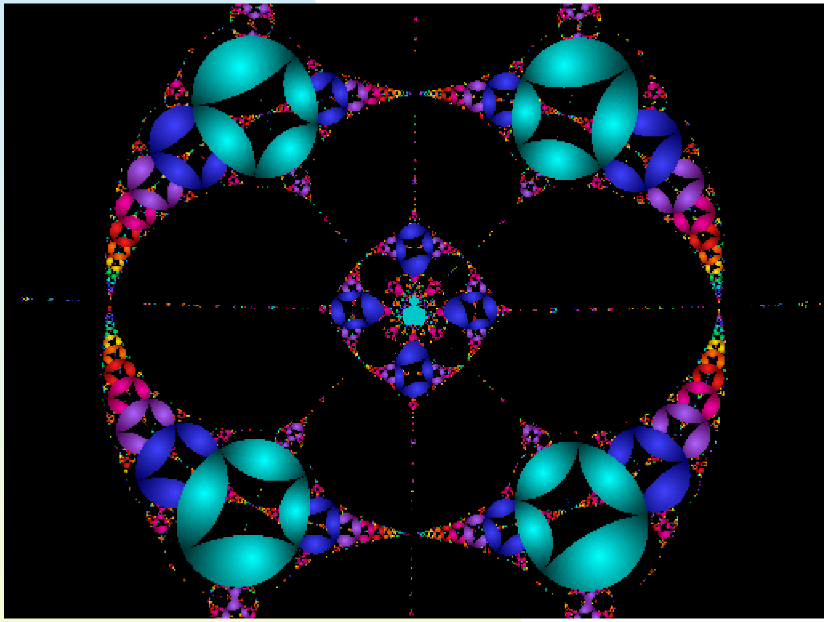
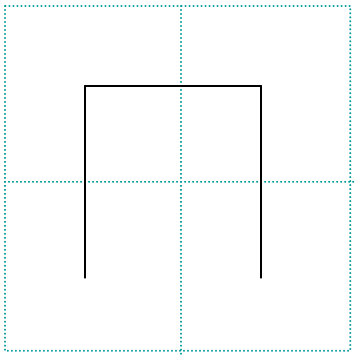
Session overview



- More on L-systems:
  Hilbert curve
  Branching and trees

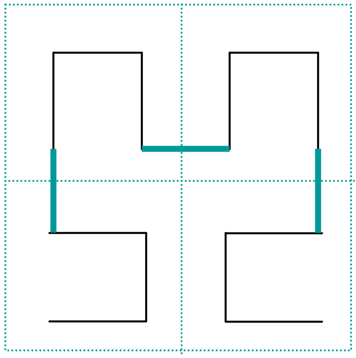
The Hilbert curve

- Start with a unit square region
- Divide the region into four squares, each one-half the length and width of the original square
- Connect the midpoints of the smaller squares



The Hilbert curve (cont.)

- Now repeat the pattern, scaled by a factor of ½, in each of the four smaller squares, with the bottom two patterns rotated appropriately
- Connect the patterns



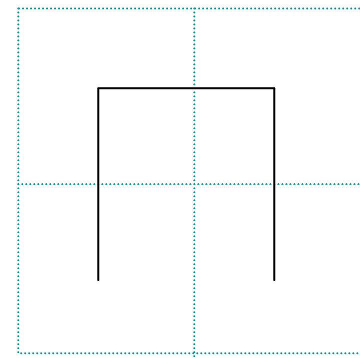
- Note that if we start tracing the pattern at the bottom left, the two bottom shapes would be traced in a counter-clockwise direction and the top two would be traced in a clockwise direction

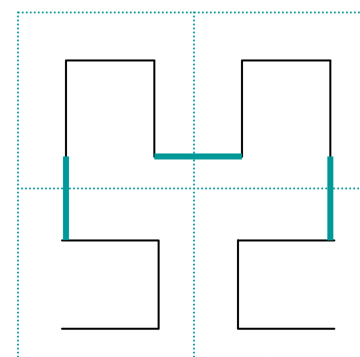
Nodes

- For the Hilbert curve, we need two nodes to represent the two different orientations of the tracing
- Let L represent the clockwise orientation (interpreted as +F-F-F+)
- Let R represent the counter-clockwise orientation (interpreted as -F+F+F-)
- Note that here we have let + indicate left turns and - right turns

L-system for Hilbert curve

- Axiom: L
- Production rules:
  - L → +RF-LFL-FR+
  - R → -LF+RFR+FL-
  - F → F
  - + → +
  - - → -
- Parameter: $\delta = 90°$
- Code is part of `Lsystems.cpp`





Modeling plants

- In plants, we have a stem with branches at various points along the way
- When we draw such a structure, we will come to a point where we need to go in two directions
- Thus, we need to remember where we are and in what direction we are going when we take a branch
- Use a stack, where each element on the stack contains the current point and current angle

Stack symbols for L-systems

- [indicates to push the current point and angle on the stack
-] indicates to pop the current point and angle from the stack

- Do quiz question

Examples

- In program `Lsystems.cpp` the following fractals demonstrate this approach:
  - Weedlike plant I
  - Weedlike plant II
  - Bush

3-D L-systems

- The current orientation is represented by a set of three orthonormal vectors **H**, **L**, and **U**
- These represent the heading, left direction, and up direction
- New symbols are needed:
  - \+ - : turn left/right around **U** axis
  - & ^ : pitch down/up around **L** axis
  - \ / : roll left/right around **H** axis
  - | : turn around (rotate 180º around **U**)

Transformation matrices

$$\mathbf{R_U}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R_L}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R_H}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Parametric L-systems

- In a parametric L-system, each of the symbols takes a parameter:
  - F(*a*): draw a line of length *a*
  - +(*a*): rotate around **U** by an angle *a*
  - &(*a*): rotate around **L** by an angle *a*
  - /(*a*): rotate around **H** by an angle *a*

Branching pattern L-system

- Axiom: A(1)
- Production: A(s) $\rightarrow$ F(s)[+A(s/1.456)][-A(s/1.456)]
- Parameter: δ = 85º
- Code for this system is also in `Lsystems.cpp`

Modeling trees

- One of the first L-system models of trees was proposed by Honda in 1971
- It made the following assumptions:
  - Tree segments are straight and their girth is not considered
  - A mother segment produces two daughter segments through one branching process
  - The lengths of the two daughter segments are shortened by constant ratios, r_1 and r_2, with respect to the mother segment

Modeling trees (cont.)

- Assumptions (cont.)
  - The mother segment and its two daughter segments are contained in the same branch plane
  - The daughter segments form constant branching angles, a_1 and a_2, with respect to the mother branch
  - The branch plane is fixed with respect to the direction of gravity so as to be closest to a horizontal plane
    - an exception is made for branches attached to the main trunk
    - in this case, a constant divergence angle α between consecutively issued lateral segments is maintained

An L-system for a tree

- Axiom: A(1, 10)
- Productions:
  - $A(l, w) \rightarrow\ !(w)F(l)[\&(a_1)B(l*r_2, w*w_r)]/(d)A(l*r_1, w*w_r)$
  - $B(l, w) \rightarrow\ !(w)F(l)[-(a_2)\$C(l*r_2, w*w_r)]C(l*r_1, w*w_r)$
  - $C(l, w) \rightarrow\ !(w)F(l)[+(a_2)\$B(l*r_2, w*w_r)]B(l*r_1, w*w_r)$
- New symbols:
  - !(*w*) sets the line width to *w*
  - $ rolls the turtle around its own axis so that vector **L**, pointing to the left of the turtle, is brought to a horizontal position
  - $ is accomplished by computing $\mathbf{L} = \mathbf{V} \times \mathbf{H}$ and $\mathbf{U} = \mathbf{H} \times \mathbf{L}$, where **V** is a vector opposite to gravity

An L-system for a tree (cont.)

- Typical constants used:
  - r_1 = 0.9 = contraction ratio for trunk
  - r_2 = 0.6 = contraction ration for branches
  - a_1 = 45 = branching angle from trunk
  - a_2 = 45 = branching angle for lateral axes
  - d = 137.5 = divergence angle
  - w_r = 0.707 = width decrease rate
- Code for generating this fractal is in `Lsystems.cpp`

Go the other way

- In the previous L-system, we started with a wide branch at a certain length and decreased the length and width of the branches in subsequent levels
- It is also possible to go the other way - to start each level at a fixed width and length and widen and elongate the previous level branches

A second tree L-system

- Axiom: !(1)F(200)/(45)A
- Productions:
  - A $\rightarrow$!(v_r)F(50)[&(a)F(50)A]/(d_1)[&(a)F(50)A]/(d_2)[&(a)F(50)A]
  - F(l) $\rightarrow$ F(l*l_r)
  - !(w) $\rightarrow$!(w*v_r)
- Typical constants used for this system:
  - d_1 = 94.74 = divergence angle 1
  - d_2 = 132.63 = divergence angle 2
  - a = 18.95 = branching angle
  - l_r = 1.109 = elongation rate
  - v_r = 1.732 = width increase rate

Tropism

- In nature, many tree branches bend, whether it be in response to gravity, sunlight, or some other stimulus
- Such an effect is called ***tropism***
- It can be implemented in an L-system by adjusting the orientation of the drawing after drawing each segment

Tropism (cont.)

- The orientation adjustment, α, is calculated by the formula $\alpha = e\mathbf{H}\times\mathbf{T}$
- *e* is a parameter capturing axis susceptibility to bending (0.22 in the example code)
- **T** is the tropism vector, indicating the direction of the effect ((0, -1, 0) in the example code)
- **H** is then replaced by $\mathbf{H} + \alpha$
- Tree II in `Lsystems.cpp` implements the second tree L-system with tropism