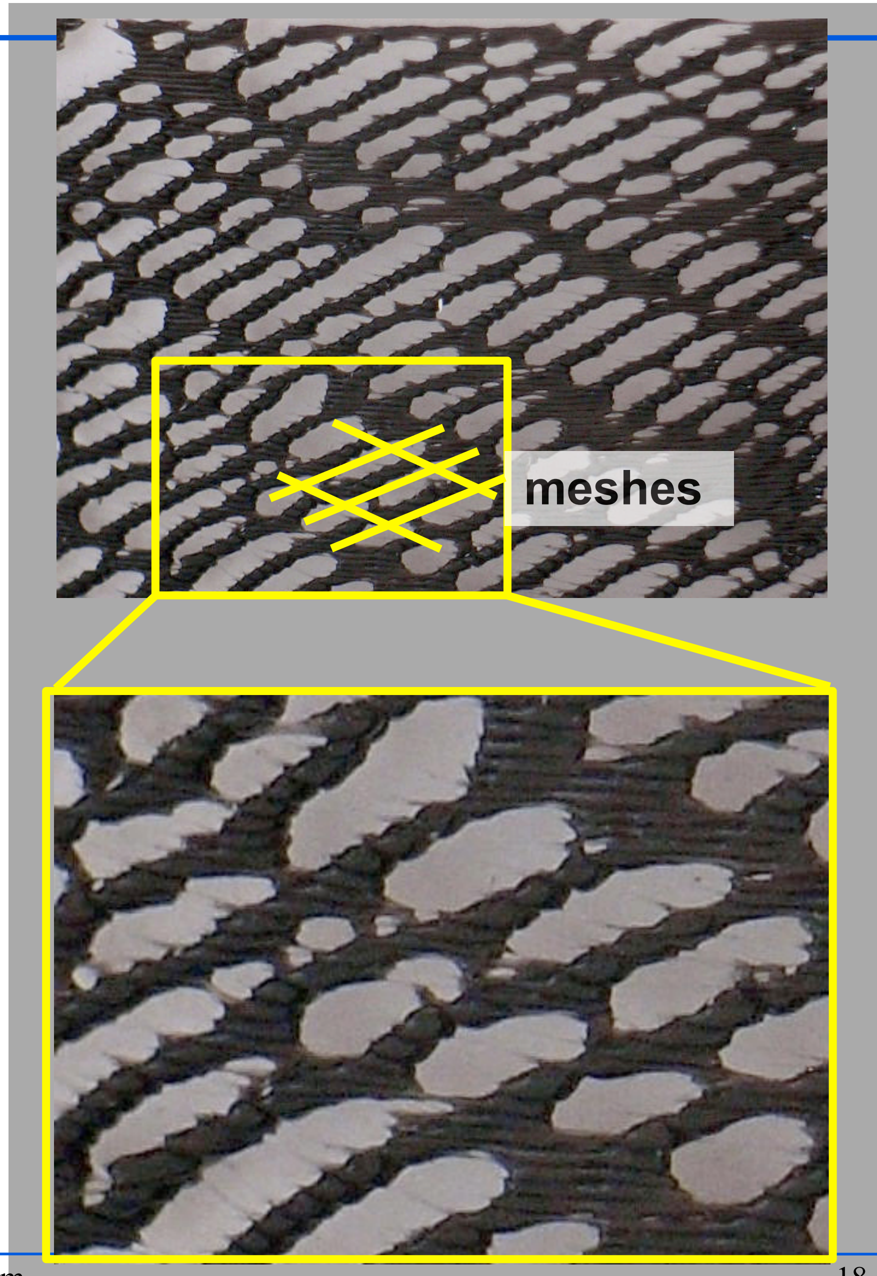
Meshes

- ► **Stripes are strongly connected.**
- ► **Meshes have not yet been successfully simulated.**

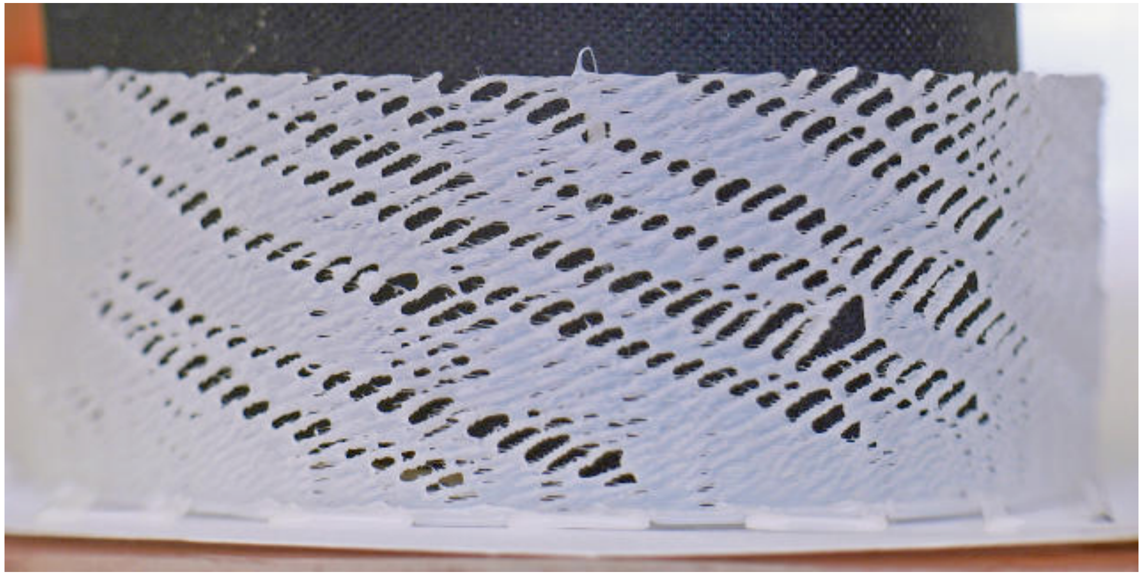
The layer height is 0.15 mm. PLA

ABS





Meshes (cont'd)



Summary

- **FDM 3D printers can generate self-organized patterns.**
  - "Pure" self-organized patterns can be generated by the 1D-CA-like printing method.
  - Fluctuating stripes, splitting and merging stripes, waves, and meshes can be generated.
- **The printed patterns can be simulated by 1D CA.**
  - Stripes can be simulated.
  - Splitting and merging stripes, waves, and meshes can be "simulated", but it is not yet certain that it is the right way.
- **Web site for this presentation**
  - http://bit.ly/1n7PJQr