GPU Programming

Course web page:
http://goo.gl/EB3aA

Outline

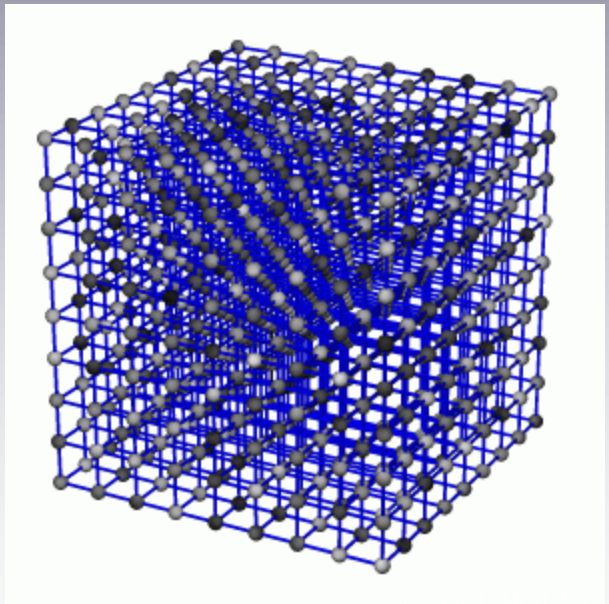
- More GLSL examples
- Perlin noise
- HW #4

Noise as a Texture Generator

- Easiest texture to make: Random values for texels
  - `noise(x, y) = random()`
- If `random()` has limited range (e.g., [0, 1]), can control maximum value via **amplitude**
  - `a * noise(x, y)`
- But the results usually aren't very exciting visually

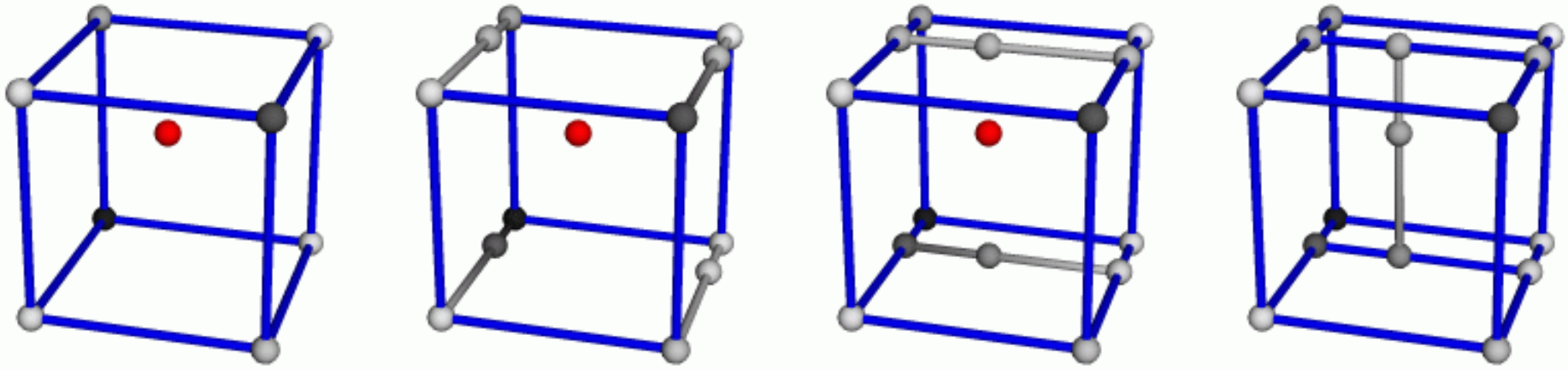
3-D Noise (aka Perlin noise)

- 3-D or solid texture has value at every point (x, y, z)
- This makes texture mapping easy :)
- Simple solid texture generator is noise function on lattice:
  - `noise(x, y, z) = random()`
- For points in between, we interpolate

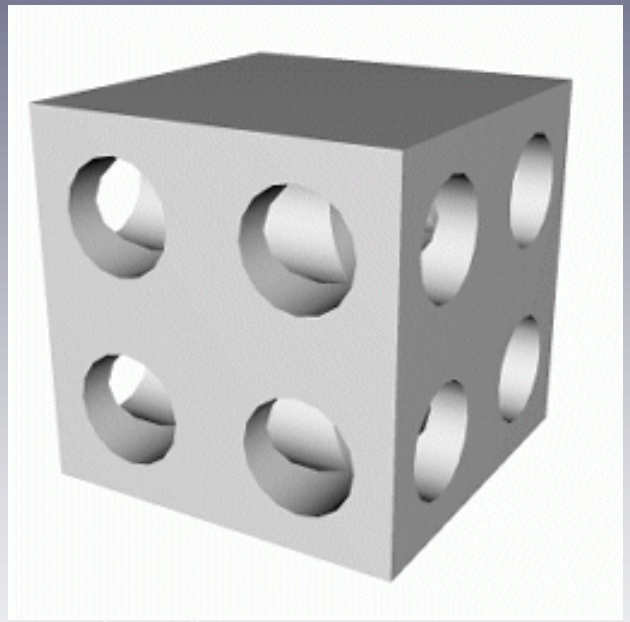


3-D Noise Interpolation

- `f(x, y, z)` can be evaluated at non-lattice points with a straightforward extension of 2-D bilinear interpolation (to **trilinear interpolation**)
- Other interpolation methods (quadratic, cubic, etc.) also applicable
- All of these are approximations of smoothing filters such as Gaussians

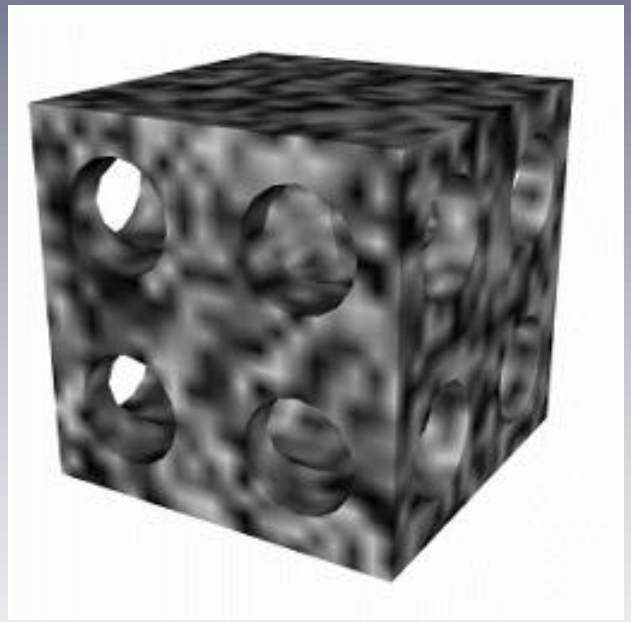


3-D Noise Texturing: Examples

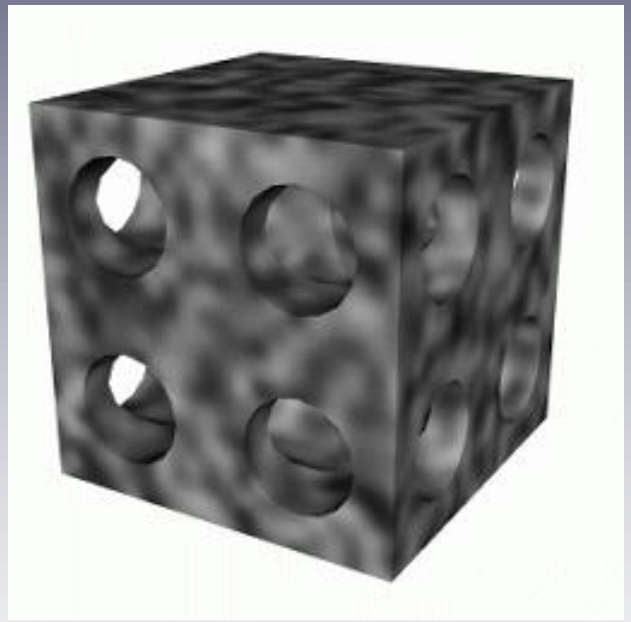


courtesy of L. McMillan

Original object



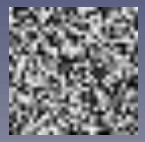
Noise with trilinear smoothing



Triquadratic noise

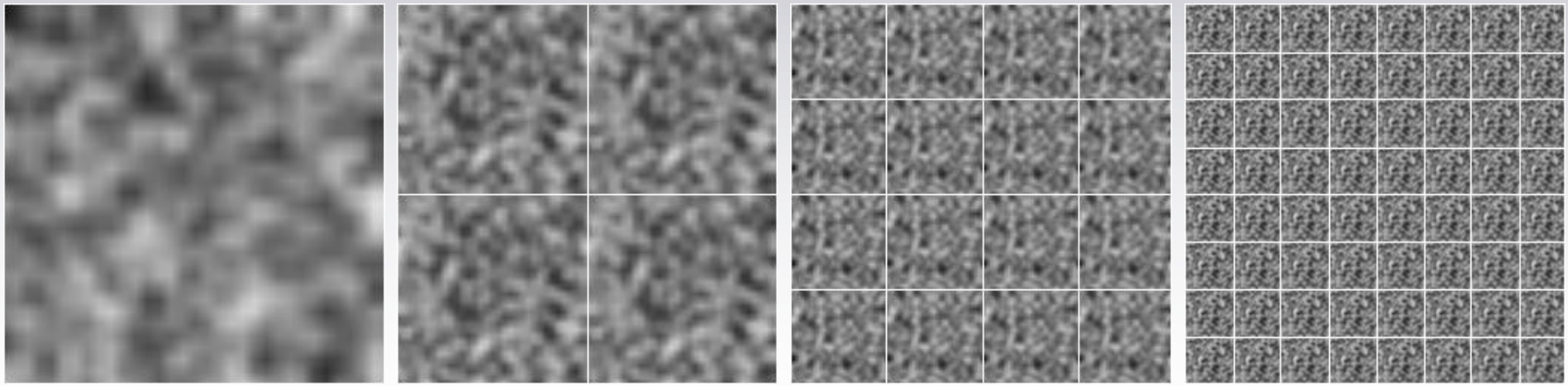
Noise Frequency

- By selecting larger spaces between lattice points, we are increasing the magnification of the noise texture and hence reducing its frequency



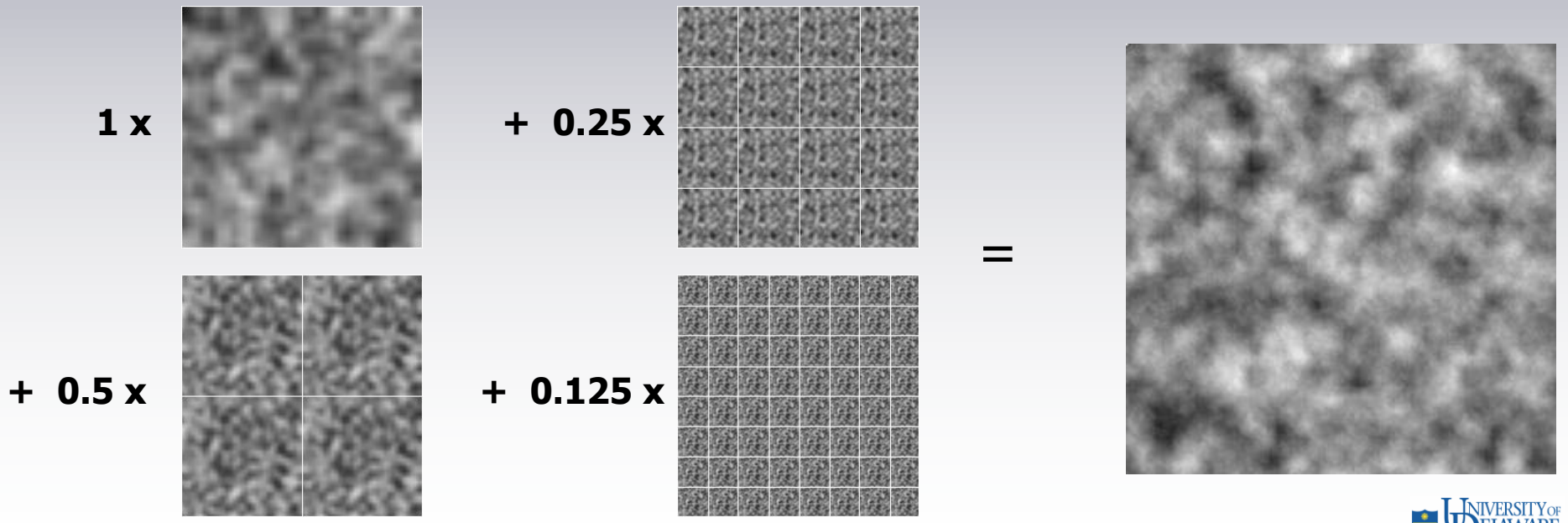
Original noise

Smoothed noise

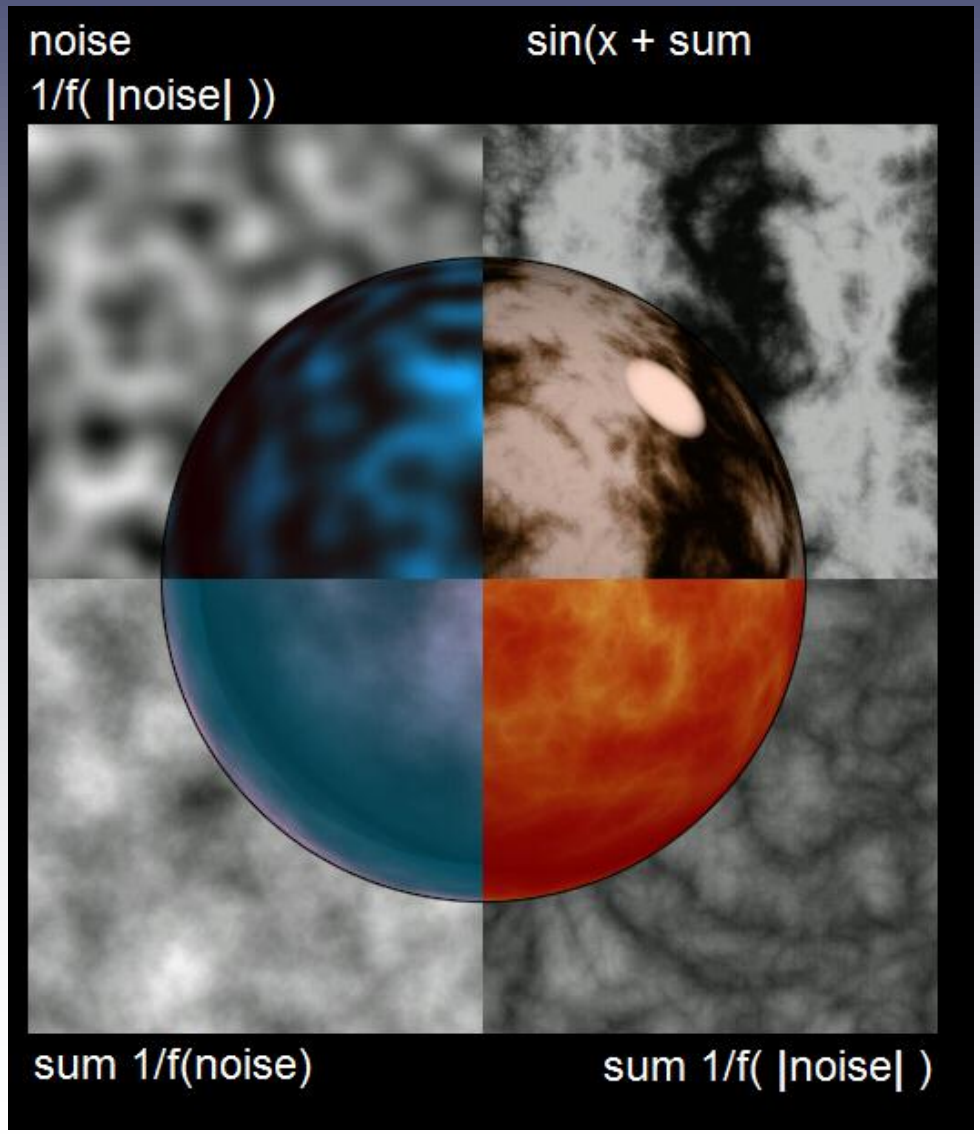


Perlin Noise for Turbulence

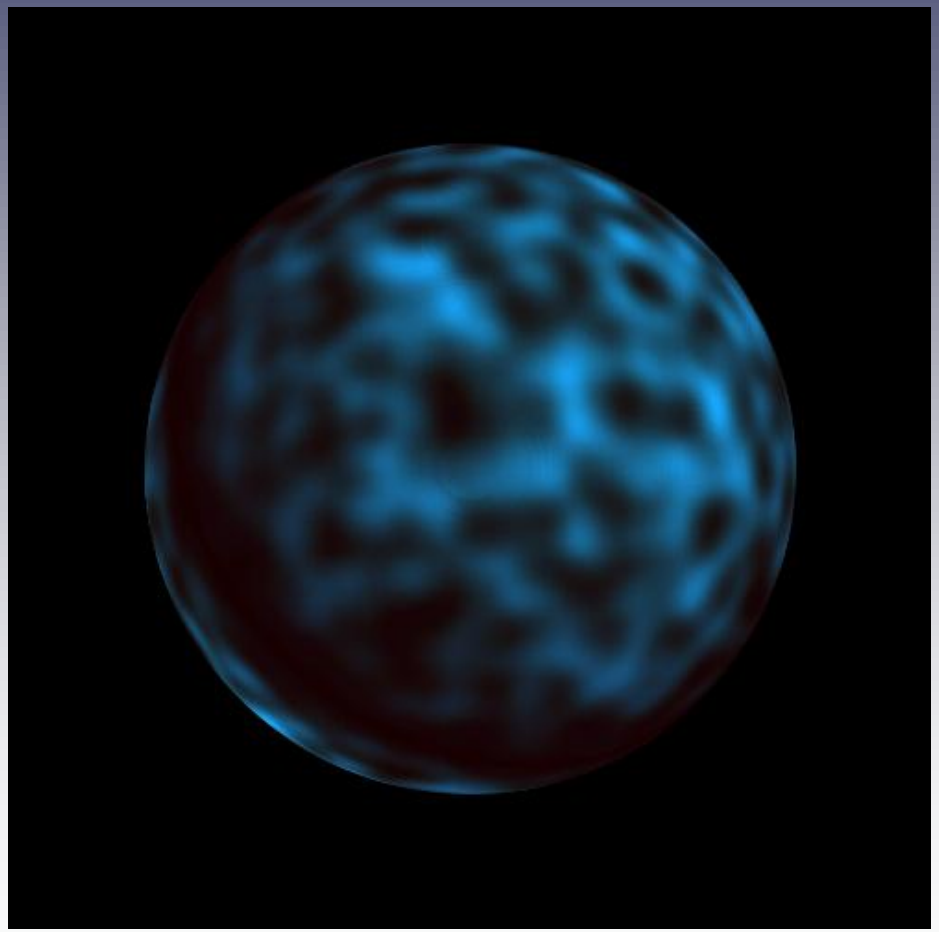
- Fractal noise: Many frequencies present, looks more natural
- Can get this by **summing** noise at different magnifications
- $\mathbf{turb(x, y, z) = \Sigma_i\ a_i * noise_i(x, y, z)}$
- Typical (but totally adjustable) parameters:
  - Magnification doubles at each level (**octave**)
  - Amplitude drops by half



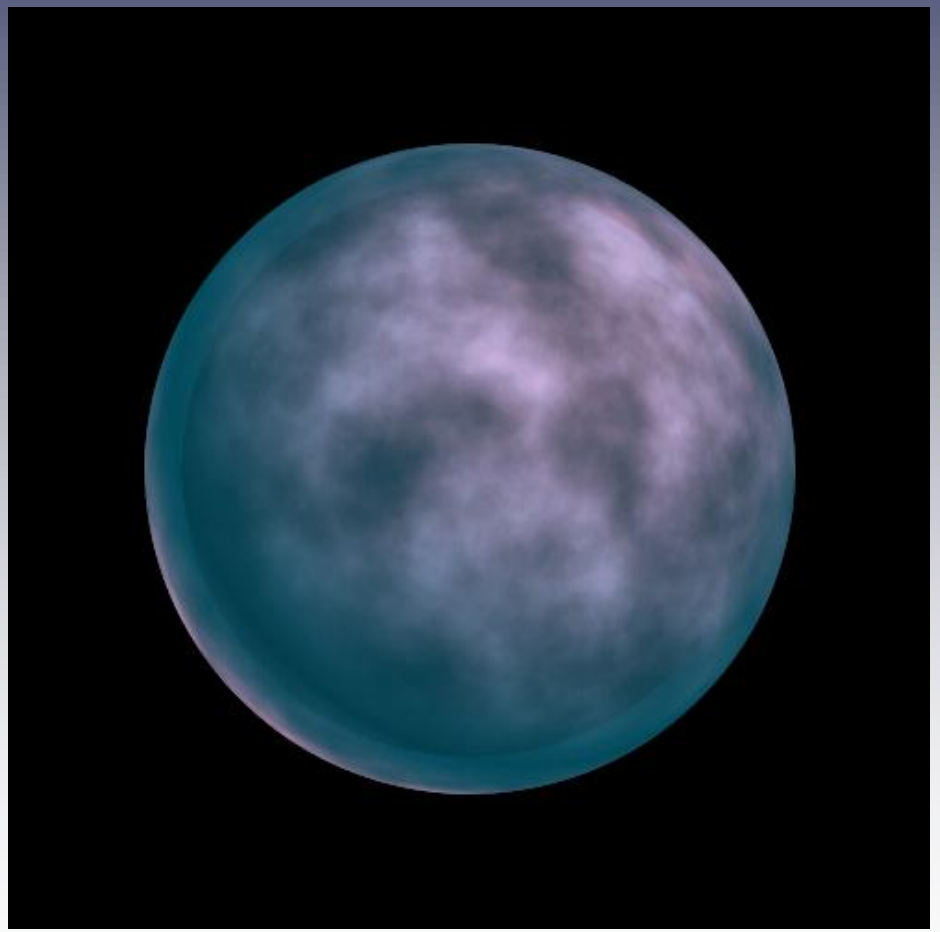
Variations on Perlin noise



Noise Variations



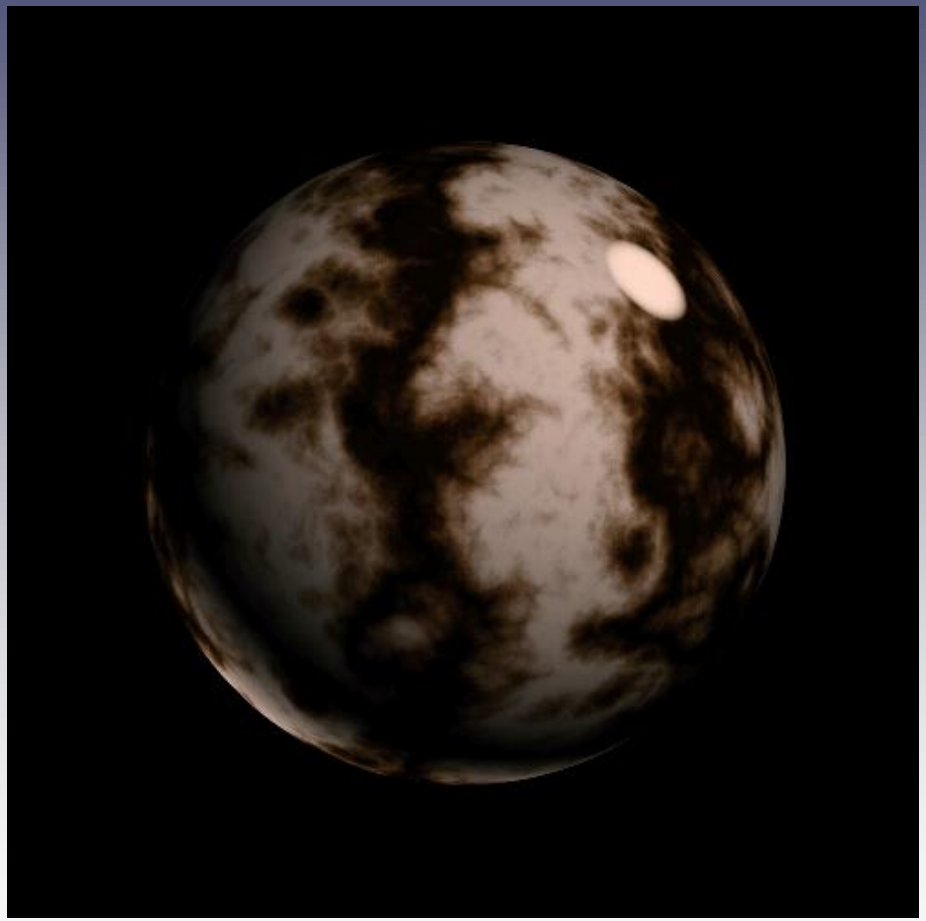
Pure Perlin noise



Turbulence

Noise Variations

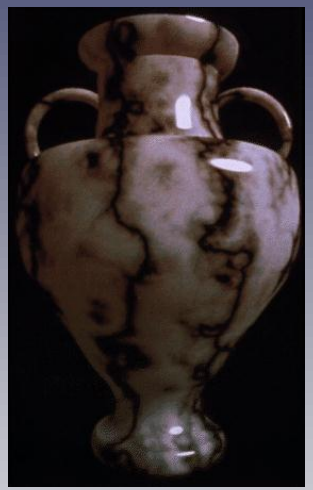
Turbulence with absolute values of noise

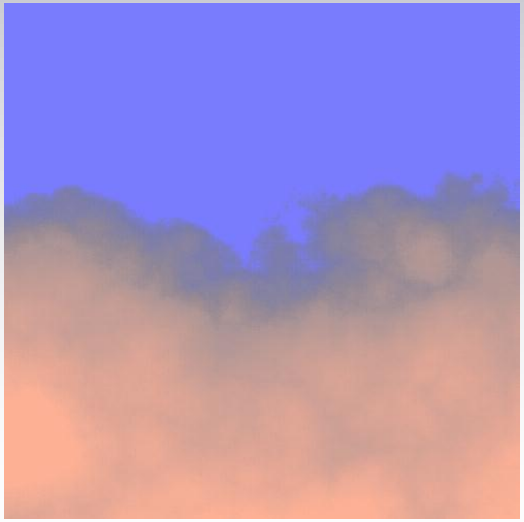


Phase shift: sin(x + noise())

Perlin Noise: Applications

- 2-D noise:
  - Traditional "wrappable" textures; good for:
    - Clouds
    - Water
    - Bump, specularity maps
  - Height maps—e.g., fractal terrain
- 3-D noise
  - Solid textures such as marble, D clouds
  - Animated 2-D textures (i.e., with time as 3rd dimension)
- 4-D noise
  - Animated solid textures





http://www.noisemachine.com/talk1/imgs/clouds500.html