Simulating Trees with Fractals and L-Systems

Background - Fractals

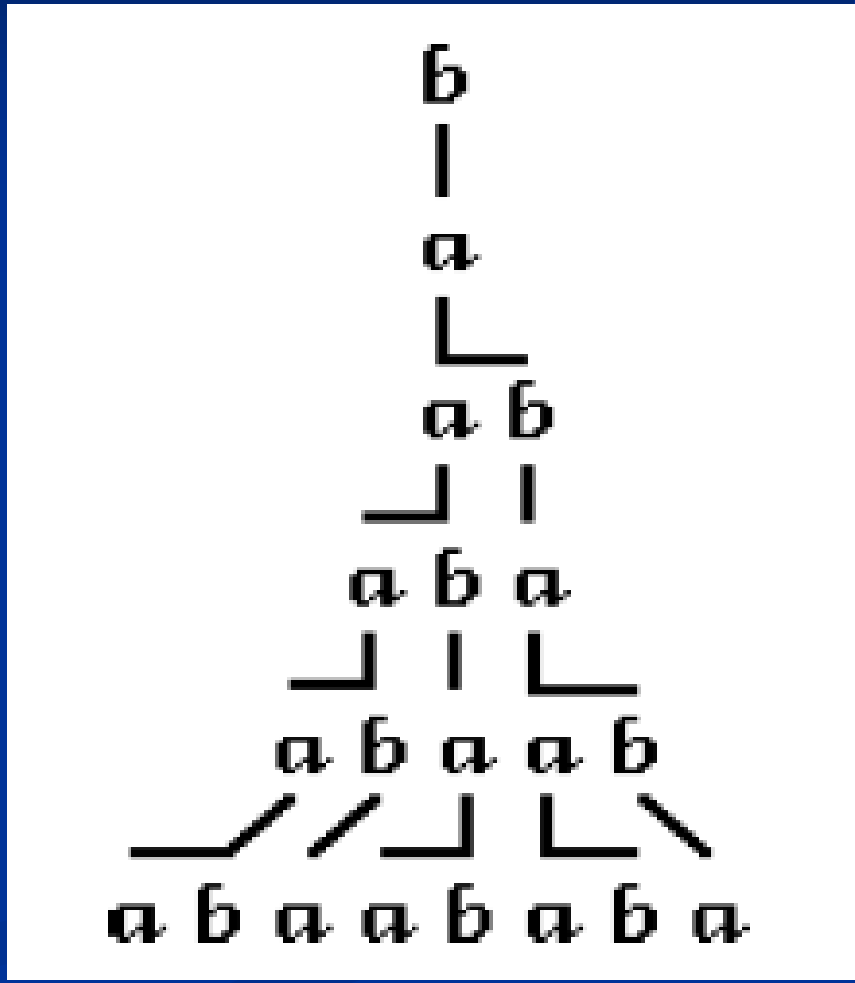
- Fractals are recursive, self-similar structures
  - Infinitely detailed – zooming in reveals more detail
  - Similar, though not necessarily identical, at any level of magnification
  - Generated using a variety of methods, such as L-Systems
- Many natural forms display fractal geometry (like trees!)

Background – L-Systems

- Formal grammar developed by Aristid Lindenmayer as a theoretical framework for studying development of simple multicellular organisms
- Subsequently applied to investigate higher plants
- Uses rewriting rules (productions) to grow a string or system
- Productions are applied in parallel (unlike Chomsky grammars)
  - This is motivated by biological considerations – this is how living organisms grow

Simple L-System

- Strings built of letters ***a*** and ***b***
- Each letter is associated with a rewriting/production rule:
  - ***a*** → ***ab***
  - ***b*** → ***a***
- Rewriting starts from an *axiom*, a starting string (b in this case)
- Each production is applied simultaneously in each step



L-Systems – Properties

- Can be context-free or context-sensitive
  - Different production rules for the same symbol based upon neighboring symbols
- If run repeatedly and interpreted as images, can produce fractal geometry
  - Can describe many "traditional" fractal patterns, such as Koch curves and constructions
  - Can describe & produce very complex fractal patterns

Describing Trees

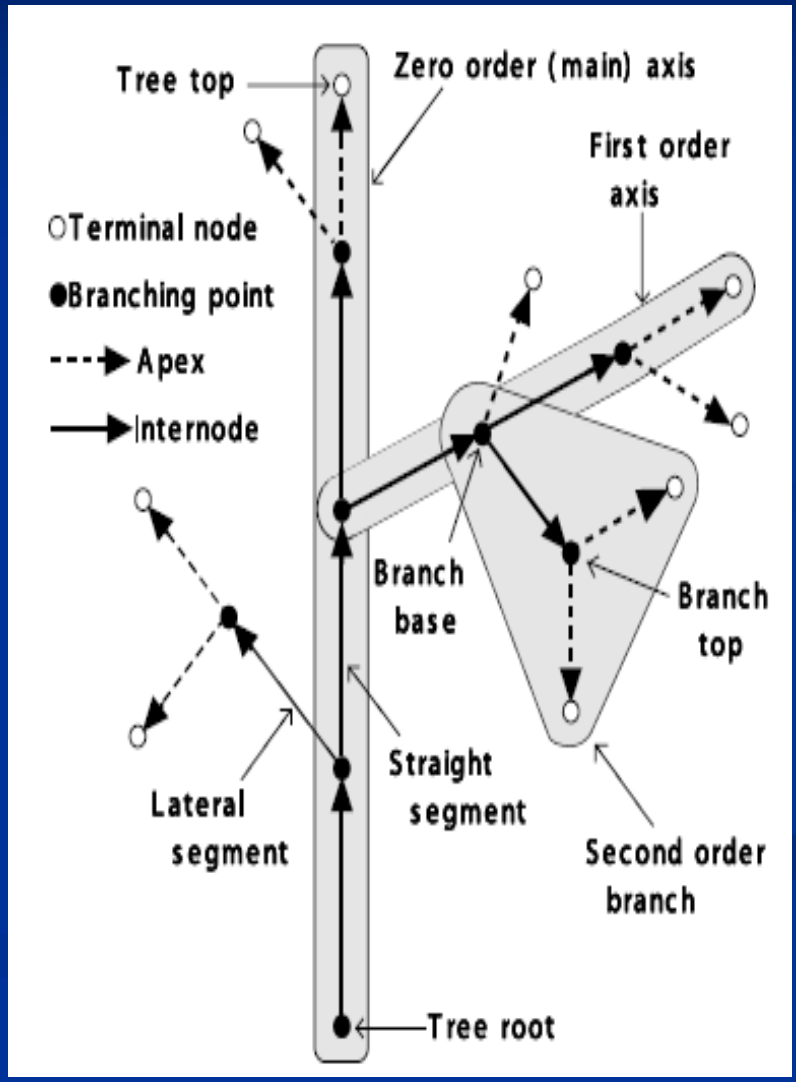
- A *rooted tree* has edges that are directed and labeled
- Edge sequences form paths from a distinguished node, called the *root* or *base*, to *terminal nodes*
  - In biological sense, these edges are *branch segments*
  - A segment followed by at least one more segment in some path is called an *internode*
  - A terminal segment (with no succeeding edges) is called an *apex*

Axial Trees

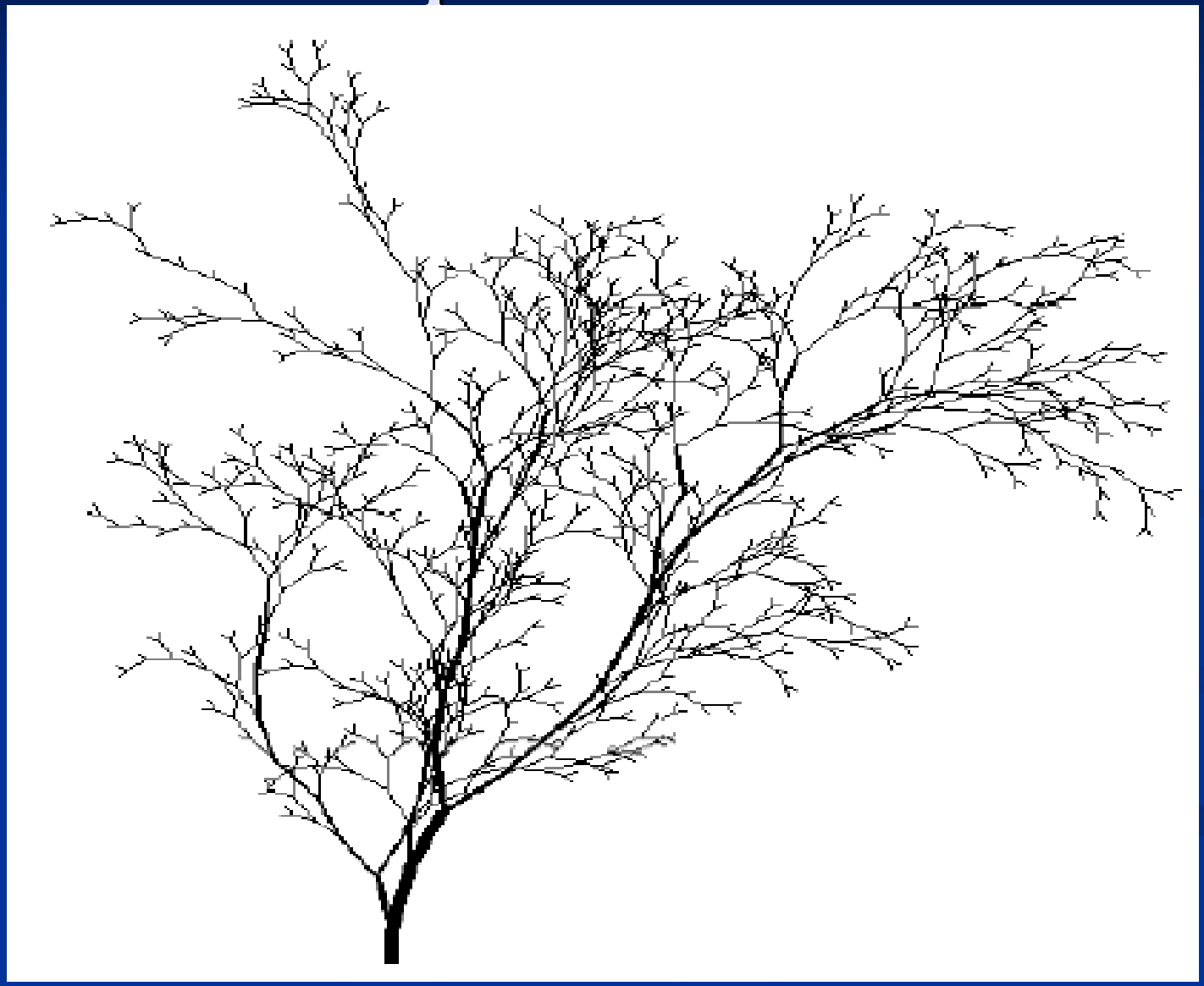
- Special type of *rooted tree*
- At each node, at most one outgoing *straight* segment is distinguished
- All remaining edges are called *lateral* or *side* segments.
- A sequence of segments is called an *axis* if:
  - the first segment in the sequence originates at the root of the tree or as a lateral segment at some node
  - each subsequent segment is a straight segment
  - the last segment is not followed by any straight segment in the tree.
- Together with all its descendants, an axis constitutes a *branch*. A branch is itself an axial (sub)tree.

Axial Trees

- Axes and branches are ordered
  - The root axis (trunk) has order zero.
  - Axis originating as a lateral segment of an *n*-order parent axis has order *n*+1.
  - The order of a branch is equal to the order of its lowest-order or *main* axis



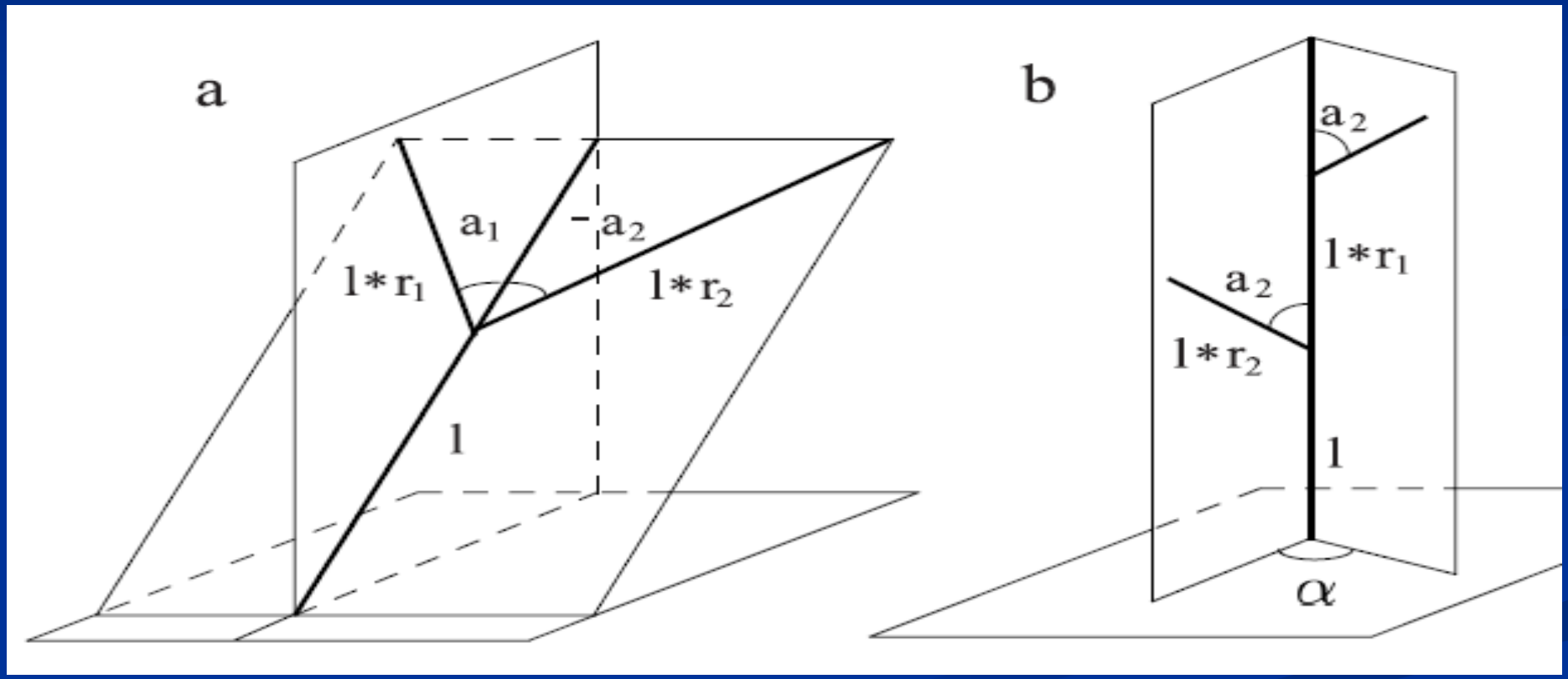
Example Axial Tree



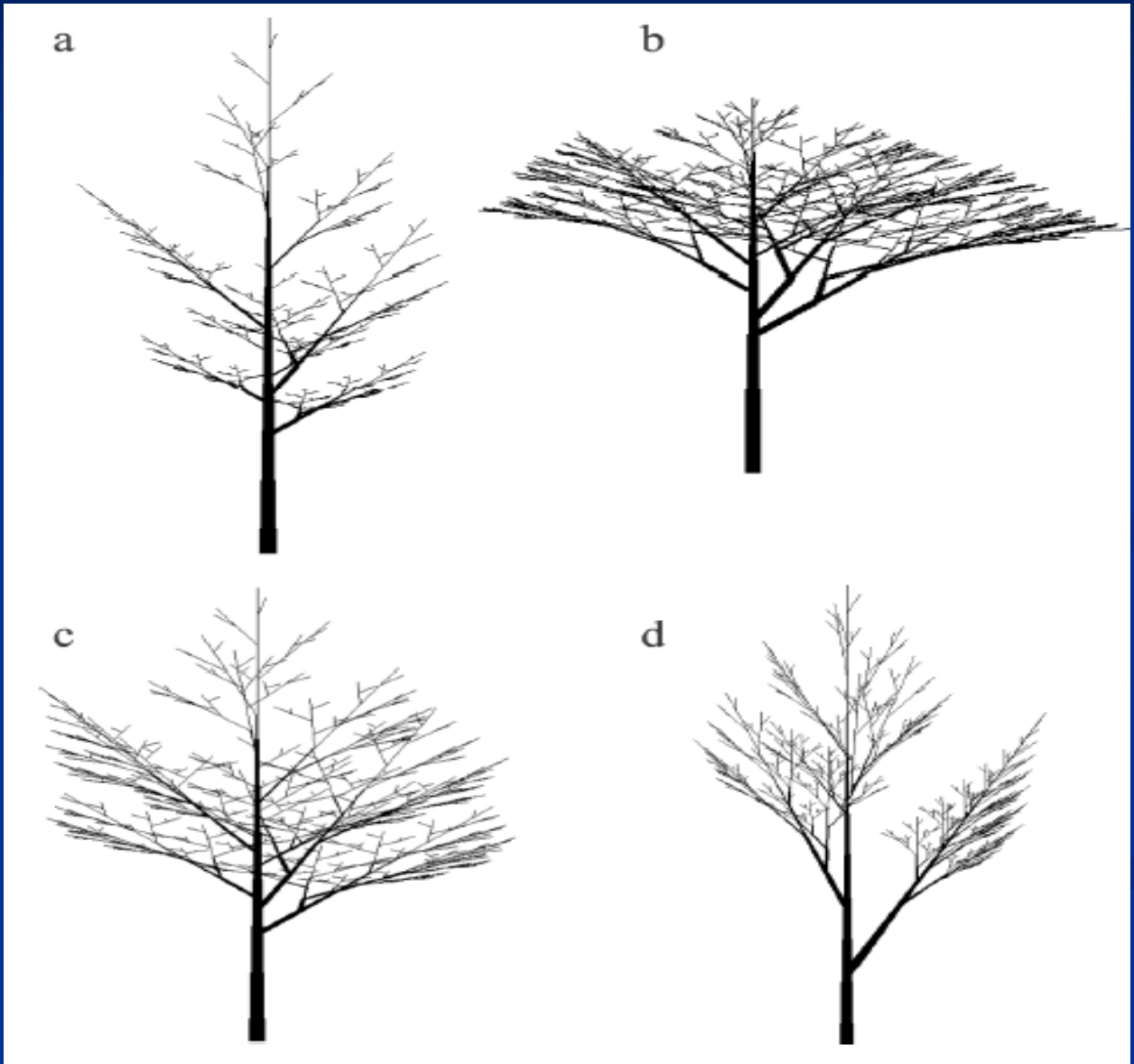
Honda's Model

- Limited model with following assumptions:
  - Tree segments are straight and their girth is not considered
  - A mother segment produces two daughter segments through one branching process
  - Lengths of the two daughter segments are shortened by constant ratios, r_1 and r_2, with respect to the mother segment
  - Mother and 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
  - 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, where a constant *divergence angle a* between consecutively issued lateral segments is maintained

Honda's Geometry



Honda's Trees



Example of L system

- **variables** : X F
- **constants** : + − [ ]
- **start** : X
- **rules** : (X → F−[[X]+X]+F[+FX]−X), (F → FF)
- **angle** : 25°
- Here, F means "draw forward", − means "turn left 25°", and + means "turn right 25°". X does not correspond to any drawing action and is used to control the evolution of the curve. The square bracket "[" corresponds to saving the current values for position and angle, which are restored when the corresponding "]" is executed.