Alife – Lecture 4
Models of Growth and Development

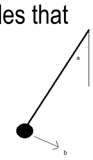
But first -- something on Dynamical Systems language

A Dynamical System is any System you wish to define with a specified list of a finite number of variables **PLUS** a set of laws specifying how each variable changes over time, depending on values of the other variables.

Acknowledgment: some later L-system slides borrowed from Gabriela Ochoa: http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html

An example of a Dynamical System

EG a pendulum swinging across a wall has 2 variables that will specify its STATE at time t, namely
a = current angle of string with vertical, and
b = current angular velocity, speed of swing

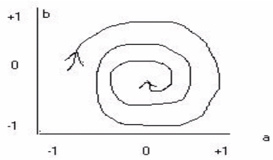


Some dynamics textbooks will give you formulae, using values for gravity and amount of friction, such that

rate of change of a = da/dt = f1(a,b)
rate of change of b = db/dt = f2(a,b)
..don't worry here exactly what formulae f1 and f2 are.

Dynamical Systems

With this simple DS with only 2 variables, you can plot how they change over time on a graph.



This graph shows the **STATE SPACE** (or phase space) of the DS -- any particular state of the pendulum corresponds to a particular point on this graph.

The particular line shown with arrows is a **TRAJECTORY** through state space, determined by the laws of (in this case) gravity and friction.

Attractors

With this pendulum, all trajectories, from any starting point in state space, will finish up with a=0 and b=0

I.e. with the pendulum straight down and stationary

That end-point of all trajectories in the state space of the pendulum DS is a **POINT ATTRACTOR**

You can have **repellors** as well -- eg the DS of a stick standing on its end (upside-down pendulum), and there may be more than one point attractor in a DS.

More Attractors

There are other kinds of attractors -- eg **cyclic attractor** when a trajectory winds up in a repetitive, never-ending cycle of behaviour.

And **strange attractors.**

Continuous or Discrete DSs

The pendulum is a 2-dimensional **continuous** dynamical system. You can have 100-dim or 1000-dim DSs. And they need not be continuous.

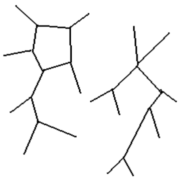
A Cellular Automaton is a **discrete dynamical system**.

If there are 100 cells, then it is a 100-dim DS with a deterministic trajectory (following all the update rules) -- strictly you need a 100-dim graph to picture it.

Basins of Attraction

A CA with a finite number of cells has 'only' a finite number of possible states-of-the-system (finite but usually **enormous**). Since rules are deterministic, there are one-way arrows giving trajectory from any one state to the next (for the next CA timestep).

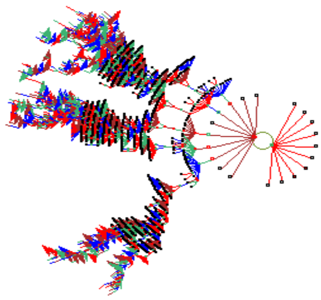
Hence with a CA all trajectories end in point attractors or limit cycles. All trajectories that wind up in the same attractor form a **BASIN of ATTRACTION.**



Garden of Eden States

Starting points of trajectories with no possible ancestors are **'Garden of Eden states'**.

cf. Andy Wuensche's work on basins of attraction



Is this relevant to Development ?

Someone like Waddington would say **yes**.
(cf. Waddington 'Towards a Theoretical Biology')

Why does an egg always end up as a hen (or cock)?
Despite different conditions, food, environment etc.

The developmental process is like a Dynamical System with an attractor (or 2 attractors, hen and cock), which are (nearly) always reached despite disturbances.

A bit like the pendulum, though **much** more complex!

Perturbations

Sometimes disturbances to developmental processes (natural or through nasty scientists) result in abnormal or misplaced structures (eg legs replace antennae in flies). These are not random structures, but usually 'sensible ones in the wrong place'.

Basins of attraction:- disturbances can possibly divert trajectories into a different basin.

No longer fully deterministic in the sense of a formal CA.

Morphogenesis

... the origin of shape or form, is another name for this kind of development. (Beware: 'development' means something else to eg psychologists)

How does **one** cell, at conception, split and double, time and time again, and differentiate into all the cells of a plant, animal or human ? All the cells of a body have the **same** DNA, but different genes are 'turned on' in different cells, so some are skin, some liver, some blood ... .

cf. Kauffman's NK Random Boolean Networks.

'Recipe' or Description ?

General assumption: the DNA does not specify 'as some kind of description' the final form of the body.

More like 'a **recipe'** for baking a cake.

Note: some people, such as Kauffman and Brian Goodwin ("How the Leopard changed its spots" 1994) emphasise 'generic constraints' on possible forms as at-least-as-important as the DNA. Others differ.

Morphogenesis in Alife

Biological morphogenesis has been the great black hole in biological theory -- though some see promising signs of real progress in the last decade or so.

A typical Alife approach is to look at possible, very general, ways to generate complex forms from relatively simple rules -- often very abstract.

See many references in Proceedings of Alife conferences

French Flag problem

Eg. Lewis Wolpert's 'French Flag problem'
Consider a 'worm' which is a linear string of cells which can be Red White or Blue (French Flag)

RRRRRRRRRRWWWWWWWWWBBBBBBBBBBB

If you cut this worm in half, the cells change state (change colour) in the appropriate way

RRRRRWWWWBBBB RRRRRWWWWBBBBBB

Exercise

Exercise: what is the simplest set of rules in each cell (treated as cells within a CA, inputs from left and right) that can produce this effect?

Do you need something like a 'chemical gradient' from left to right?

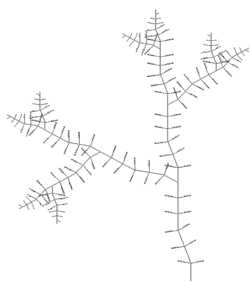
Lindenmayer L-Systems

Aristid Lindenmayer
Herman & Rosenberg , 'Developmental Systems and Languages' 1975
Prusinkiewicz.

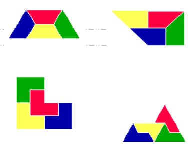
Simple **rewrite** rules to model development in plants, initially specifically branching structures.

L-Systems

- A model of morphogenesis, based on forn grammars (set of rules and symbols)
- Introduced in 1968 by the Swedish biologi Lindenmayer
- Originally designed as a formal descriptioı the development of simple multi-cellular organisms
- Later on, extended to describe higher plar and complex branching structures.

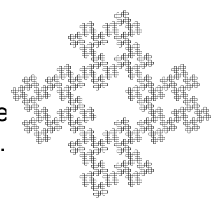


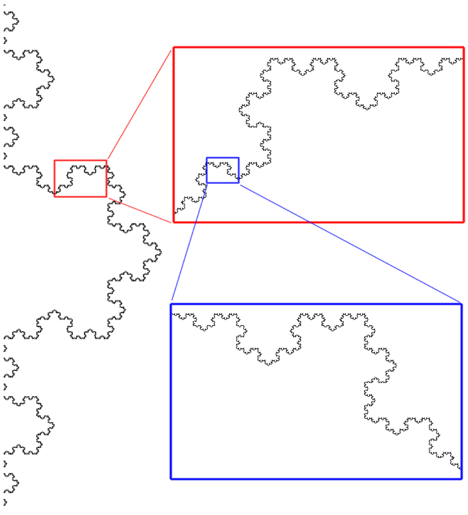
Self-Similarity



"When a piece of a shape is geometrically similar to the whole, both the shape and the cascade that generate it are called self-similar" (Mandelbrot, 1982)

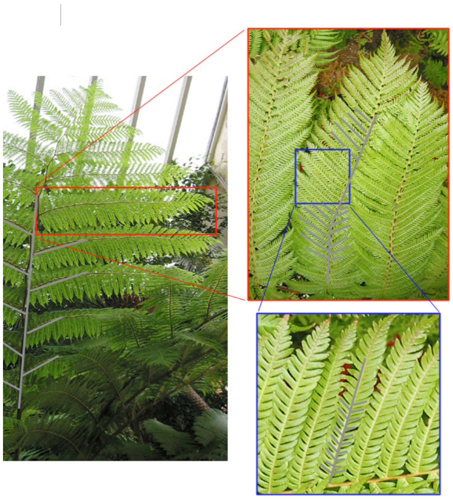
The recursive nature of the L-system rules leads to self-similarity and thereby fractal-like forms are easy to describe with an L-system.





Self-Similarity in Fractals

- Exact
- Example Koch snowflake curve
- Starts with a single line segment
- On each iteration replace each segment by 
- As one successively zooms in the resulting shape is exactly the same

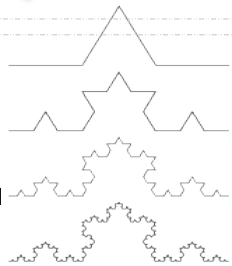


Self-similarity in Nature

- Approximate
- Only occurs over a few discrete scales (3 in this Fern)
- Self-similarity in plants is a result of developmental processes, since in their growth process some structures repeat regularly. (Mandelbrot, 1982)

Rewriting

- Define complex objects by successively replacing parts of a simple object using a set of rewriting rules or productions.
- Example: Graphical object defined in terms of rewriting rules - Snowflake curve
- Construction: recursively replacing open polygons



First four orders of the Koch Curve

Rewrite rules

Example Rewrite Rules:

A -> CB use all rules in parallel
B -> A
C -> DA
D -> C

Example Seed: A

A -> CB -> DAA -> CCBCB -> ...

L-Systems and Plant Growth

Need some convention for branching Left or Right
( ) for a left branch
[ ] for a right branch

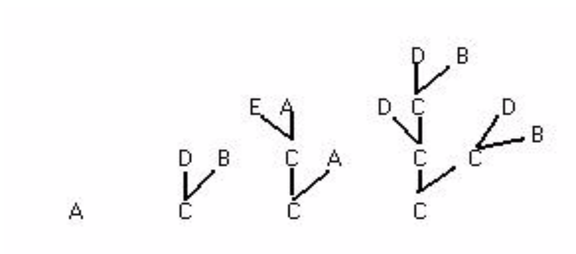
Example Rule Set:

A -> C [B] D
B -> A
C -> C
D -> C (E) A
E -> D

From a seed A, this grows:

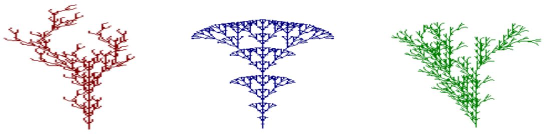
A -> C[B]D -> C[A]C(E)A -> C[C[C]D]C(D)C[B]D

Looking like ...



Modelling plants with L-systems

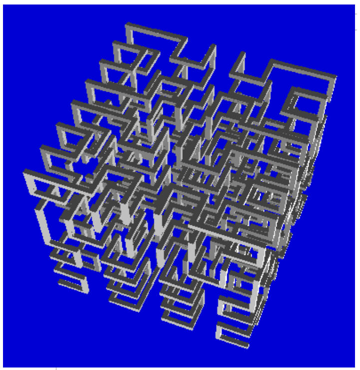
See Gabriela Ochoa's useful website (ex-COGS) mirrored at
http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html

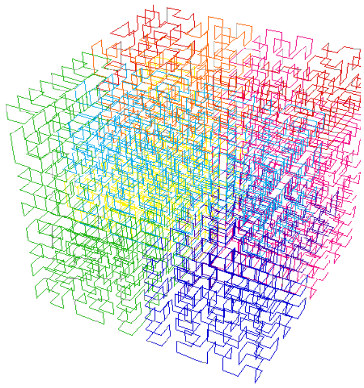


When to branch, what angle, when to produce a leaf or a flower.

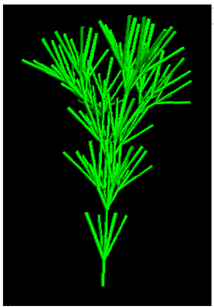
Use in computer animation techniques, Hollywood. Computer Art -- Karl Sims, William Latham

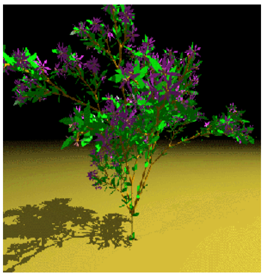
3D L-Systems





3D Bracketed L-Systems





Developmental rules for Neural Networks - 1

Firstly, biological neural networks:
there is simply not enough information in all our DNA to specify all the architecture, the connections within our nervous systems.
(nervous system is **part** of our body, whereas
the term 'brain' is usually **contrasted** with body)

So DNA (... with other factors ...) must provide a developmental **'recipe'** which in some sense (partially) determines nervous system structure -- and hence contributes to our behaviour.

Developmental rules for Neural Networks - 2

Secondly, artificial neural networks (ANNs):
we build robots or software agents with (often) ANNs which act as their nervous system or control system.

We can either design (or evolve) the architecture of an ANN explicitly

OR

We can design (or evolve) a 'recipe' for developing the architecture.

Developing ANNs - Gruau

Two of the earliest people to look at developmental programs for ANNs were Frederick Gruau, and Hiroaki Kitano.

Gruau invented 'Cellular Encoding', with similarities to L-Systems, and used this for (eg) evolutionary robotics.

Eg see
Cellular Encoding for Interactive Evolutionary Robotics
Gruau & Quatramaran paper as a gzipped ps file
http://cogslib.informatics.scitech.susx.ac.uk/csr_abs.php?type=csrp&num=425
Or (easier link) as a pdf file: http://drop.io/alergic => csrp425.pdf

Developing ANNs - Kitano

Kitano invented a 'Graph Generating Grammar'

This is a Graph L-System that generates not a 'tree', but a connectivity matrix for a network.

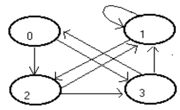
Eg see
Designing Neural Networks Using Genetic Algorithms with Graph Generation System.
Hiroaki Kitano. Complex Systems, 4(4), 1990.
http://www.complex-systems.com/Archive/volume04/issue4/article06/abstract.html

Graph Generating Grammar

Genes encode the **rewrite rules** that are repeatedly applied to a seed, creating a **matrix** of 0s and 1s

If the matrix is 4x4, this can specify all the possible connections between nodes ('neurons') in an ANN with 4 nodes -- specifies the **architecture,** not the weights.

	0	**1**	**2**	**3**
0	0	0	1	1
1	1	0	1	0
2	0	1	0	1
3	1	1	0	0



1=connection, 0 = no connection

From Genotype string to graph

Gene: sabcd b0011 a1c01 c001e d1100 e1001

Rules: s-> ab a-> 1c b->00 c->00 d->11 e->10
cd 01 11 10 00 01

Growth: s ab cd 1c00 0111 0011 1000

	01234567
0	11000000
1	11100000
2	00111111
3	00111111
4	00001111
5	00111111
6	11000000
7	11000000

References

In the seminars we can discuss suitable reading matter and I can make further suggestions

– but I am hoping that **you** will actively seek out appropriate sources and recommend them to each other (and indeed to me !)

Reminder – Seminars (and GA ex)

Current details always available on
http://www.informatics.susx.ac.uk/users/inmanh/easy/alife10/seminars.html.

Please swap times only after consultation – need to swap people to keep numbers balanced
inmanh@susx.ac.uk

… and please hand me your GA exercise for feedback by Tue lec – one possible way of doing the exercise will be posted on website.