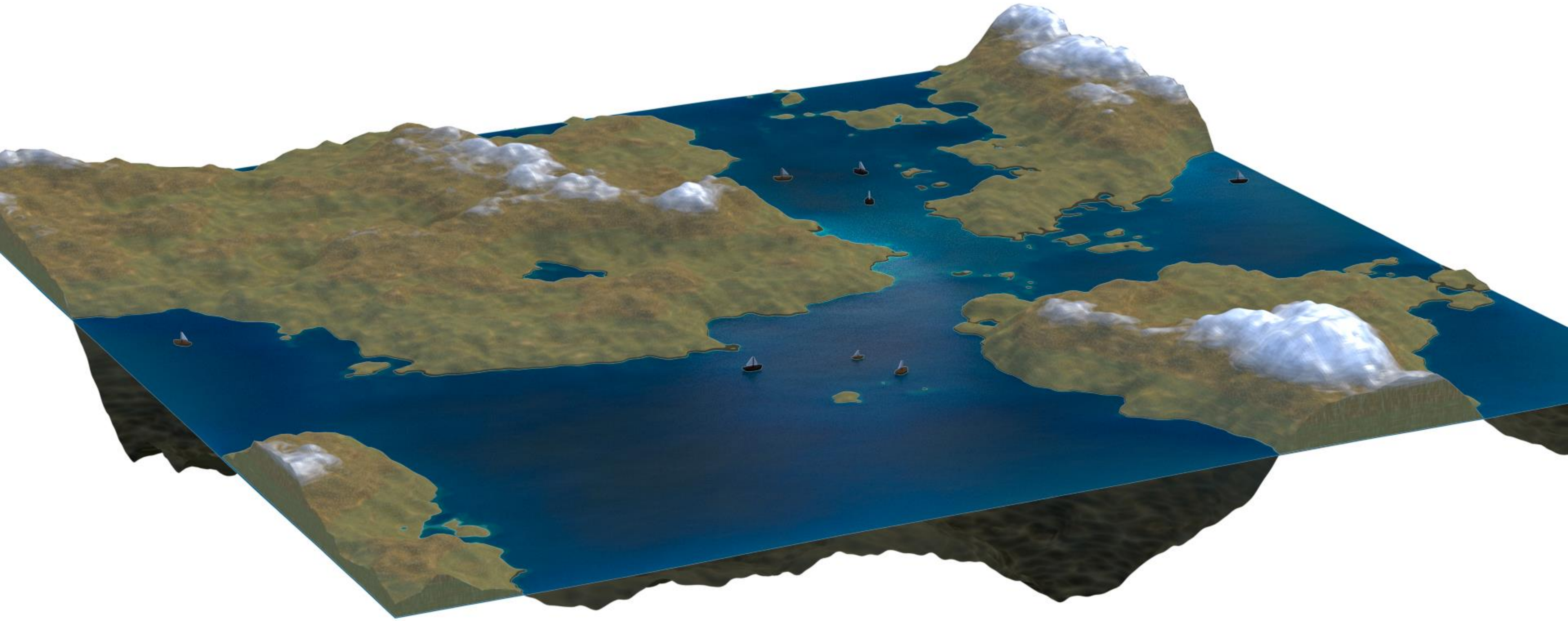
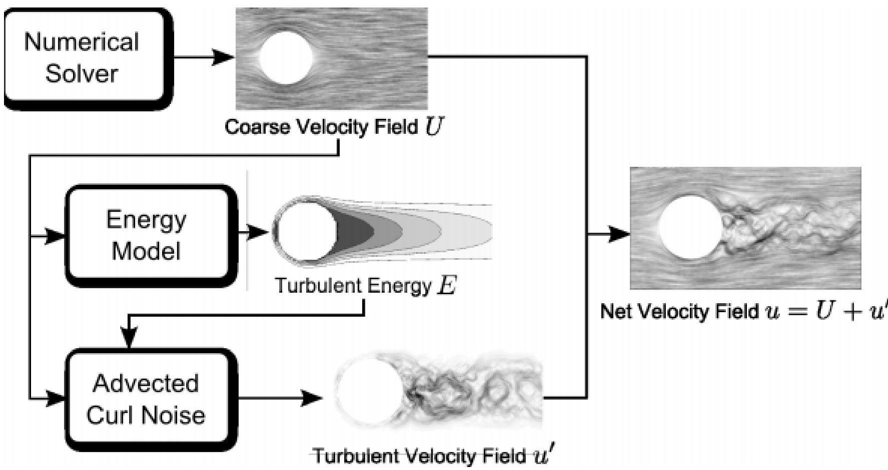
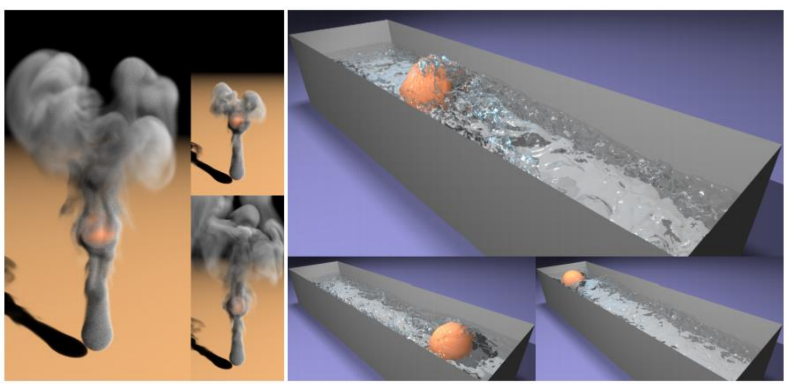
Procedural terrain and texture generation



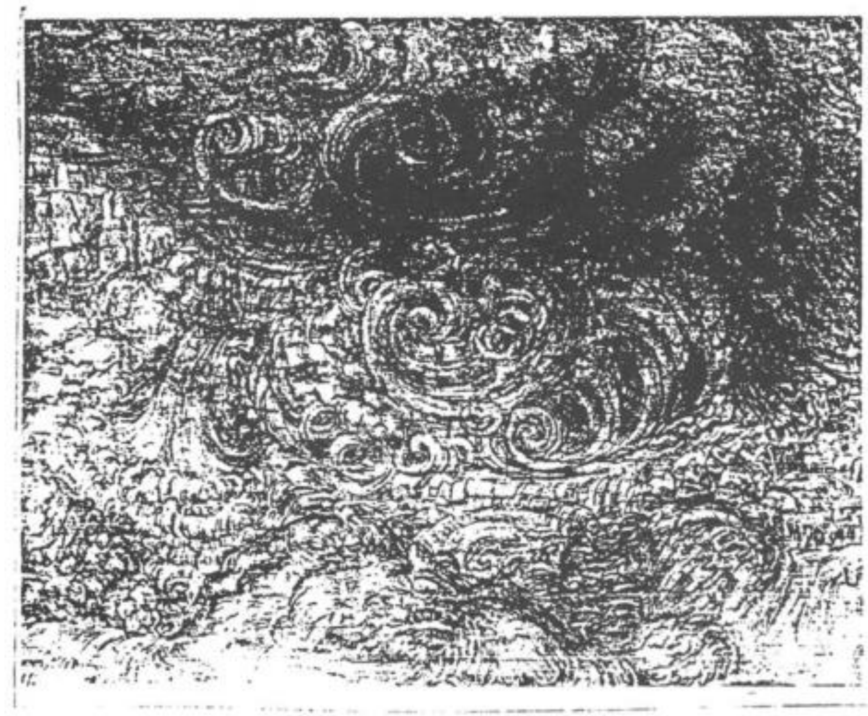
Rendered by Lapineige
Mesh generated by Y. Thorimbert

When fluids need noise





images: [Narain 2008]



The Deluge, by Leonardo Da Vinci.

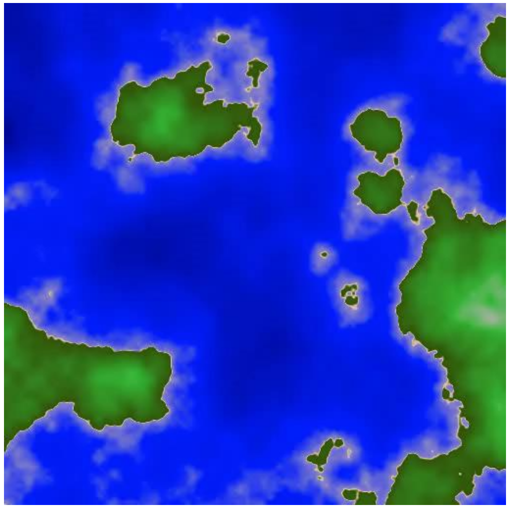
Big Whorls have little whorls,
which feed on their velocity;
And little whorls have lesser
whorls,
And so on to viscosity.
by L.F. Richardson

image: [Kappraff 1986]

Content

1. Procedural generation
   - Procedural content
   - Coherent noise
2. Survey of popular methods
   1. Midpoint displacement
   2. Perlin noise
   3. Simplex noise
3. Study of gradient noise methods

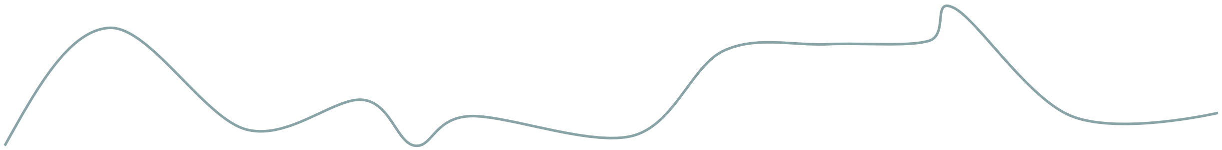
Procedural generation

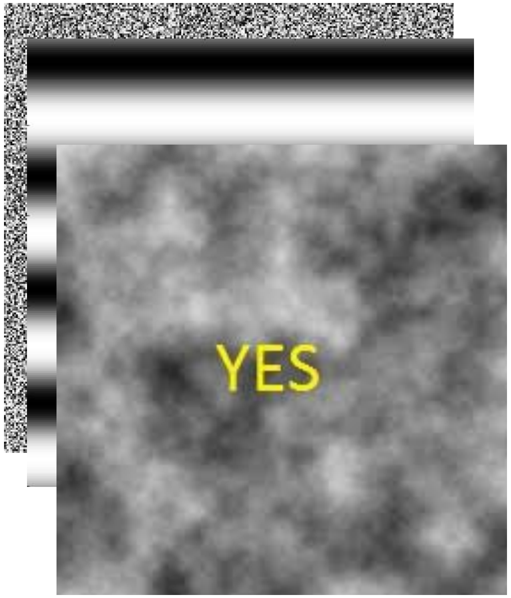


- Random but structured:
  - Use algorithms to generate content
  - Save human work
  - Save computer memory
- Generation of names, labyrinths, donjons, bricks, music, textures, vegetation, landscapes, cities, …
- Industrial applications: Vehicle dynamics[Dawkins 2012], military training[Smelik 2009], …

Procedural generation (2)

Coherent noise

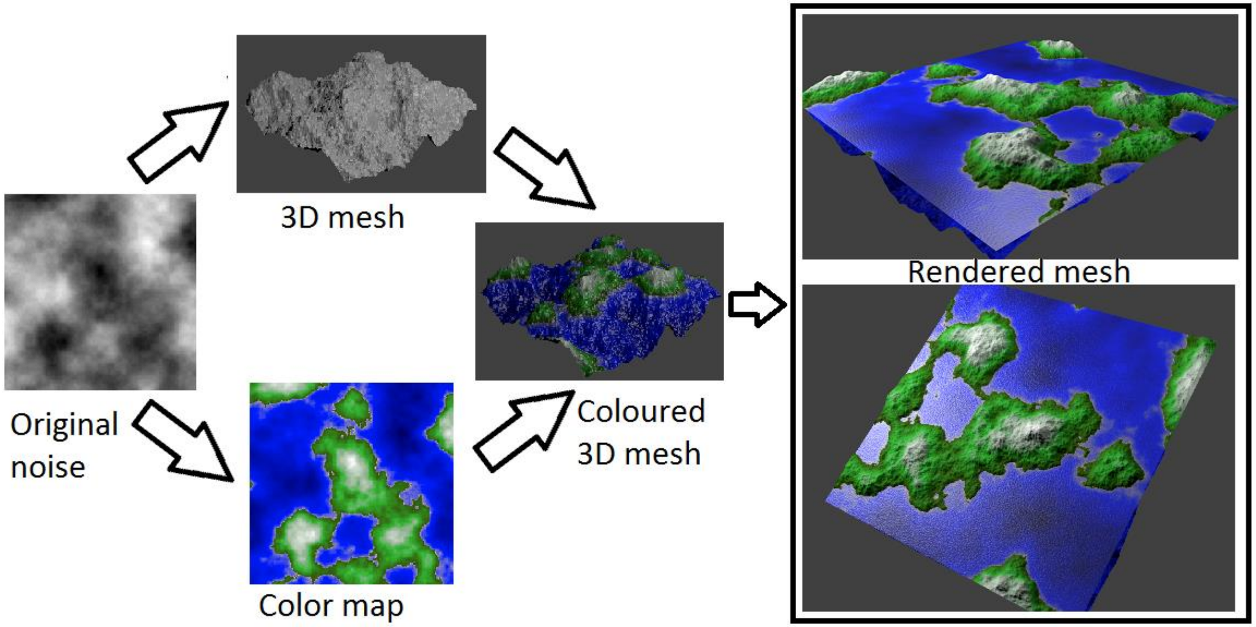




- Replicable
- A small displacement results in a small variation of height
- A large displacement results in a random variation of height

From noise to terrain

- Map value to height
  - Colorscale (2D model) or meshed terrain (3D model)



  - Need 3D height map for generating caves

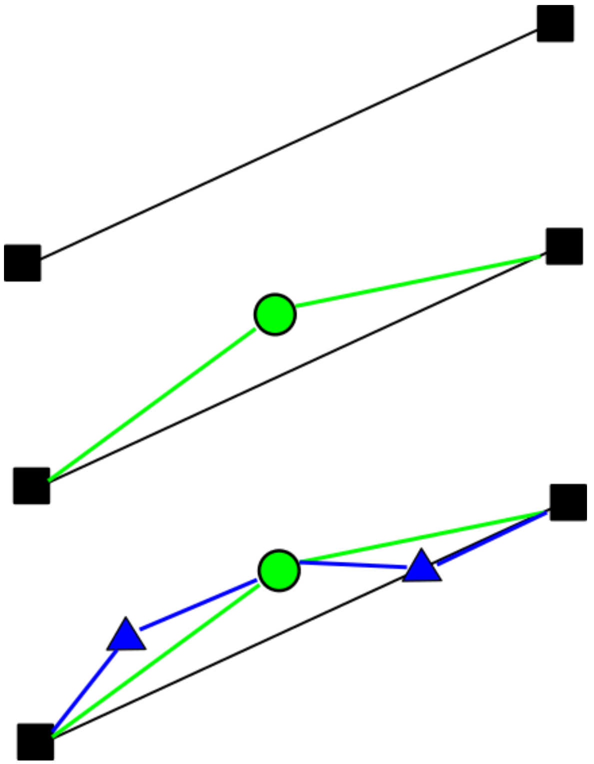
Popular methods

- Midpoint displacement (1986)
- Perlin noise (1985)
- Simplex noise (2001)

Midpoint displacement method

[Miller 1986]

- Every new point is the average of its neighborhood…
- …plus a random offset



Diamond-Square algorithm

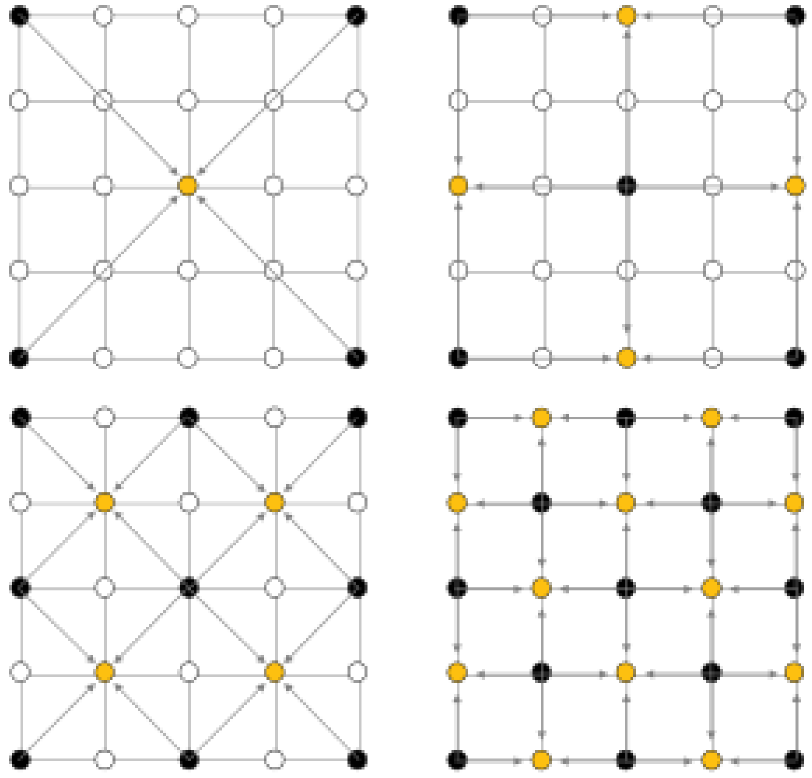
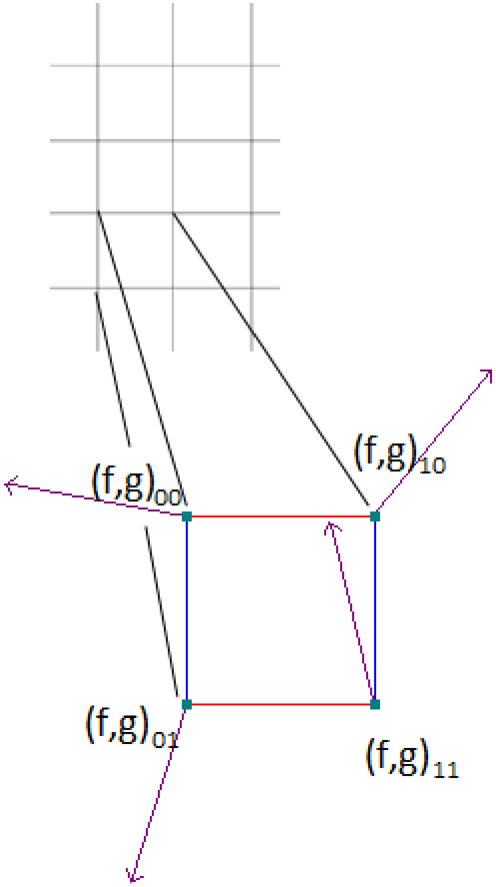


image: Christopher Ewin

Perlin noise [Perlin 1985]

- Regular grid
- Gradient noise
- Multiple octaves
  - Superimposition of multiples independant noises of increasing level of detail and decreasing amplitude



Perlin noise (2)

Y-interpolation:

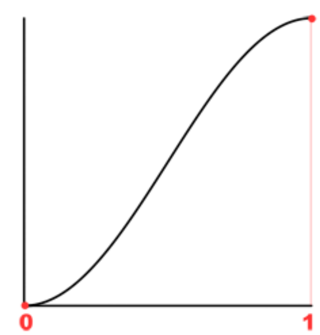
$$h(x,y) = h_0(x,y) + S(y)[h_1(x,y) - h_0(x,y)].$$

X-interpolation at *y=j, j* ∈ *{0,1}*:

$$h_j(x,y) = v_{0j}(x,y) + S(x)[v_{1j}(x,y) - v_{0j}(x,y)],$$

with corner-induced values i, j ∈ *{0,1}* :

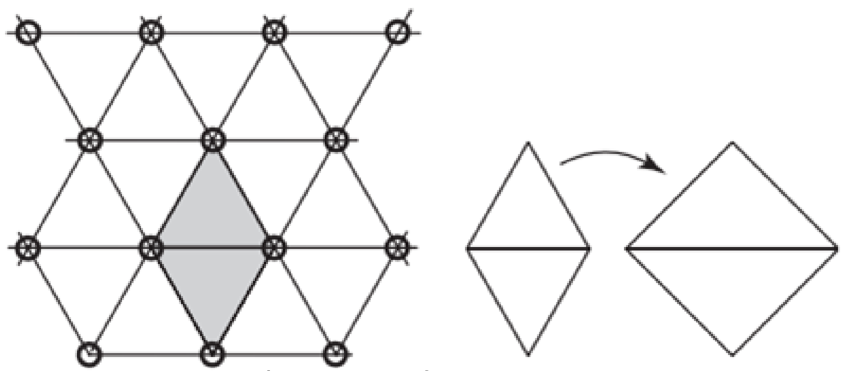
$$v_{ij}(x,y) = f \cdot (x-i) + g \cdot (y-j).$$



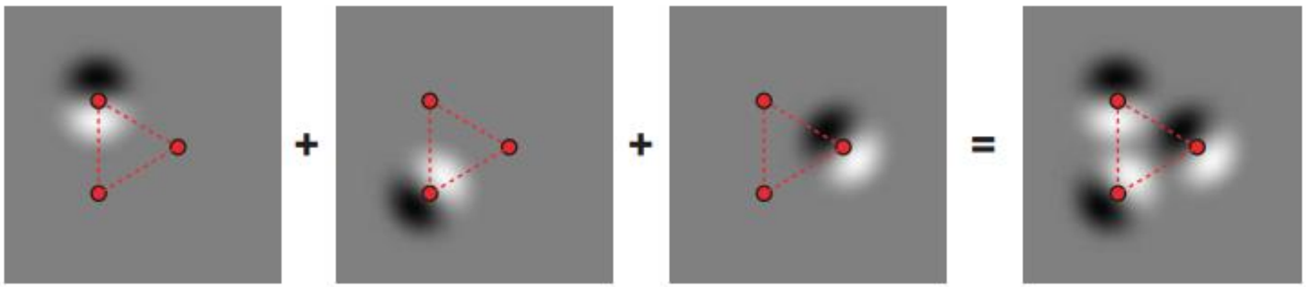
S(t) : smoothstep interpolation function

Simplex noise [Perlin 2001]

- Apply Perlin noise to a simplex grid
- Hypercube in D dimensions has 2^D corners
- Simplex in D dimensions has $D + 1$ corners



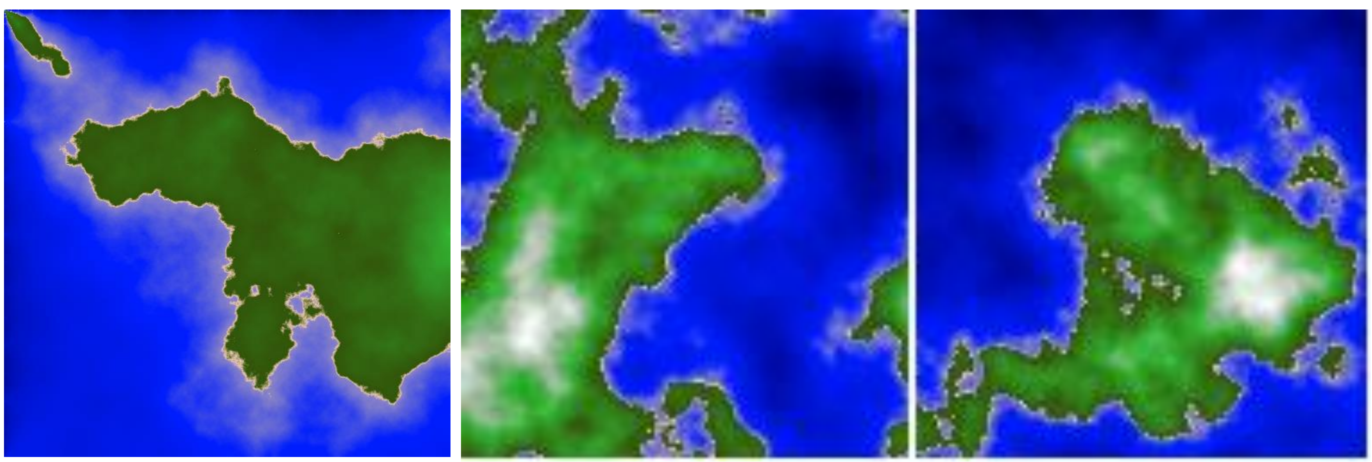
images: Stefan Gustavson



Comparison

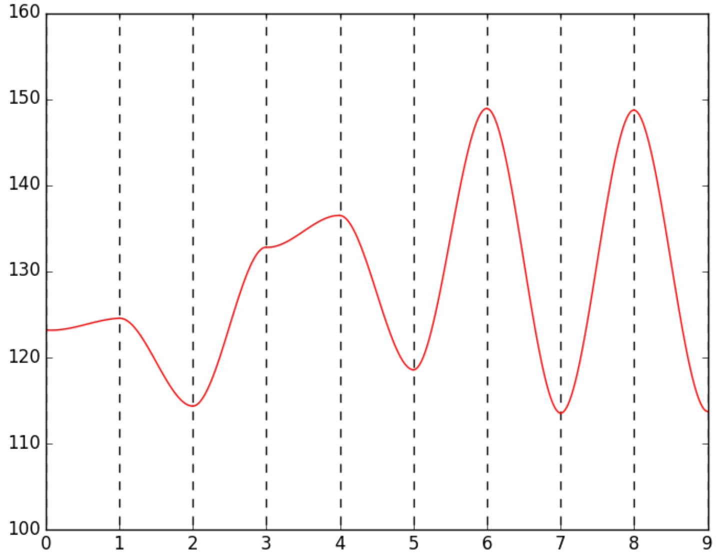
N: number of octaves R: resolution

	Midpoint disp.	Perlin noise	(Open) Simplex
Performance	R^D	$N \cdot R^D + 2^{N \cdot D}$	$N \cdot R^D + D^N$
Pros	Simple	Control parameters	Control parameters
Cons	Non-local	Slow in high dimensions	Complex, originally patented



Another model: polynomials

- Simple idea: constrain value and gradient of a polynomial at boundaries.
- 1D : trivial.



Polynomial method (1)

- One step back: what do we want ? Controlled height and gradient:

$$h^{\mathbf{d}}(\mathbf{X}) = \frac{\partial^{d_1+d_2+..+d_D}}{\partial^{d_1}x_1\ \partial^{d_2}x_2\ ..\ \partial^{d_D}x_D} h(\mathbf{X}), \qquad (1)$$

- Height and gradient along an edge connecting two points must depend only on constraints on these points and no other: $(\boldsymbol{\Delta}_{ij} \equiv \mathbf{s_j} - \mathbf{s_i})$

$$h\left(\mathbf{s_i} + k \cdot \boldsymbol{\Delta}_{ij}\right) = f(\mathbf{s_i}, \mathbf{s_j})\ \forall k \in [0,1] \Leftrightarrow \|\boldsymbol{\Delta}_{ij}\| = 1 \qquad (2)$$

Polynomial method (2)

- Arbitrary choice: $h(\mathbf{X})$ is a polynomial of degree n:

$$h(\mathbf{X}) = \sum_{\mathbf{a} \in I} \left(c_{\mathbf{a}} \prod_{k=1}^{D} x_k^{a_k} \right) , \tag{3}$$

where I is the set of all vectors on the form $(a_1, a_2, \dots, a_D)$ such that $0 \leq a_k \leq n \ \forall k$.

Polynomial method (3)

- Dimension D, constraint order m and degree n are related:

$$2^D \cdot \sum_{i=0}^{m} \frac{(D+i-1)!}{i!(D-1)!} \leq (n+1)^D \,. \tag{4}$$

- Simplest case for $D = 2$: $m = 1$ and $n = 3$.

$$h(x,y) = \sum_{i,j} c_{ij} x^i y^j \,, \tag{5}$$

Polynomial method (4)

- 4 constraints h_{ij} on height eq.(5), 4 constraints f_{ij} on x-derivative, 4 constraints g_{ij} on y-derivative, and edge condition eq.(2):

$$c_{00} = h_{00},$$
$$c_{11} = h_{01} + h_{10} - h_{00} - h_{11} + f_{01} + g_{10} - g_{00} - f_{00},$$
$$c_{22} = c_{33} = c_{32} = c_{23} = 0,$$

$$\begin{aligned} c_{10} &= f_{00}, \\ c_{20} &= 3(h_{10} - h_{00}) - 2f_{00} - f_{10}, \\ c_{30} &= f_{10} + f_{00} - 2(h_{10} - h_{00}), \\ c_{21} &= 3(h_{11} - h_{01}) - 2f_{01} - f_{11} - c_{20}, \\ c_{31} &= f_{11} + f_{01} - 2(h_{11} - h_{01}) - c_{30}, \end{aligned}$$

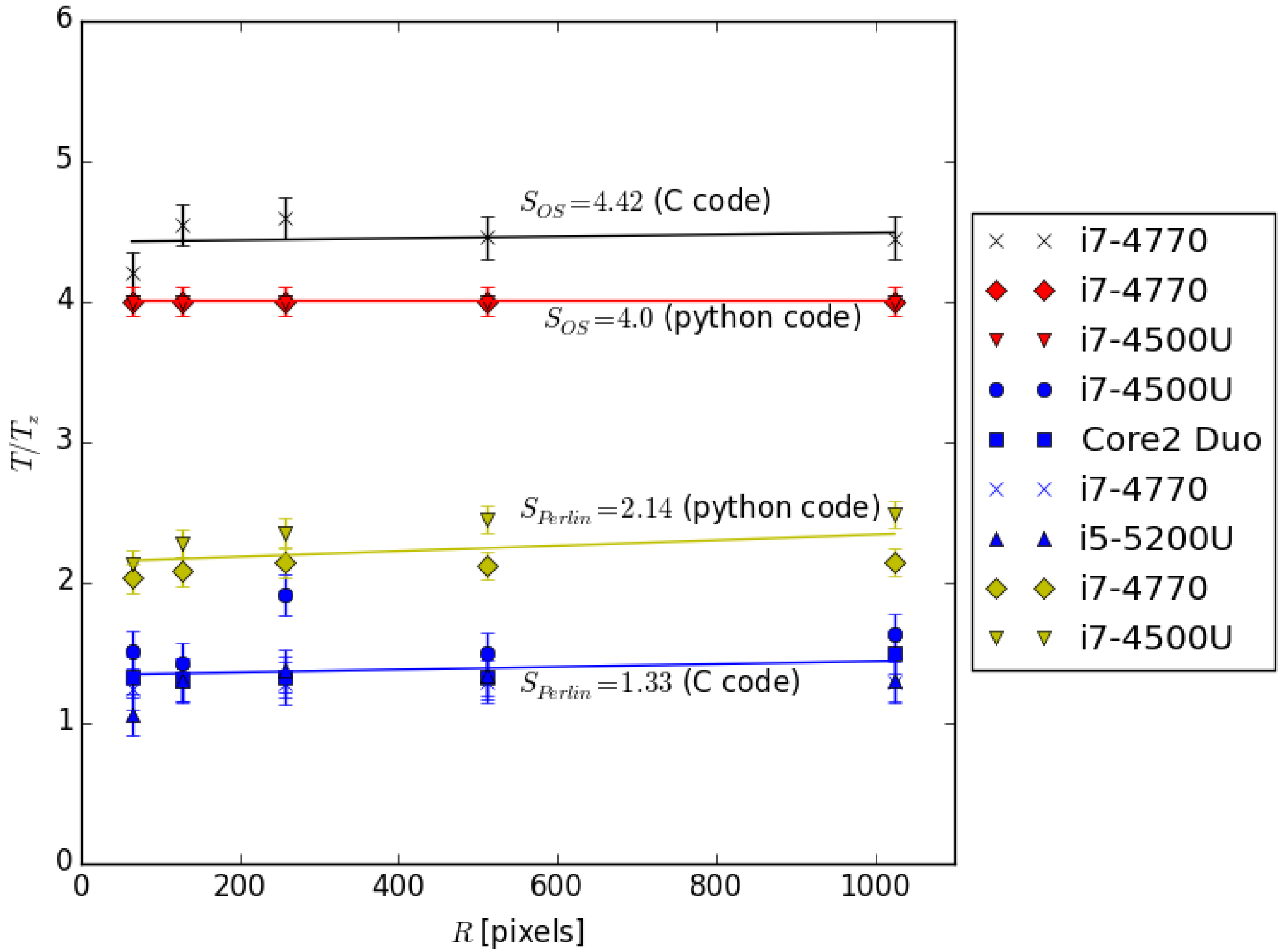
$$\begin{aligned} c_{01} &= g_{00}, \\ c_{02} &= 3(h_{01} - h_{00}) - 2g_{00} - g_{01}, \\ c_{03} &= g_{01} + g_{00} - 2(h_{01} - h_{00}), \\ c_{12} &= 3(h_{11} - h_{10}) - 2g_{10} - g_{11} - c_{02}, \\ c_{31} &= g_{11} + g_{10} - 2(h_{11} - h_{10}) - c_{03}\,. \end{aligned}$$

Zero-Gradient polynomial

- Our generic model is slower than Perlin.
- Perlin's value on corner is always zero.
- Similarly, we build a model with zero gradient at corners: $f_{ij} = g_{ij} = 0$.
- The resulting model reduces the number of both *mult* and *add* compared to Perlin noise:

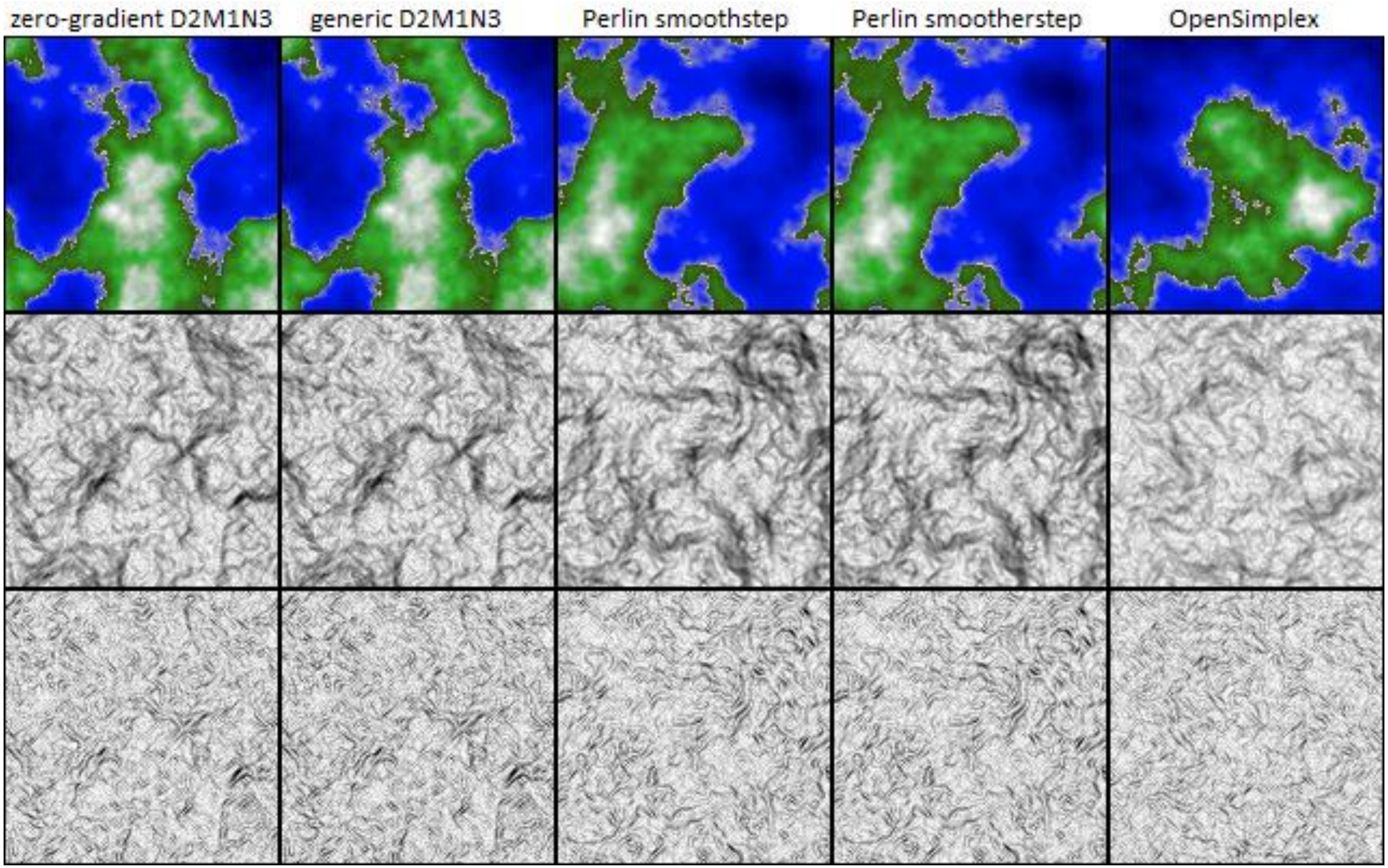
$$h(x,y) = h_{00} + S_3(x)\Delta x + S_3(y)\Delta y + \\ A\left[S_3(x)\cdot y + S_3(y)\cdot x + xy\right] \quad (6)$$

Performance comparison



Comparison

height map and gradients



Fractals

- Most real landscapes (and many natural structures in general) exhibit fractal features.
- Fuzzy definition: mathematical set that is invariant under rescaling.
- Tautological definition: a fractal object is an object whose each part is also a fracal object.

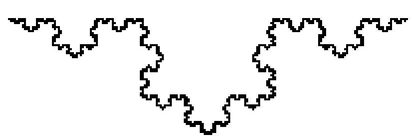
Image: Bert Hickman

Fractal dimension

- Richardson effect [Mandelbrot 1977]
- $L(\epsilon) = k \cdot \epsilon^{1-D}$
- Counting box method with N octaves :

$$D = \lim_{(N,\epsilon)\to(\infty,0)} \frac{-\log L(N)}{\log \epsilon} + 1. \qquad (7)$$

- Koch snowflake: $D \approx 1.26$



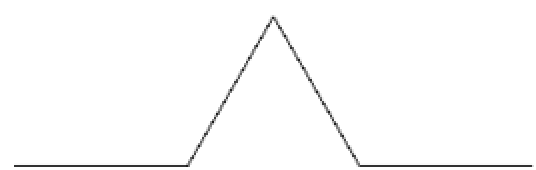
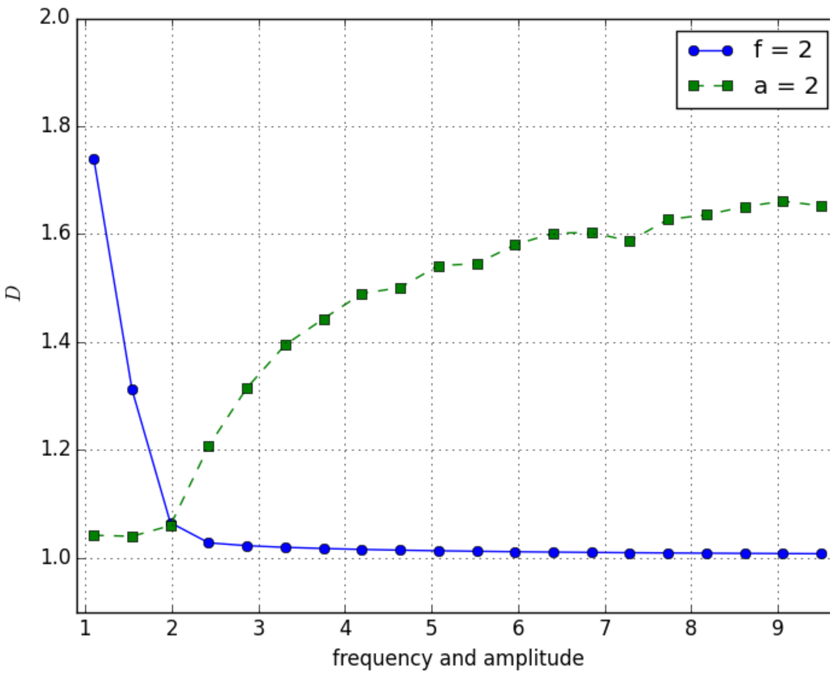
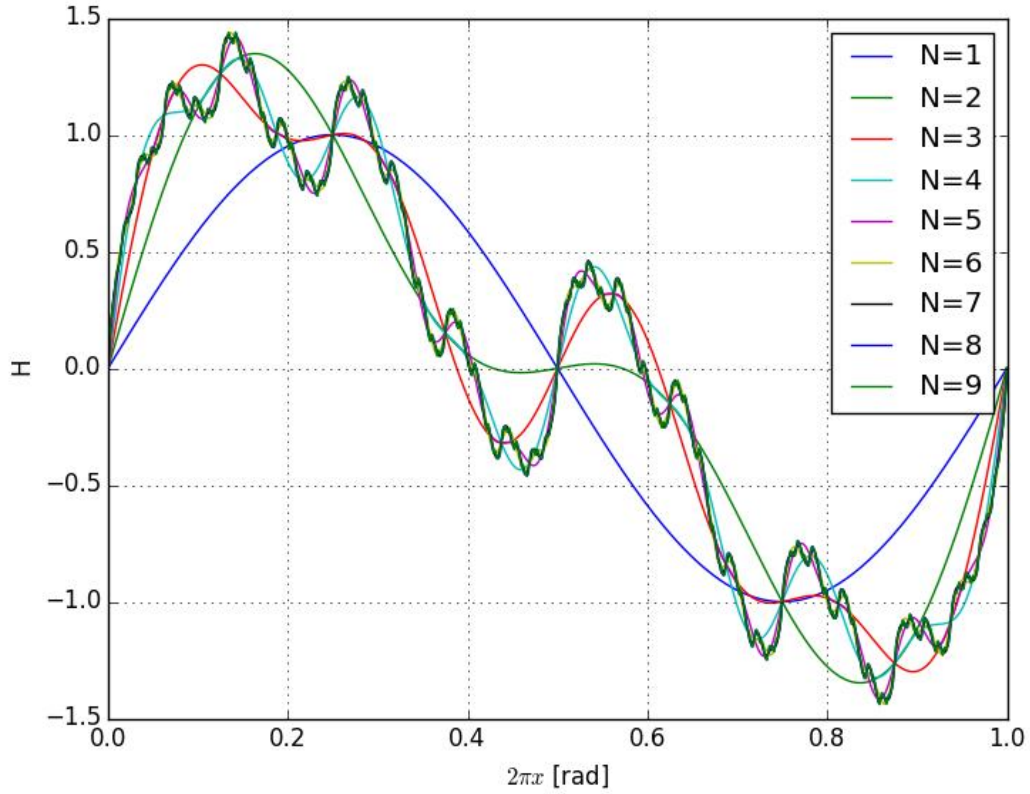


Image: Christophe Dang Ngoc Chan

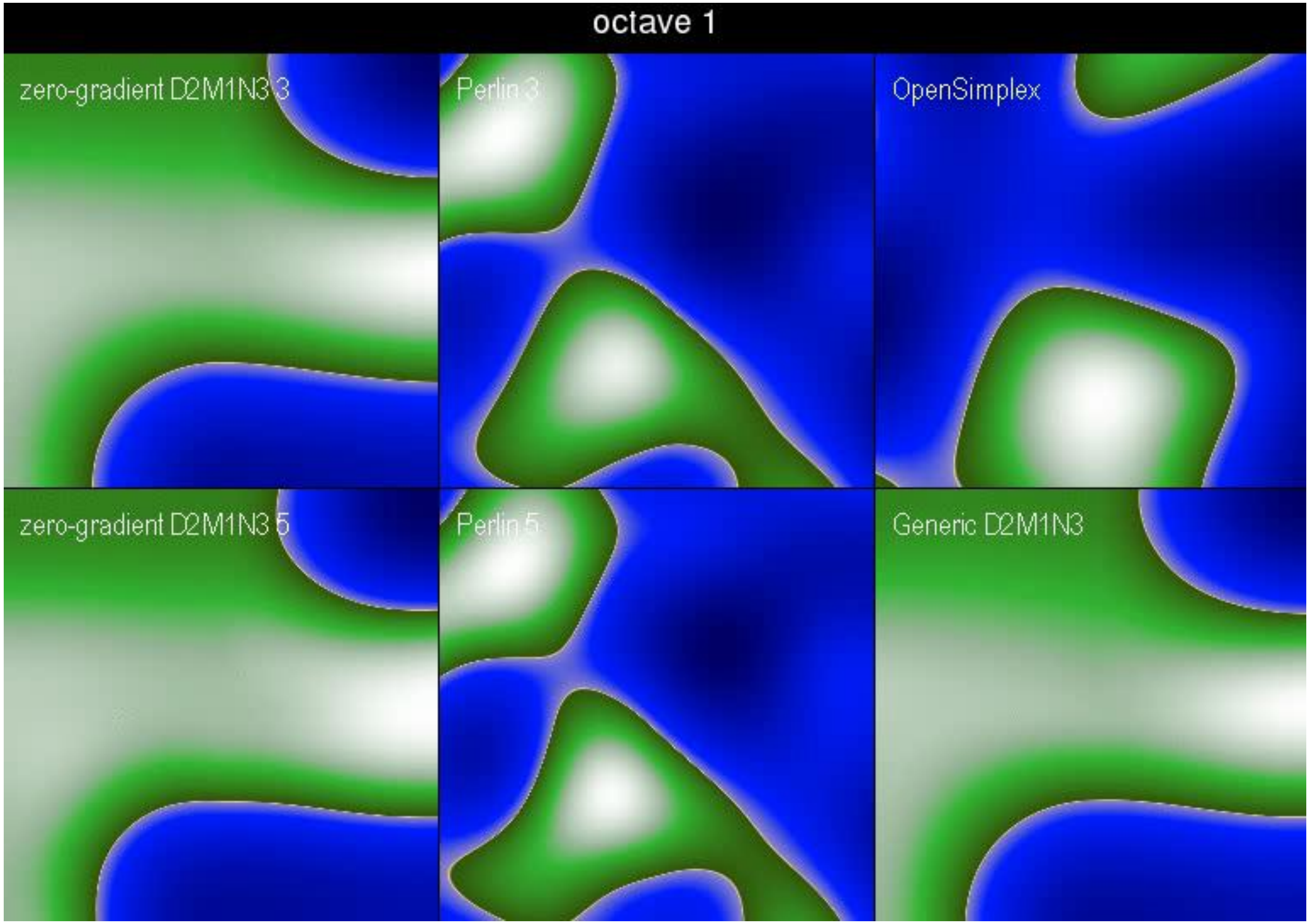
Fractality emerging from octaves superimposition

- Simple sin case: $H(x) = \sum_{i=0}^{N} a^{-i} \sin(f^{i} x)$

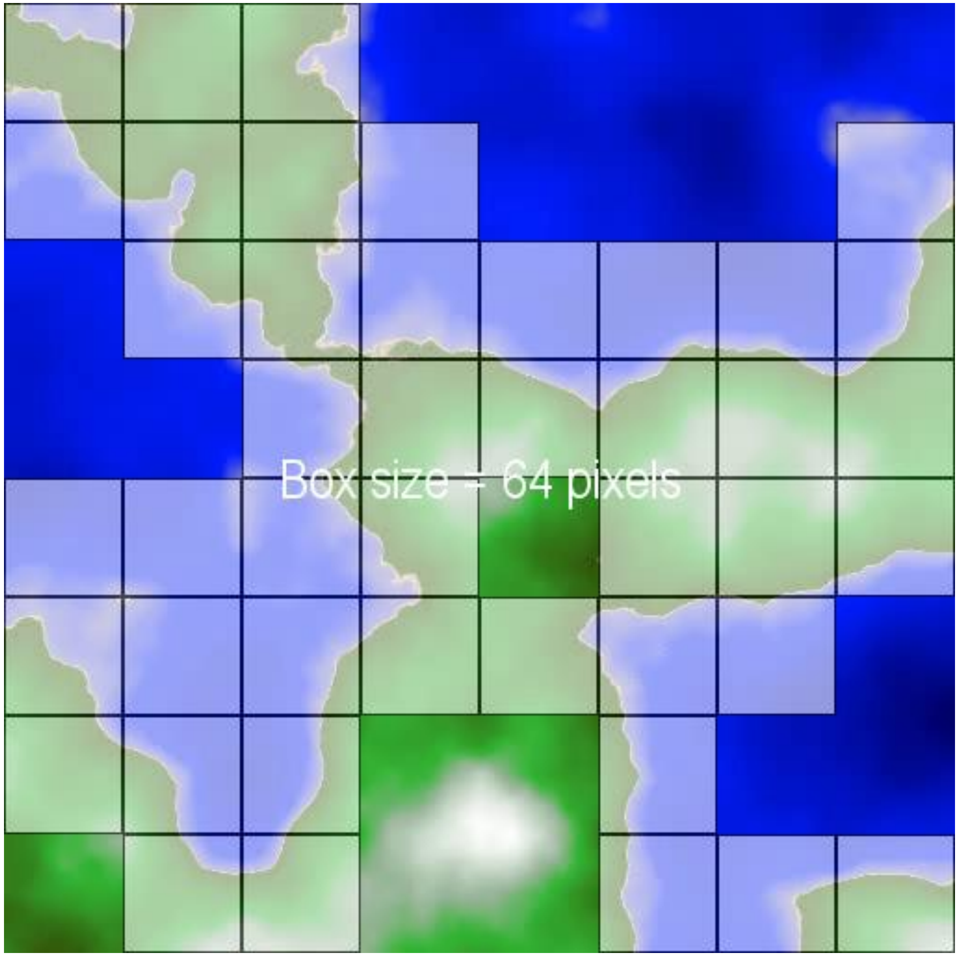




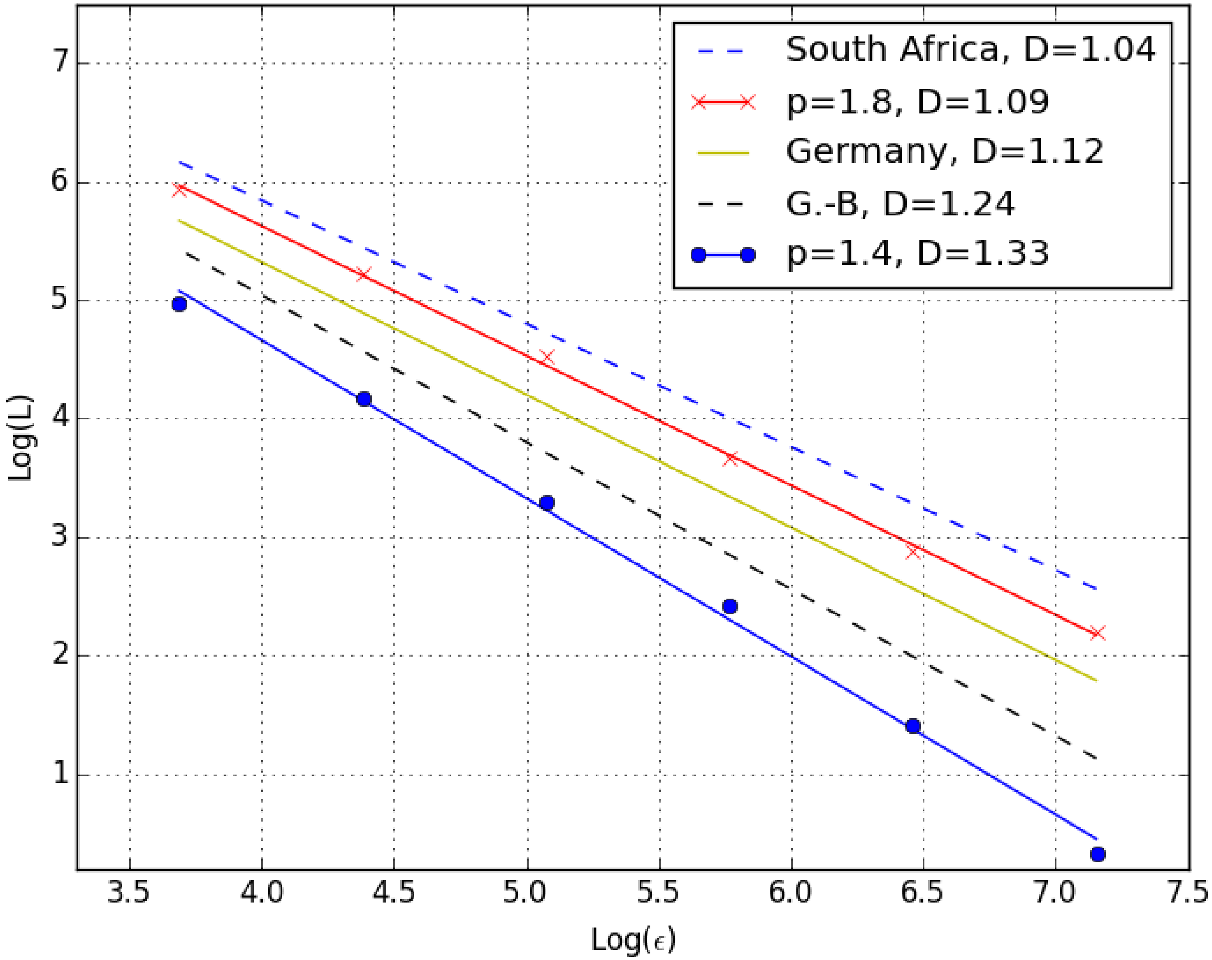
Octaves superimposition



Fractal dimension of generated terrain



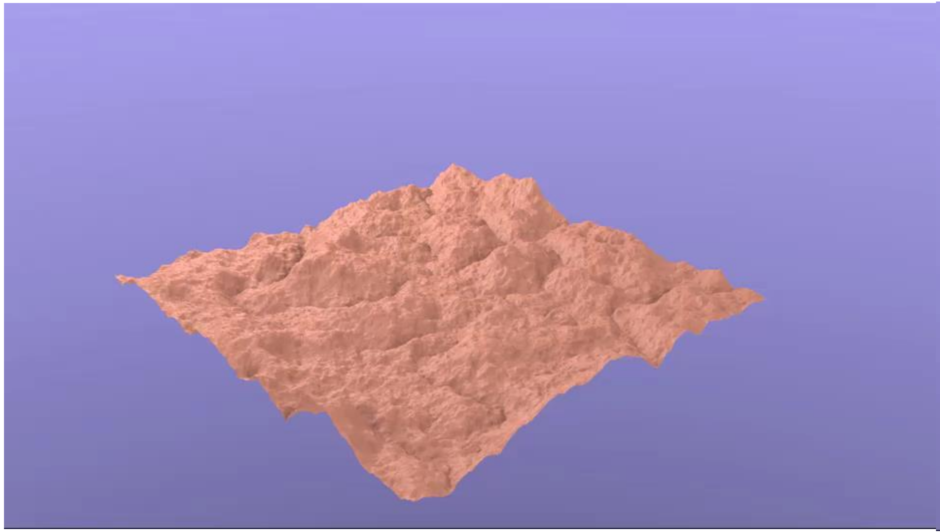
Fractal dimension of generated terrain (2)



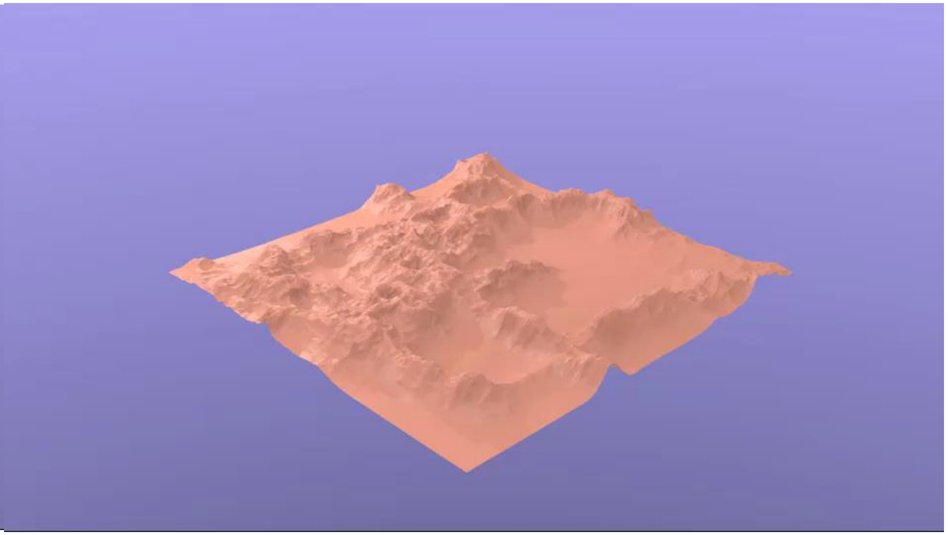
Modified noise

- Abs noise : $|h|$
- Ridged noise

Abs noise

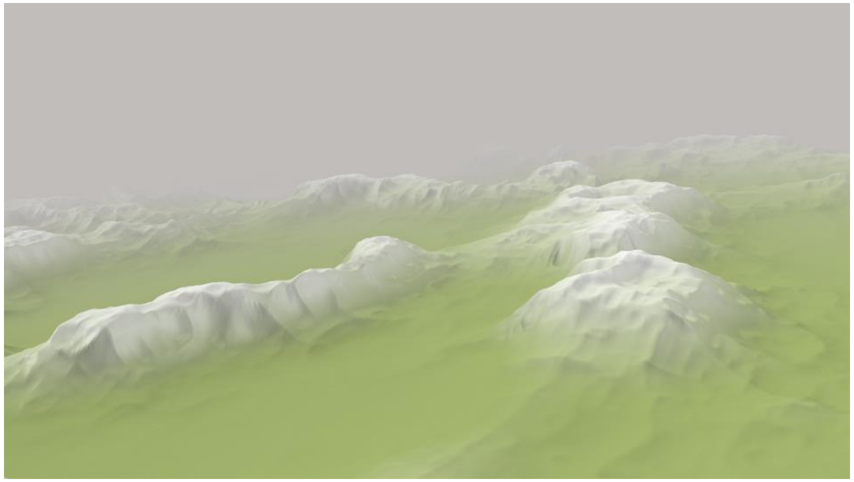


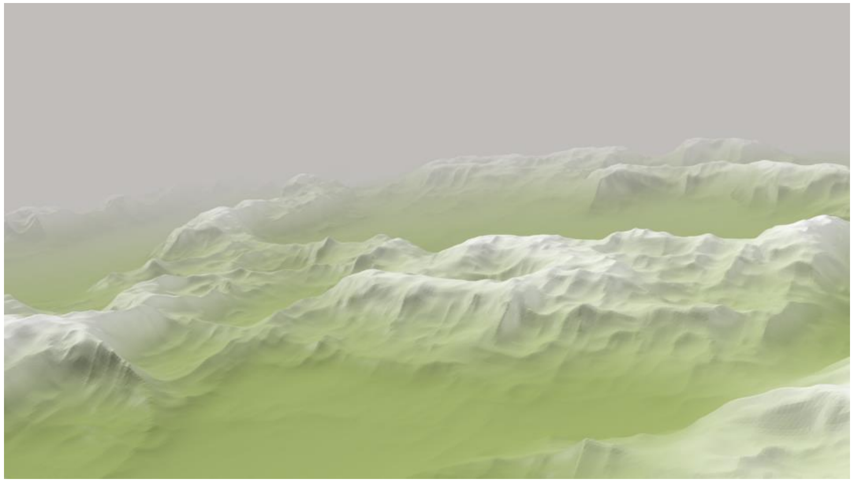
Ridged noise



Ridged noise

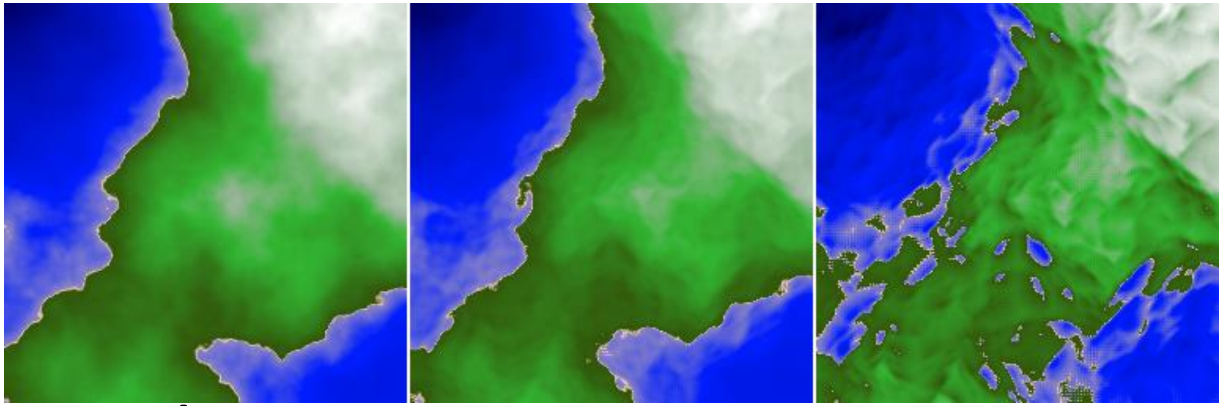
- Ridged noise [Libnoise]:
  - Modification per octave with depth dependance
  - $h_i \rightarrow k \cdot (1 - |h_i|)^2$
  - $k \rightarrow 2h_i$



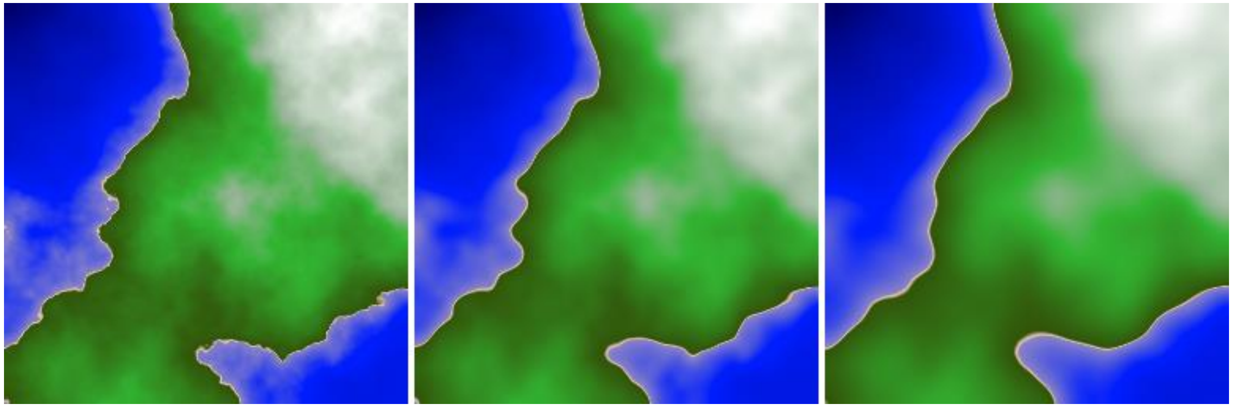


Erosion [Archer 2011]

- Hydrolic erosion

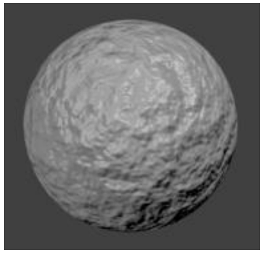
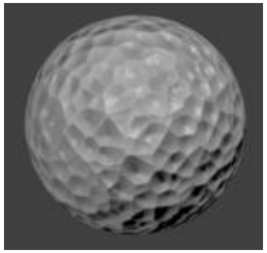
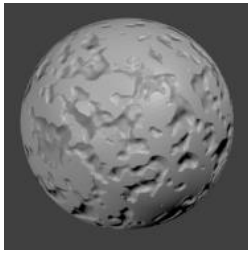


- Thermal erosion

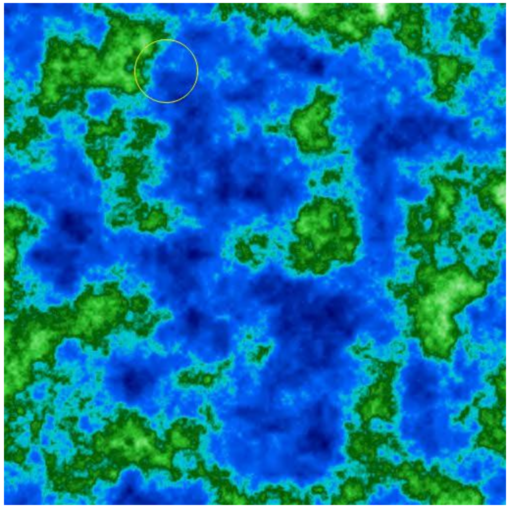


Other uses

- Pixel shaders

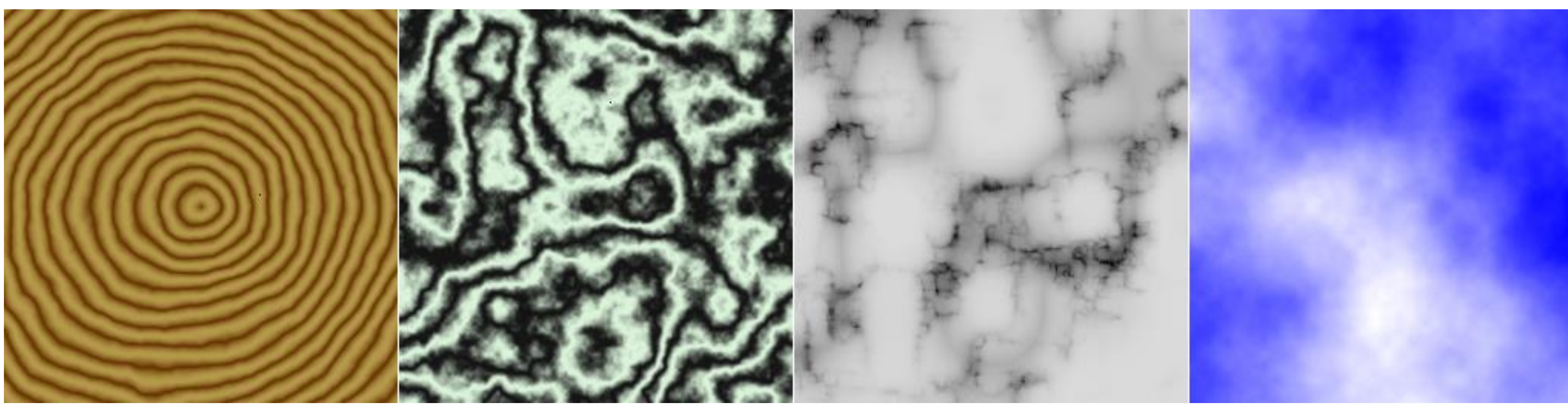




- Dynamic modeling

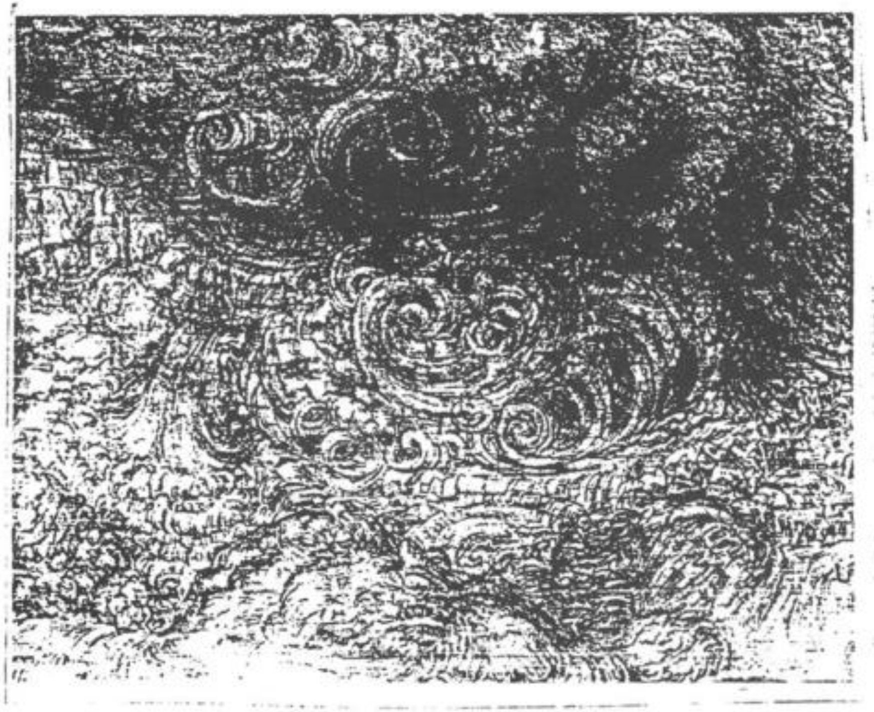



From noise to texture

- Use modified noise and colormap:
  - Radial sin function → wood
  - sin function → marble
  - Ridged noise → marble
  - Raw noise → clouds



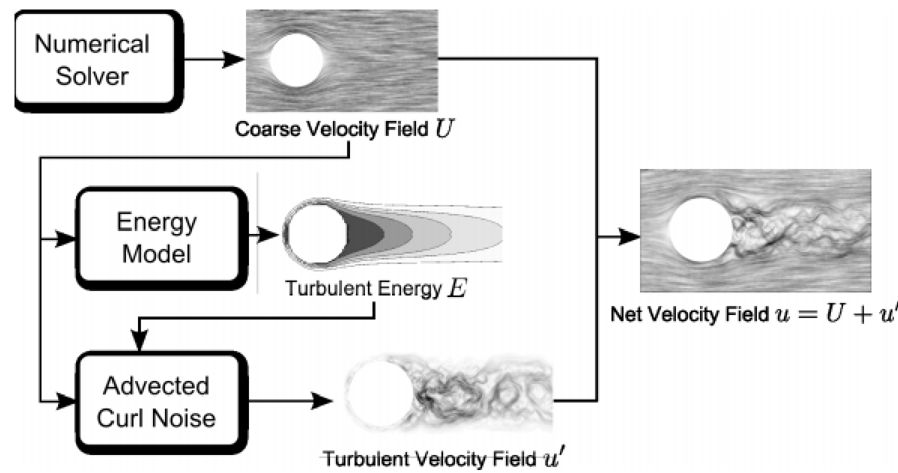
When fluids need noise

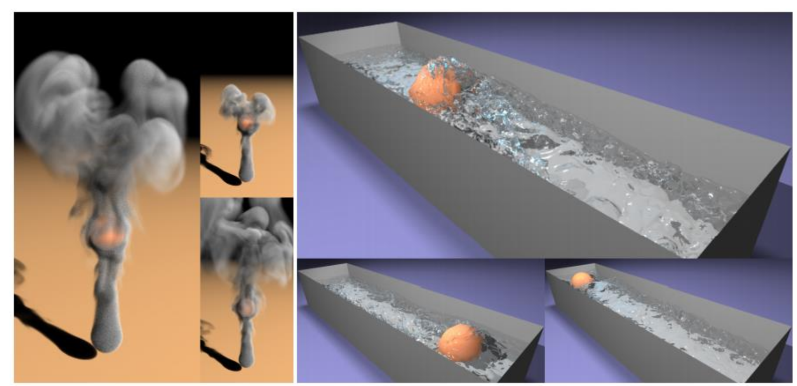


The Deluge, by Leonardo Da Vinci.

Big Whorls have little whorls,
which feed on their velocity;
And little whorls have lesser
whorls,
And so on to viscosity.
by L.F. Richardson

image: [Kappraff 1986]





images: [Narain 2008]

Conclusion

- Geomorphology: imitation or understanding?
- Large use, minor renown
- Coupling with other noises (Voronoi, Worley [Worley 96, …])
- More controlability ➔ Feature-based models [Zhou 2007, …]
- More realism ➔ Physically-based models [Génevaux 2013, …]

Image : [Génevaux 2013]

Image : [Zhou 2007]

References

- Code, presentation and more references on http://cui.unige.ch/~thorimby/procgen/
- All images produced with zero-gradient model (unless otherwise specified); 3D renderings with Blender

[Archer 2011] http://micsymposium.org/mics_2011_proceedings/mics2011_submission_30.pdf (accessed 11/11/2016)

[Dawkins 2012] Jeremy J Dawkins, David M Bevly, and Robert L Jackson. Evaluation of fractal terrain model for vehicle dynamic simulations. *JOURNAL OF TERRAMECHANICS*, 49(6), 2012.

[Génevaux 2013] Jean-David Genevaux, Eric Galin, Eric Guerin, Adrien Peytavie, and Bedrich Benes. Terrain generation using procedural models based on hydrology. *ACM Trans. Graph* 32, July 2013.

[Kappraff 1986] Jay Kappraff. The geometry of coastlines: a study in fractals. *Computers & Mathematics with Applications*, 12(3), 1986.

[Libnoise] http://libnoise.sourceforge.net/ (accessed 11/11/2016)

[Mandelbrot 1977] Benoit B. Mandelbrot. The Fractal Geometry of Nature. *W. H. Freeman and Company,* 1977.

[Miller 1986] Gavin S P Miller. The definition and rendering of terrain maps. *SIG-GRAPH Comput. Graph.*, 1986.

References (2)

[Narain 2008] Rahul Narain, Jason Sewall, Mark Carlson, and Ming C. Lin. Fast animation of turbulence using energy transport and procedural synthesis. In *ACM SIGGRAPH Asia 2008 Papers*, NY, USA, 2008. ACM.

[Perlin 1985] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph*. July 1985.

[Perlin 2001] Ken Perlin. Noise Hardware. *SIGGRAPH Course Notes*, 2001.

[Smelik 2009] Ruben M. Smelik, Tim Tutenel, Klaas Jan de Kraker, and Rafael Bidarra. A case study on procedural modeling of geo-typical southern Afghanistan terrain. In *Proceedings of IMAGE 2009*, St.Louis, Missouri, jul 2009.

[Worley 96] Steven Worley. A cellular texture basis function. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, *SIGGRAPH '96*, ACM.

[Zhou 2007] Howard Zhou, Jie Sun, Greg Turk, and James M. Rehg. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, July/August 2007