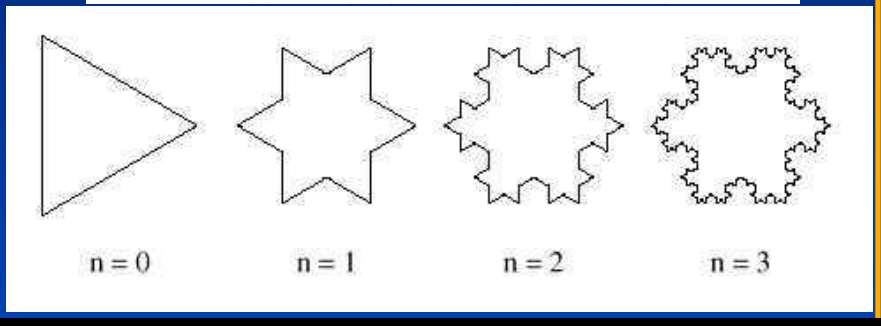
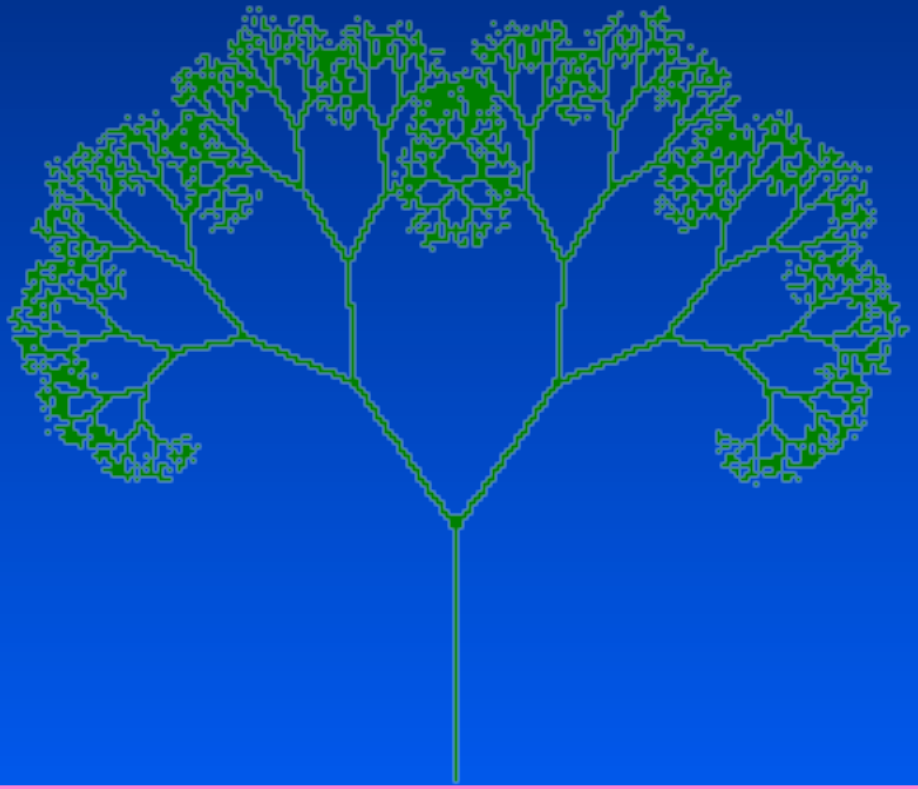
Procedural Textures and Models

Procedurally-generated: Fractals

- Data amplification
- Self-similarity (statistical)

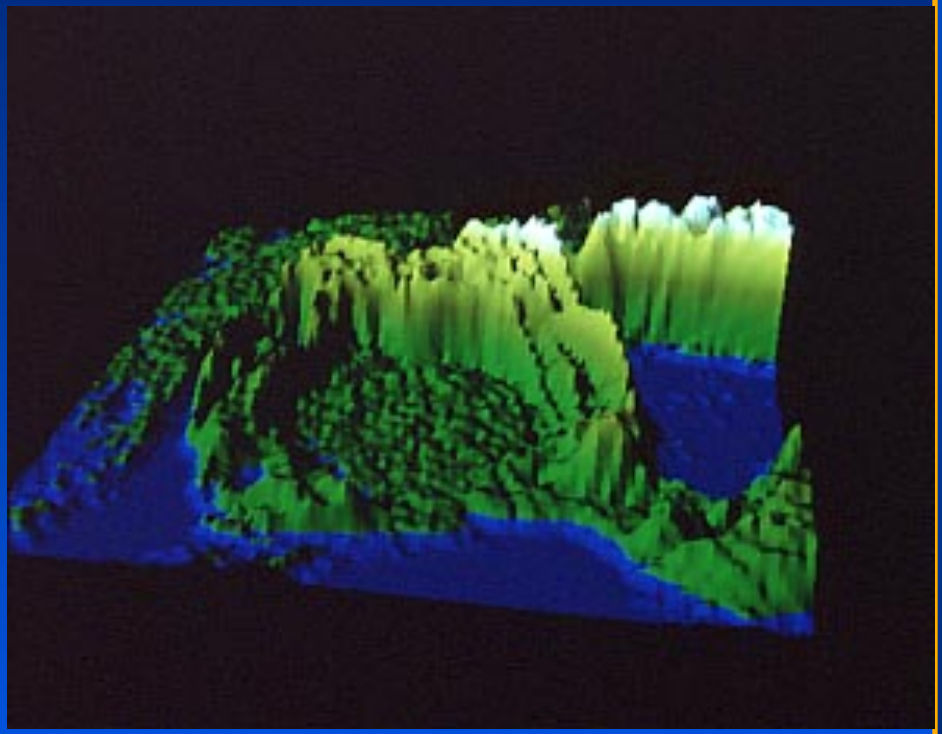


- Often occur in nature



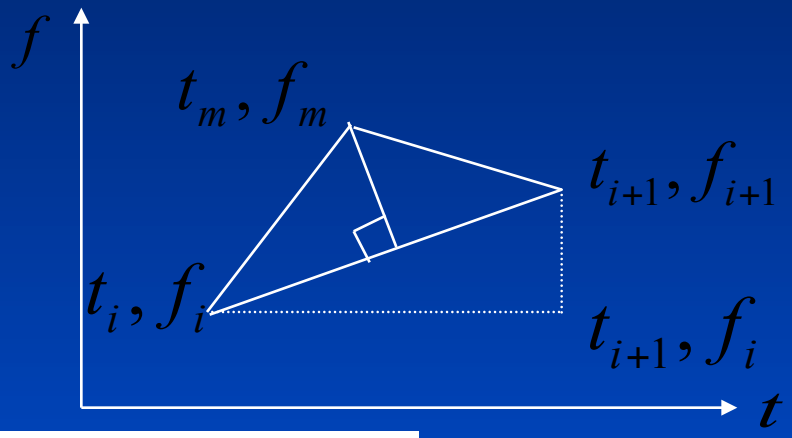
Terrain Generation

- Subdivision
- Split polygon into smaller polygons
- Perturbation proportional to size of polygon being perturbed



Example of Line

- Displacement dependent on
  - Roughness
  - Scale given by f_{i+1}-f_i
  - Random number

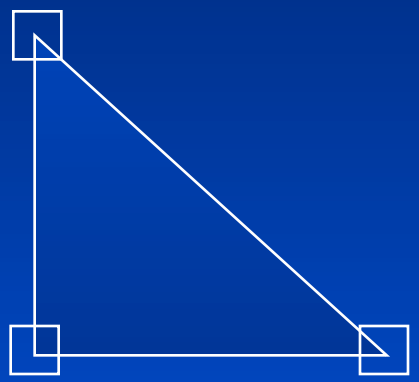


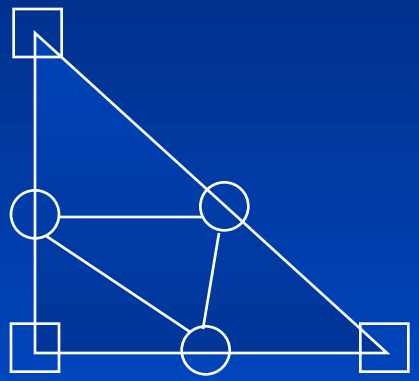
$$t_m = \frac{t_i + t_{i+1}}{2} - roughness * (f_{i+1} - f_i) * Rand$$

$$f_m = \frac{f_i + f_{i+1}}{2} + roughness * (f_{i+1} - f_i) * Rand$$

- Displacement not dependent on coordinate system used

Polygon





- Connect midpoints
- Displace midpoints along normal of original polygon

Internal consistency

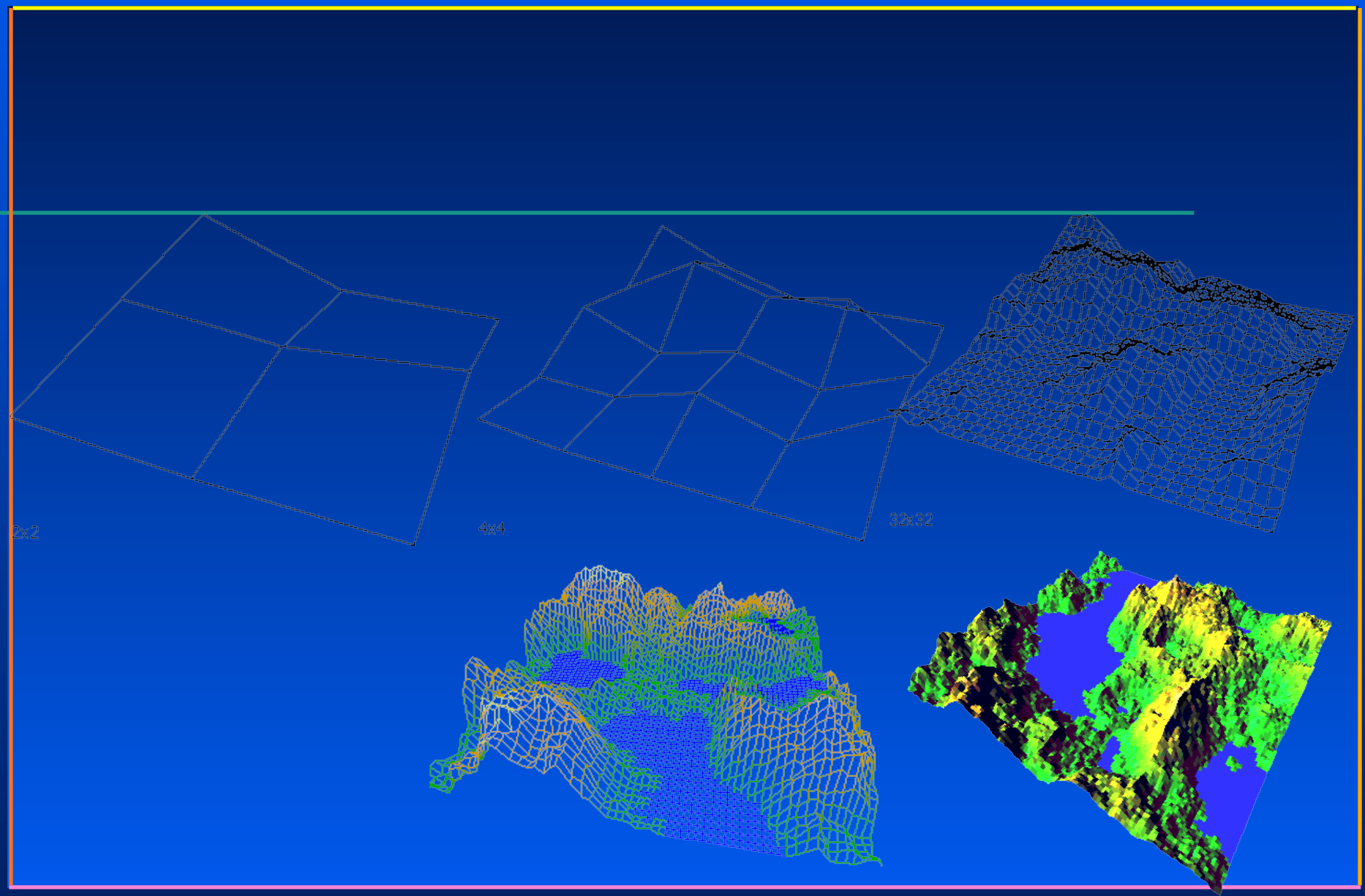
- Orientation independence
  - Random number do not change as object/ viewpoint change
- Coarse detail same if rendered at higher resolution
  - Same random number generated in the same order at given level of subdivision
- Needed since objects are often created at runtime

External consistency

- Same random displacement for adjacent triangles
- To prevent “cracks” direction of displacement same for adjacent triangles

Termination criterion

- Shade of subdivision same as shade w/o subdivision
  - Size of subdivision approximately equal to pixel size
- Will added detail result in aliasing?
  - Having much detail not necessarily good



Perlin noise

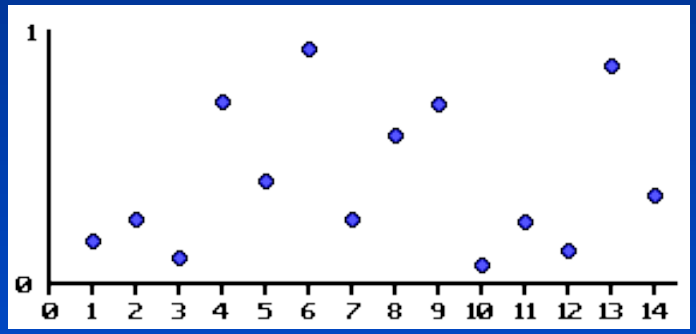
- Ken Perlin won the 1997 Academy Award for Technical Achievement for this
- Attempt to generate a wide variety of textures from noise and turbulence
  - Based on using characteristics of noise occurring in nature (e.g. 1/f)
  - Effective use is based on creative use of the noise
- First (and most commonly) used in solid textures but need not be limited to solid textures

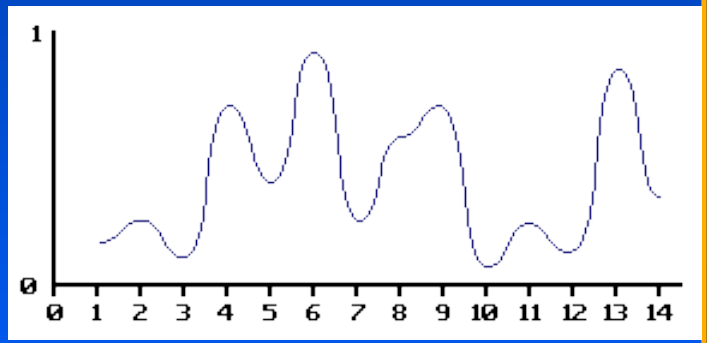
Solid noise

- Basis for all other noise
- Statistical invariance under rotation and translation
- 3D noise
- Narrow band pass limit in frequency
  - Known frequency range-useful for anti-aliasing

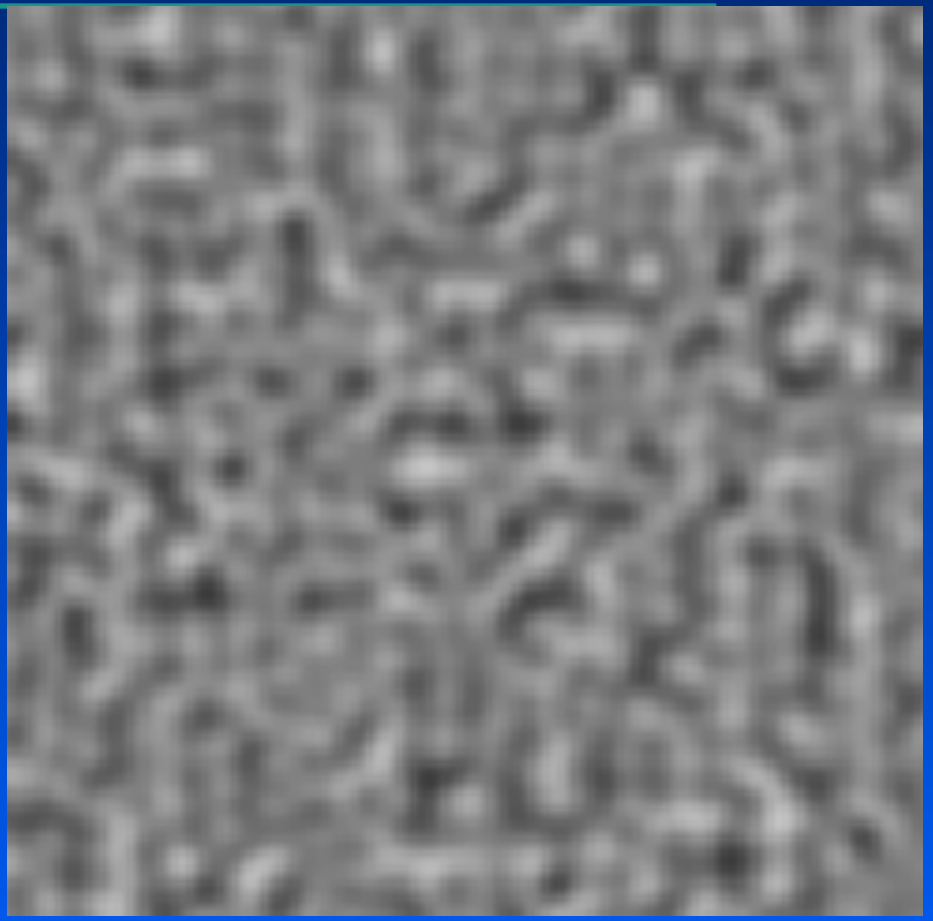
Perlin noise

- For integer coordinate points, assign random number
- Interpolate to get rest
  - Gives it continuity (e.g. cubic)
  - Why is this band pass limited?
- In 3-D, 3-D lattice of random values, bilinear interpolation or bicubic interpolation
- Modulo arithmetic used to wrap around the noise





- Perlin noise mapped to grayscale



Perlin Turbulence

- $$turbulence(x) = \sum_{i=0}^{k} abs(\frac{noise(2^i x)}{2^i})$$
  - $noise(2^i x)$ gives the noise that has been "compressed in the x-direction" by 2^i: signal with frequency proportional to 2^i
  - Division by 2^i makes sure that the amplitude is proportionally reduced
- Truncate k when $\frac{1}{2^{k+1}} < \text{size of pixel}$
  - since noise "wavelength" is approximately lattice size and inversely proportional to frequency

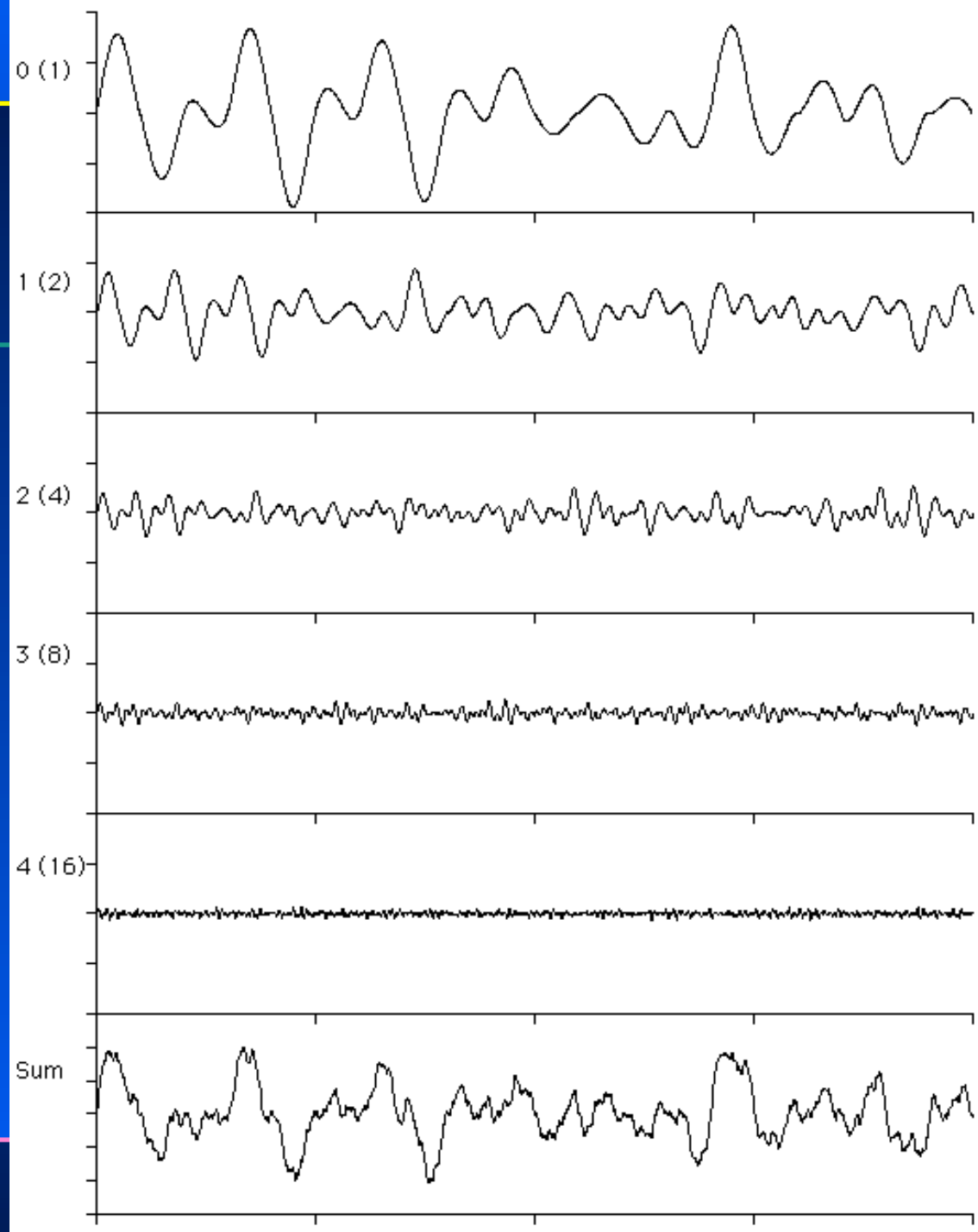
- Each term higher frequency, lower amplitude

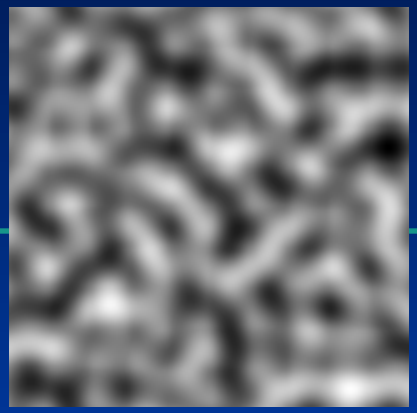
$$noise(x) + \frac{noise(2x)}{2} + \frac{noise(4x)}{4} + \cdots$$

Twice the freq. **4X the freq.**

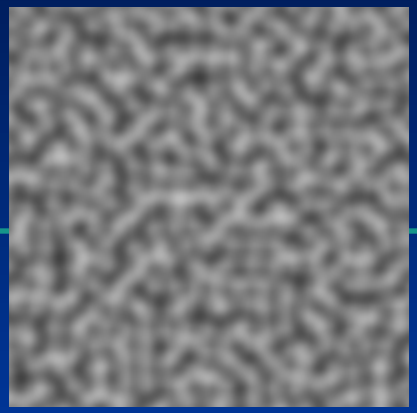
- Amplitude is inverse proportional to frequency
  - 1/f noise
  - Statistically self similar

- First five terms of turbulence and the sum

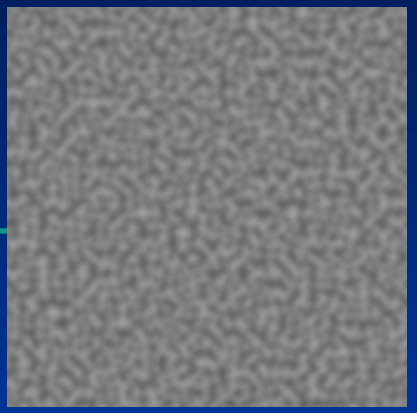




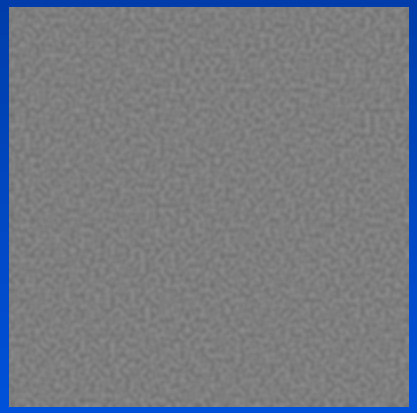
0 (1)



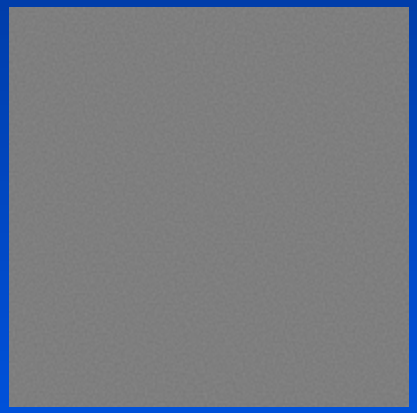
1 (2)



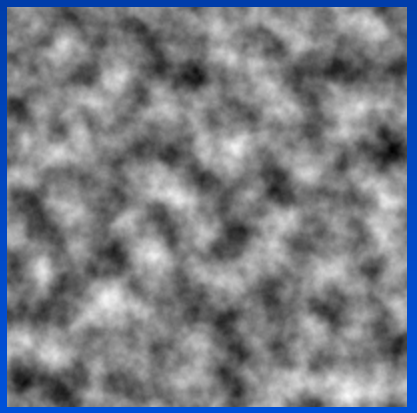
2 (4)



3 (8)



4 (16)

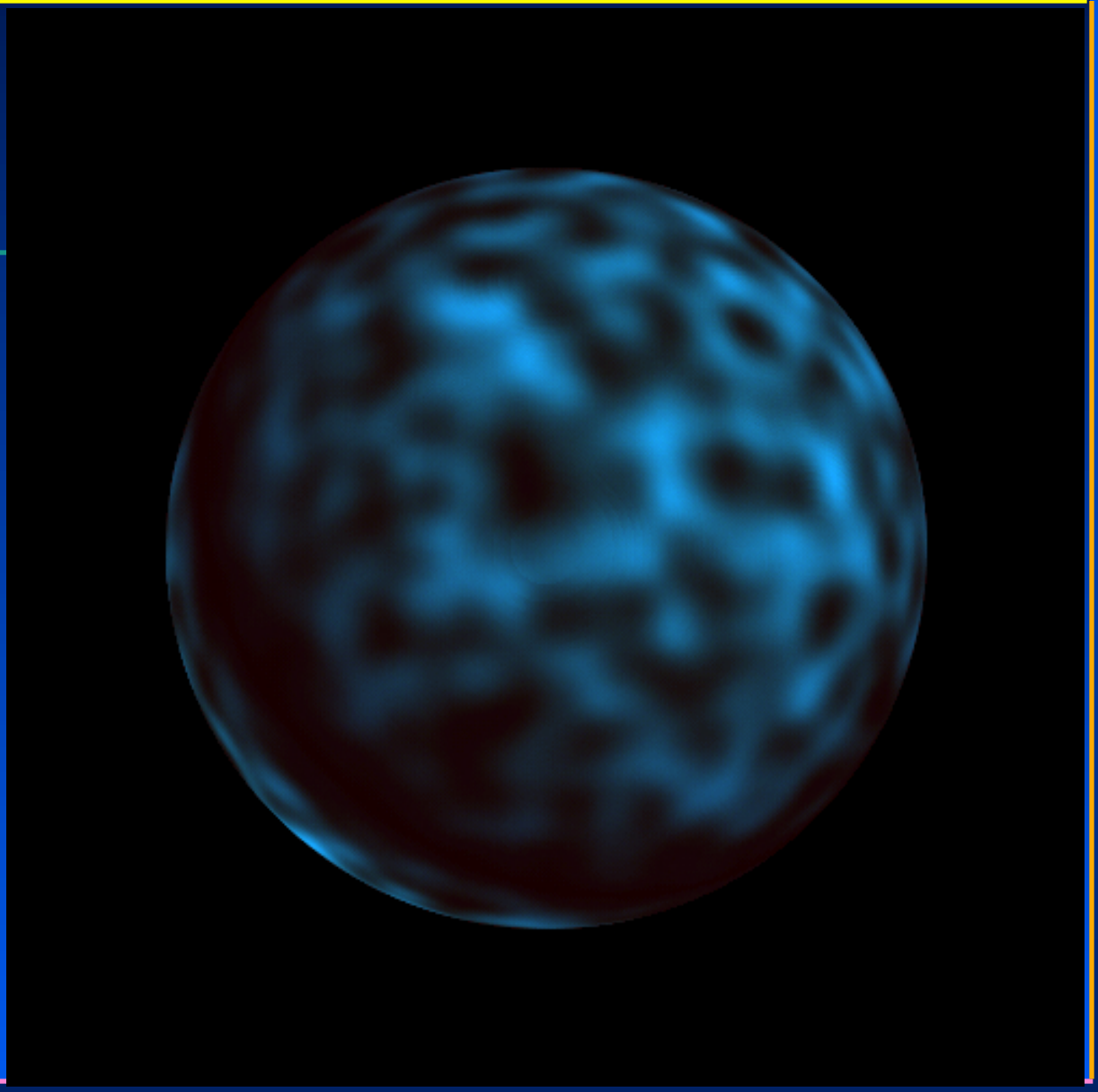


Sum

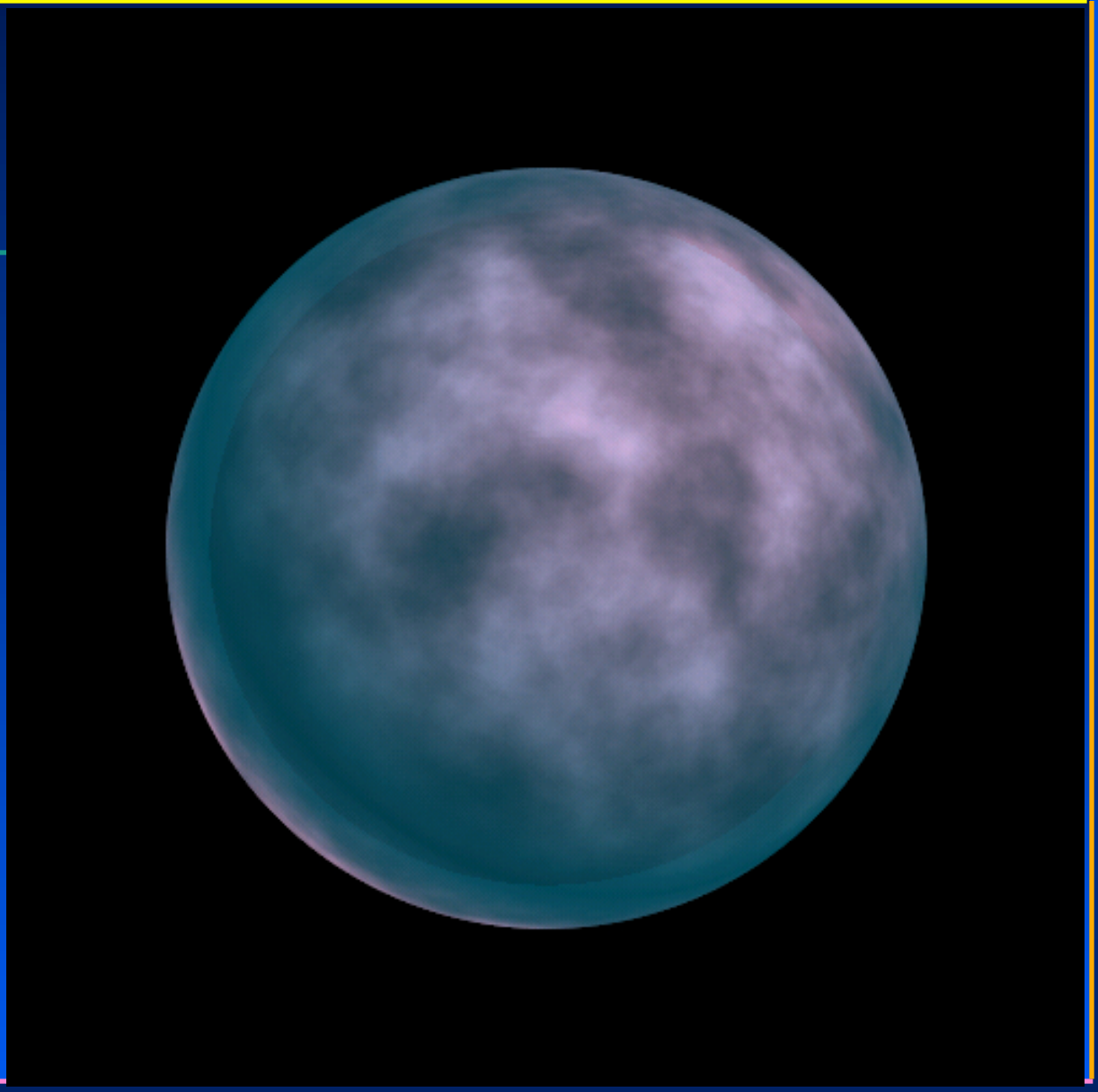
Using Turbulence

- Start with basic, first-order structure
- Perturb basic structure using turbulence function
- Since simulate the effect of turbulence not the physics behind it, creative use necessary

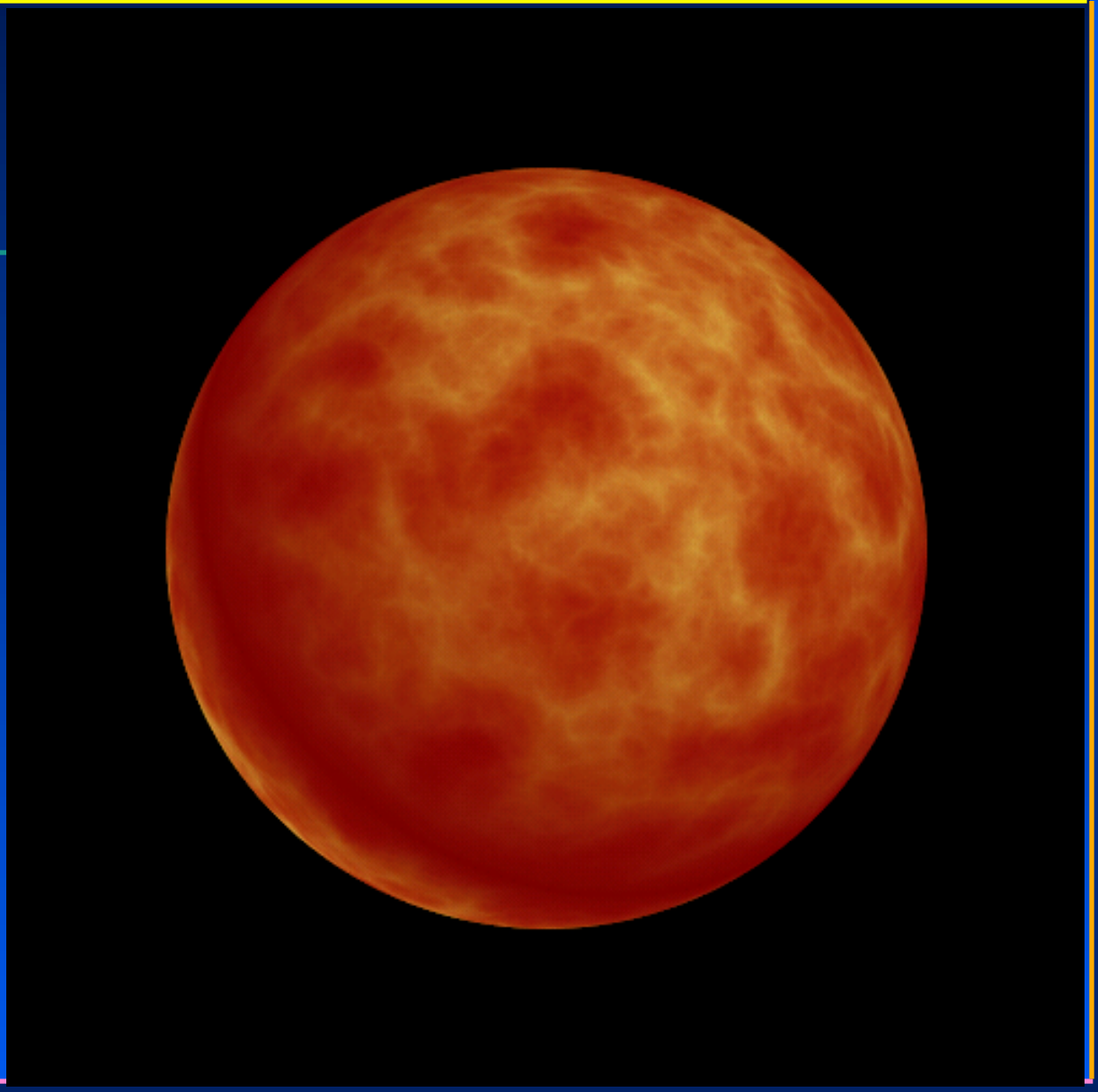
- Noise



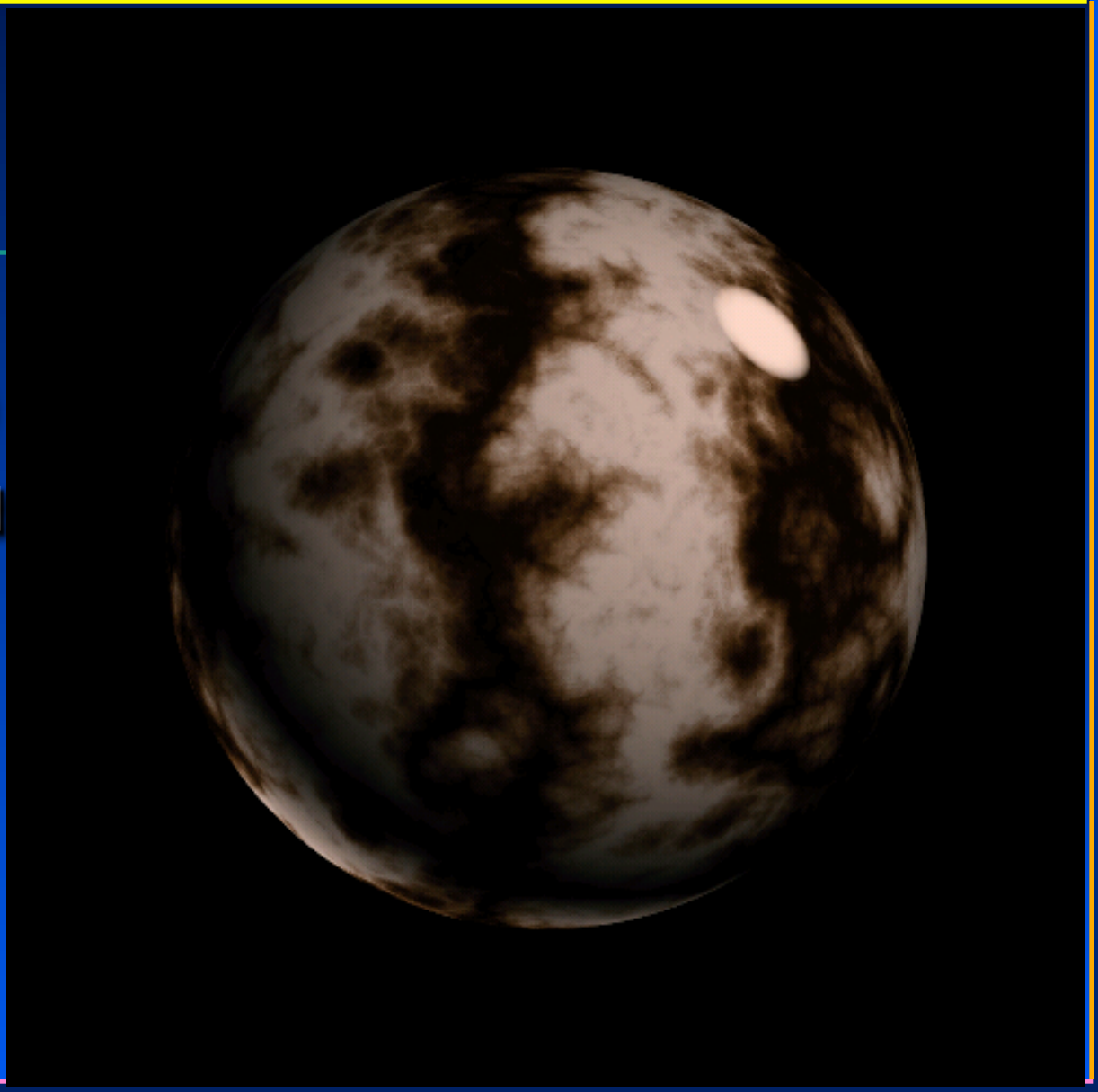
- Sum 1/f(noise)



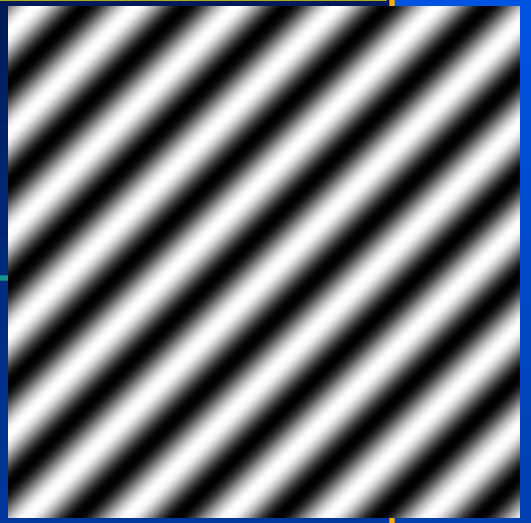
- Sum 1/f(|noise|)
- Crease at zero crossing



- Sin (x+ sum 1/f(|noise|))
- Turbulence used to phase-shift simple sine function



Marble

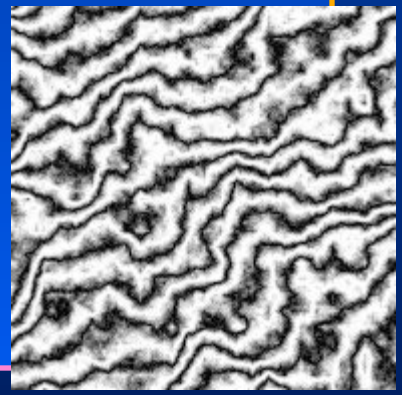


- Basic marble:

$$basic\_marble(x) = \text{marble\_color}(\sin(x))$$

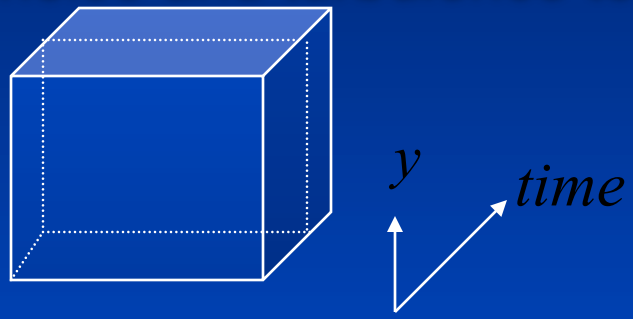
- Marble-color map scalar value to color (a palette) vector
- Marble:

$$marble(x) = \text{marble\_color}(\sin(x + \text{turbulence}(x)))$$



Animating Turbulence

- Can extend dimension of noise and turbulence to time
- For 2-D and time:

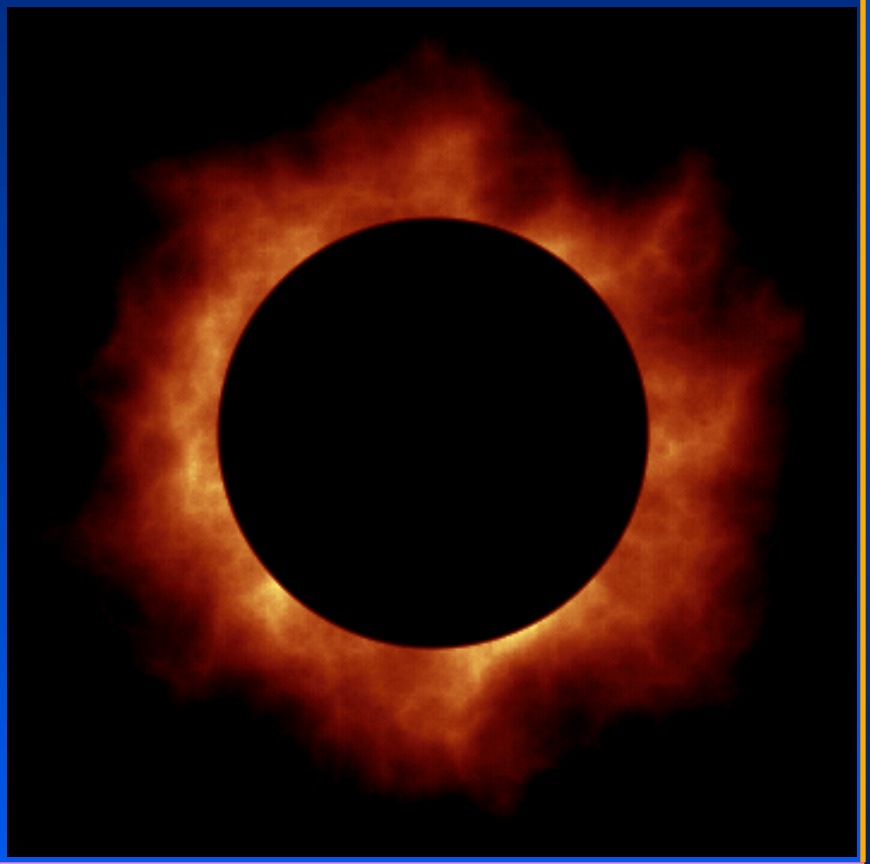


♪

♪

  - Moving plane in time axis causes general change in texture
  - Moving plane in –y direction causes static texture to move upward
  - Combination of two causes a varying texture to move upward

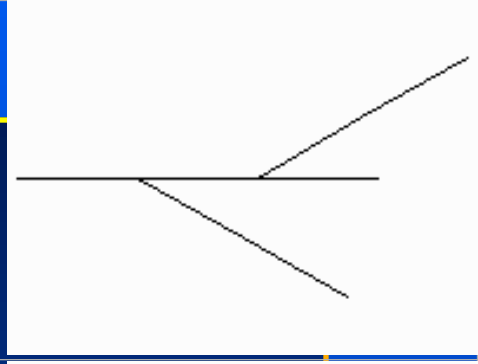
- Corona: moving in time dimension



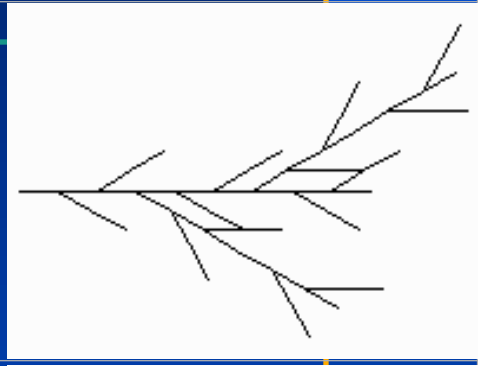
Plants

- Self-similarity (like fractals)
- Grammars and graftals
- Rules to generate strings of expressions (“genotype”)
- Expressions then represented geometrically (“phenotype”)
- L-systems

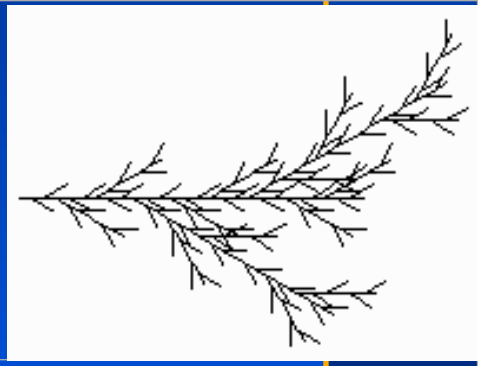
Gen. 1



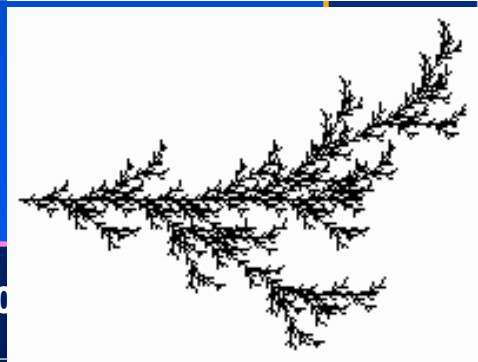
Gen. 2

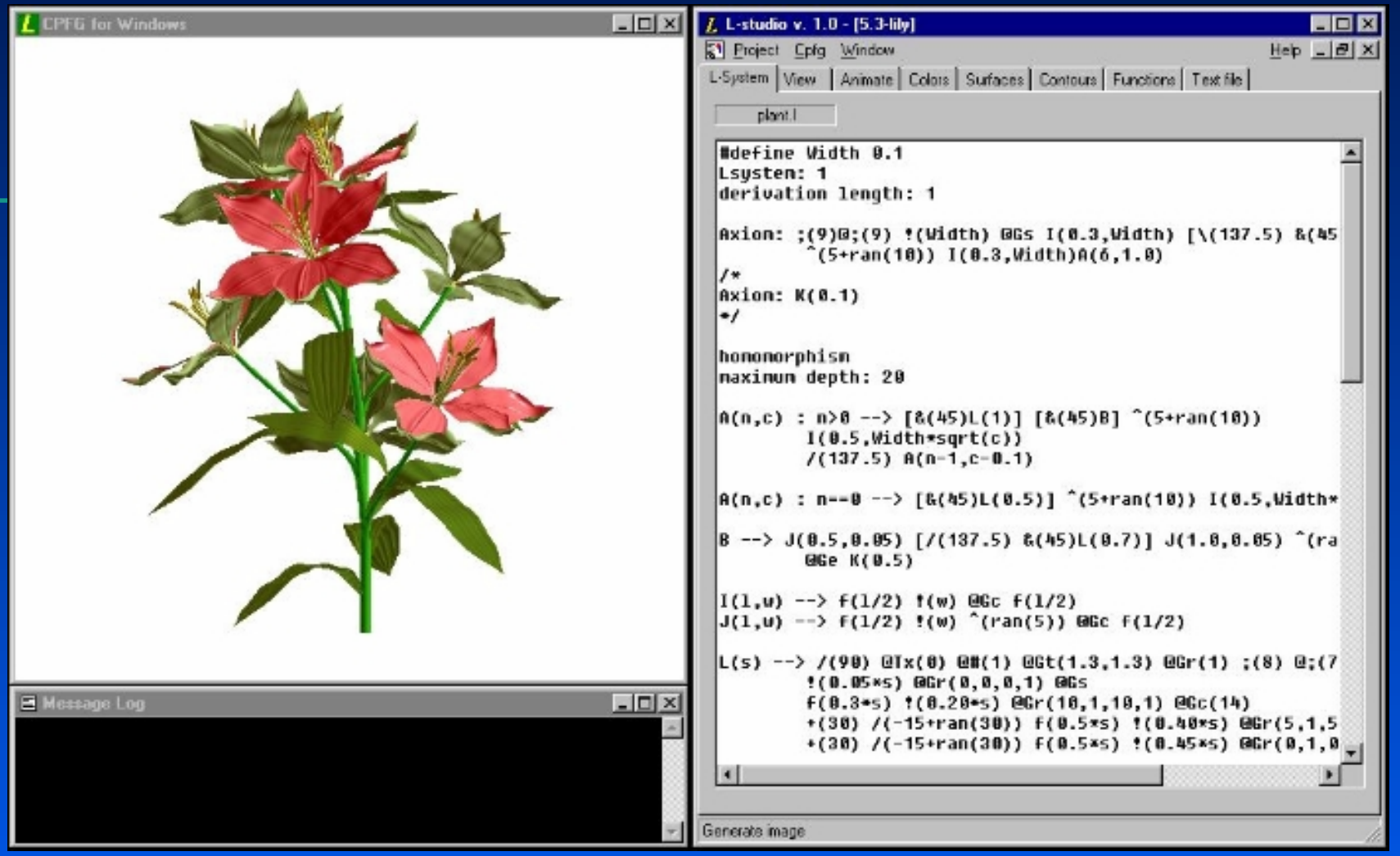


Gen. 3



Gen. 4



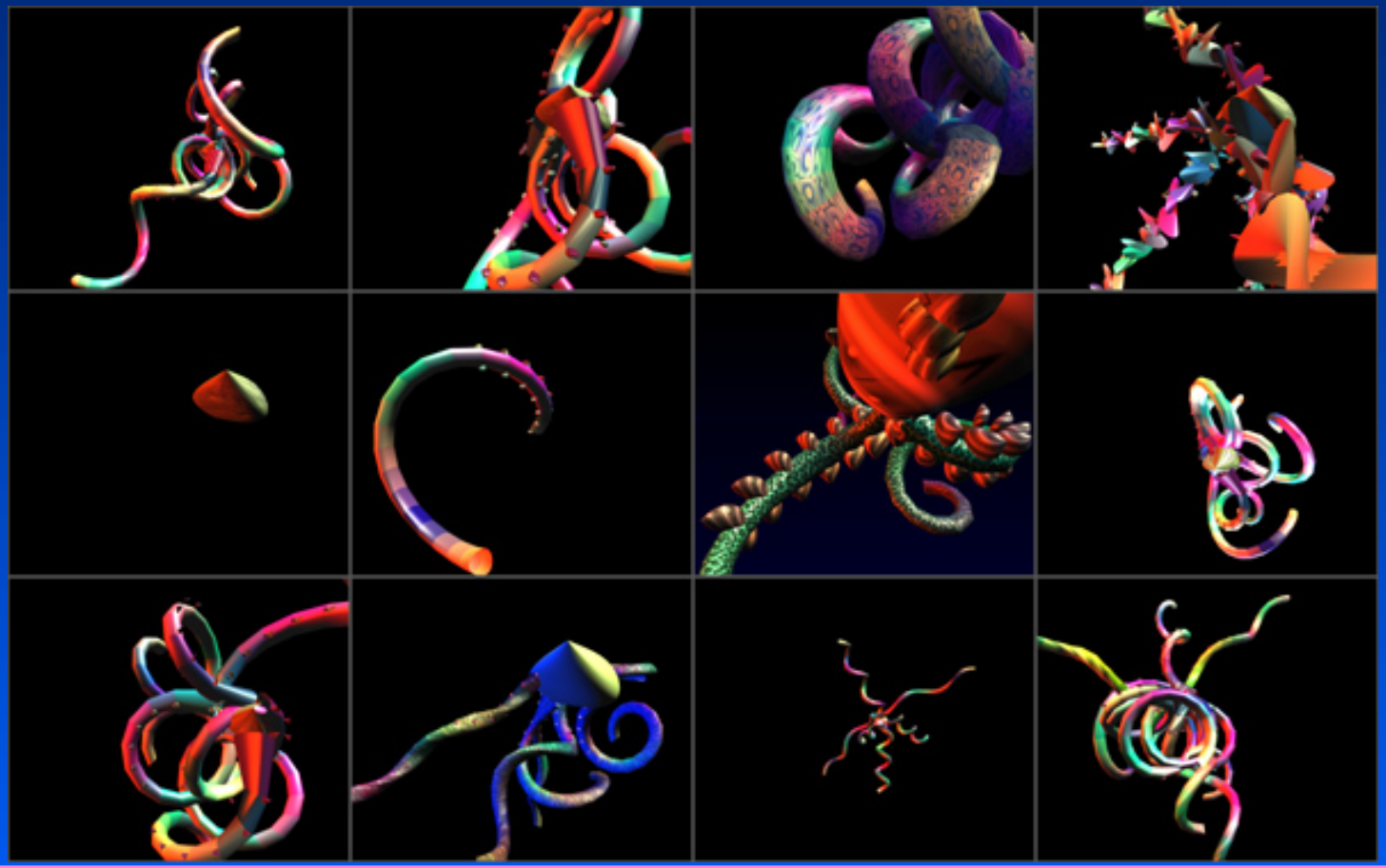


http://www.cpsc.ucalgary.ca/~pwp/

Evolutionary procedural modeling

- Evolve genotype: reproduction, mutation
- Fitness metric-evaluate phenotype
- Phenotype: realization of genotype

Evolutionary modeling
Karl Sims (http://genarts.com/)



Sims (siggraph ’91)

- Genotype is LISP expression
- Composed of : + , -, *, /, mod, round, min, max, noise, sin, cos, …
- For a number of arguments (include x, y) return color
- Mutation: e.g. Change function, replace random parameters, etc.
- Mating : e.g. Crossover-change node of a parent with another parent