Fractals & Statistical Models I

Nonlinear Computational Science in Action by Example

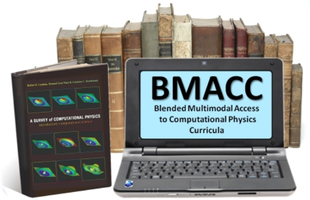
Rubin H Landau

Sally Haerer, Producer-Director

Based on *A Survey of Computational Physics* by Landau, Páez, & Bordeianu

with Support from the National Science Foundation

Course: **Computational Physics II**



Fractal = Fractional Dimension (Math)

Dimension $\stackrel{\text{def}}{=}$? $\neq$ n

- "fractals:" Mandelbrot, IBM
- Geometric fractals: 1 d_f
- Statistical: random, local d_f
- Agree: line, triangle, cube
- Hausdorff–Besicovitch d_f
- Uniform density, side L

$$M(L) \propto L^{d_f}$$

$$\rho = \frac{M(L)}{A} \propto \frac{L^{d_f}}{L^2} \propto L^{d_f - 2}$$

- Works for 1-D, 2-D, 3-D!

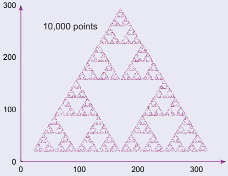
Our 1st Fractal: Sierpiński Gasket

Game of Chance (Randomness) Sierpin.py

1. Draw equilateral triangle.
2. Place dot at $P = (x_0, y_0)$ within
3. Random 1 2, or 3:
4. If 1, $\vec{P} = (\vec{P} + \vec{1})/2$
5. If 2, $\vec{P} = (\vec{P} + \vec{2})/2$
6. If 3, $\vec{P} = (\vec{P} + \vec{3})/2$
7. Repeat 10,000 $\times$ new P

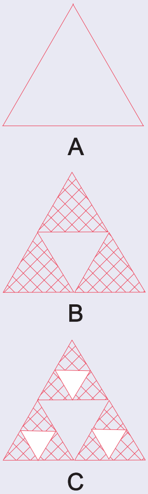
$$(x_{k+1}, y_{k+1}) = \frac{(x_k, y_k) + \vec{n}}{2}$$

$$n = \text{int}\,(1 + 3r_i)$$



Determine d_f via $\rho = CL^{d_f - 2}$

Geometric Sierpiński Gasket (Midpoints)



- **Self-similar:** part $\sim$ whole
- Dots, $m_{dot} = 1$, or
- $\ln \rho \propto (d_f - 2) \log L$
- $\rho_{new} = \frac{3}{4} \rho_{old}$ (fills space less)

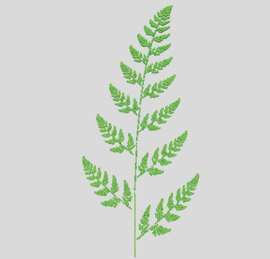
$$d_f - 2 = \frac{\Delta \log \rho}{\Delta \log L}$$

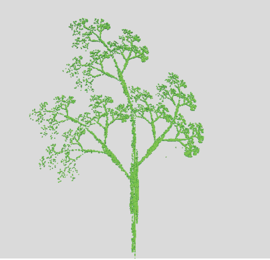
$$d_f = 2 + \frac{\Delta \log \rho(L)}{\Delta \log L}$$

$$= 2 + \frac{\log 1 - \log \frac{3}{4}}{log 1 - \log 2}$$

$$\simeq 1.585$$

Example 2: Beautiful Plants





- Nature + chance $\Rightarrow$ high regularity & symmetry?
- EG fern or tree = beautiful, graceful & random?
- Key: self-similar, fractal
- Simple random algorithm $\Rightarrow$ beauty?
- If algorithm $\rightarrow$ ferns $\Rightarrow$ algorithm $\exists$ fern?

Self-Affine Connection (Theory)

Point–Point Relation $\Rightarrow$ Self Similarity

- Recall Sierpiński gasket:

$$(x_{k+1}, y_{k+1}) = (x_k, y_k)/2 + (a_n, b_n)/2$$

- s = scaling factor $= \frac{1}{2}$, $\quad (a_n, b_n)$ = translation
- $s > 0$ = amplification, $s < 0$ = reduction

$$\begin{aligned}
(x', y') &= s(x, y) && \text{(General Scaling)} \\
(x', y') &= (x, y) + (a_x, a_y) && \text{(General Translation)} \\
x' &= x\cos\theta - y\sin\theta, && \text{(Rotation)} \\
y' &= x\sin\theta + y\cos\theta && \text{(Rotation)}
\end{aligned}$$

- Affine Connection = point–point scale + rotate + translate

Barnsley's Fern (Fern3D.py)

CODE

Affine Connection + Random (2D)

$$(x, y)_{n+1} = \begin{cases} (0.5, 0.27y_n), & \text{with 2\% probability}, \\ (-0.139x_n + 0.263y_n + 0.57 \\ \quad 0.246x_n + 0.224y_n - 0.036), & \text{with 15\% probability}, \\ (0.17x_n - 0.215y_n + 0.408 \\ \quad 0.222x_n + 0.176y_n + 0.0893), & \text{with 13\% probability}, \\ (0.781x_n + 0.034y_n + 0.1075 \\ \quad -0.032x_n + 0.739y_n + 0.27), & \text{with 70\% probability}. \end{cases}$$

- Start $(x_1, y_1) = (0.5, 0.0)$
- Repeat iterations
- Not completely self-similar
- Δ d_f different parts
- Stem = compress fronds
- Nonlinear indirectly

Barnsley's Fern (Fern3D.py)

CODE

Affine Connection + Random

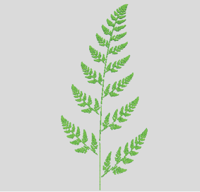
$$\text{Select with probability} \quad \mathcal{P} = \begin{cases} 2\%, & r < 0.02, \\ 15\%, & 0.02 \leq r \leq 0.17, \\ 13\%, & 0.17 < r \leq 0.3, \\ 70\%, & 0.3 < r < 1. \end{cases}$$

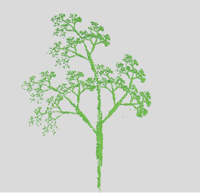
Combined rules (program):

$$(x, y)_{n+1} = \begin{cases} (0.5, 0.27 y_n), & r < 0.02, \\ (-0.139 x_n + 0.263 y_n + 0.57 & \\ \quad 0.246 x_n + 0.224 y_n - 0.036), & 0.02 \leq r \leq 0.17, \\ (0.17 x_n - 0.215 y_n + 0.408 & \\ \quad 0.222 x_n + 0.176 y_n + 0.0893), & 0.17 < r \leq 0.3, \\ (0.781 x_n + 0.034 y_n + 0.1075, & \\ \quad -0.032 x_n + 0.739 y_n + 0.27), & 0.3 < r < 1. \end{cases}$$

Self-Affine Trees (Tree.py)

CODE





$$
(x_{n+1}, y_{n+1}) = \begin{cases}
(0.05x_n, 0.6y_n), & \text{10\% probability}, \\
(0.05x_n, -0.5y_n + 1.0), & \text{10\% probability}, \\
(0.46x_n - 0.15y_n, 0.39x_n + 0.38y_n + 0.6), & \text{20\% probability}, \\
(0.47x_n - 0.15y_n, 0.17x_n + 0.42y_n + 1.1), & \text{20\% probability}, \\
(0.43x_n + 0.28y_n, -0.25x_n + 0.45y_n + 1.0), & \text{20\% probability}, \\
(0.42x_n + 0.26y_n, -0.35x_n + 0.31y_n + 0.7), & \text{20\% probability}.
\end{cases}
$$

Good Time for a Break