HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

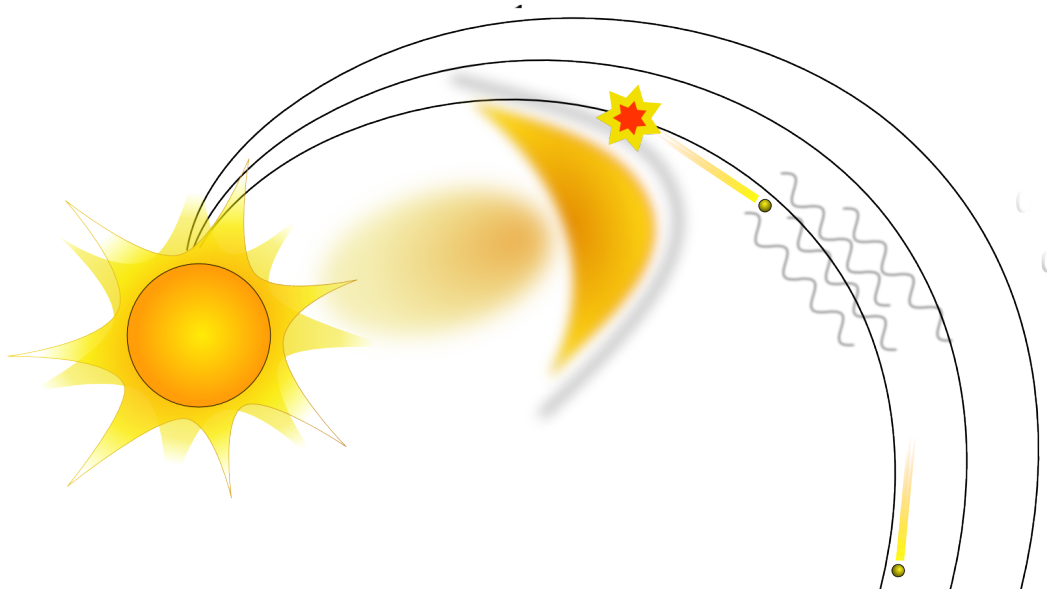
Perlin Noise as a Turbulence Model for Particle Transport

Urs Ganse[1], Rami Vainio[1], Alex Ivascenko[2], Felix Spanier[2]

[1] University of Helsinki

[2] North-West University Potchefstroom

Particle Transport and Turbulence



- Particle transport in heliospheric and interstellar medium
- Test particle simulations
- Turbulent fields play important role.

Simulated Turbulence

- Getting turbulence right is difficult.
- Self-consistent turbulence simulations quite limited in scales (kinetic $\Leftrightarrow$ heliospheric)

Simulated Turbulence

- Getting turbulence right is difficult.
- Self-consistent turbulence simulations quite limited in scales (kinetic $\Leftrightarrow$ heliospheric)
- Often, effect of turbulence approximated through diffusion.
- Alternatively, ad-hoc model of scattering is used.

Simulated Turbulence

- Getting turbulence right is difficult.
- Self-consistent turbulence simulations quite limited in scales (kinetic $\Leftrightarrow$ heliospheric)
- Often, effect of turbulence approximated through diffusion.
- Alternatively, ad-hoc model of scattering is used.
- We'd rather have a way to construct quasi-turbulent fields.

Quasi-Turbulent Fields

- Simply taking random numbers gives wrong spectrum.

Quasi-Turbulent Fields

- Simply taking random numbers gives wrong spectrum.
- Creating correct spectrum in Fourier space works.
- Again bound to a grid.
- Time evolution?

Quasi-Turbulent Fields

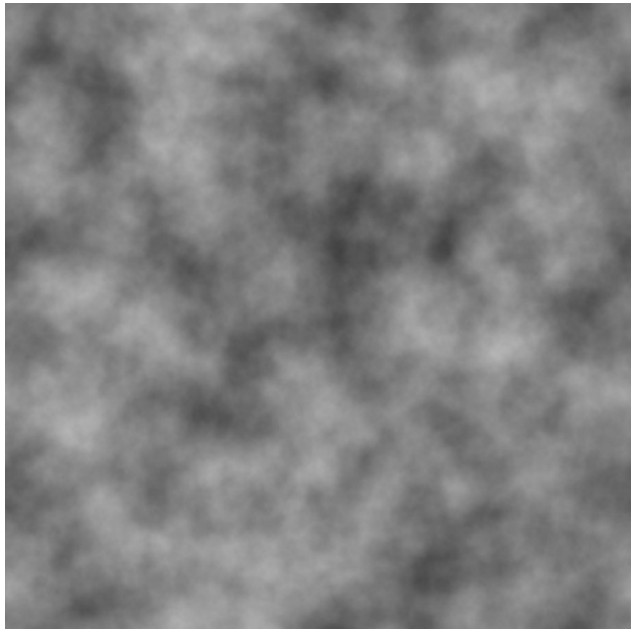
- Simply taking random numbers gives wrong spectrum.
- Creating correct spectrum in Fourier space works.
- Again bound to a grid.
- Time evolution?
- We would like to have deterministic randomness with the correct spectrum without having to look at all space.

Perlin Noise

- Idea from computer graphics

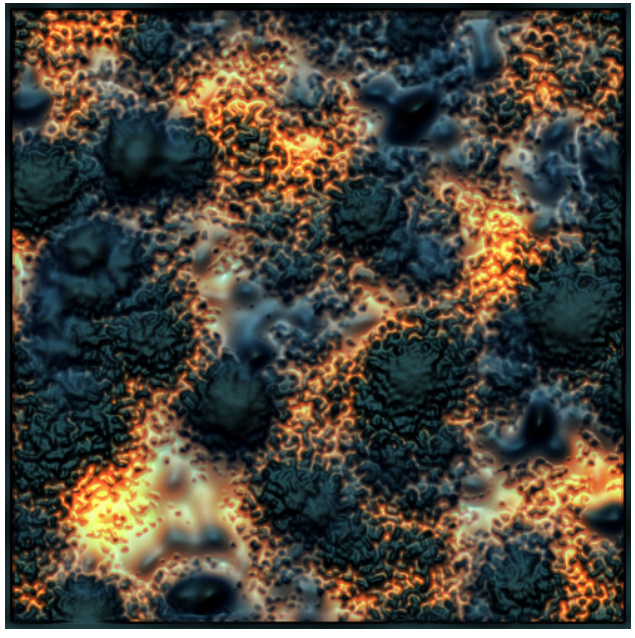
Perlin Noise

- Idea from computer graphics
- Developed by Ken Perlin for the movie "Tron" (Disney, 1982)
- Originally a cloud-texture generator.
- Structures of fixed scale.



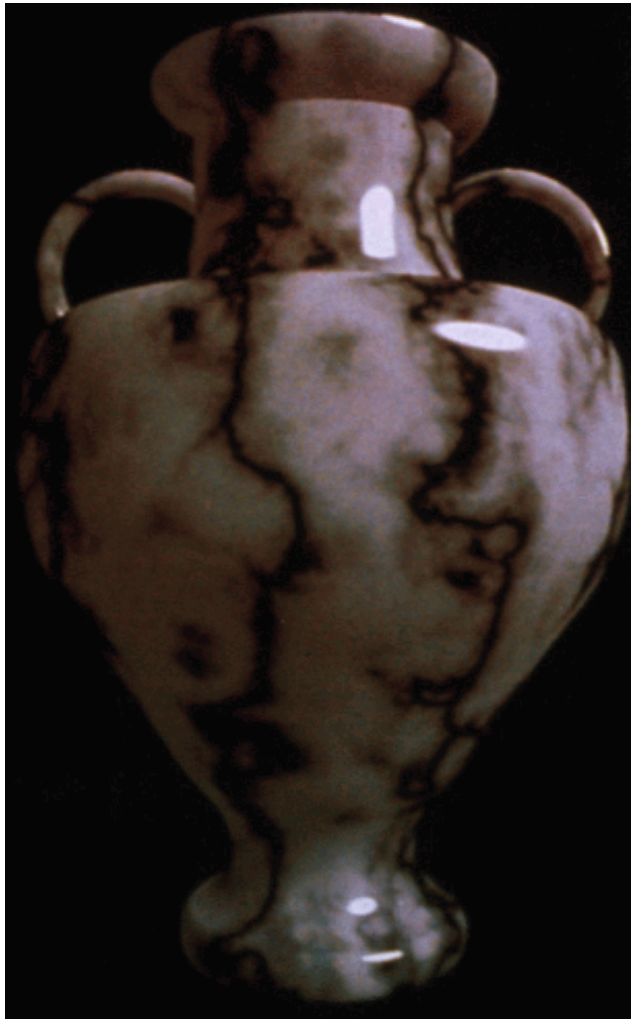
Perlin Noise

- Idea from computer graphics
- Developed by Ken Perlin for the movie "Tron" (Disney, 1982)
- Originally a cloud-texture generator.
- Structures of fixed scale.



Perlin Noise

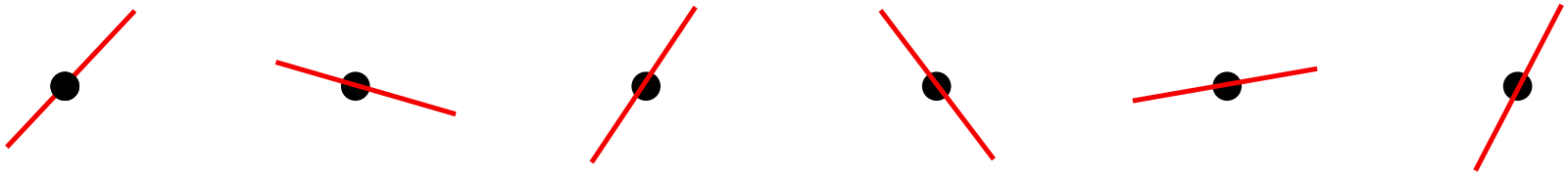
- Idea from computer graphics
- Developed by Ken Perlin for the movie "Tron" (Disney, 1982)
- Originally a cloud-texture generator.
- Structures of fixed scale.



Perlin Noise - internals

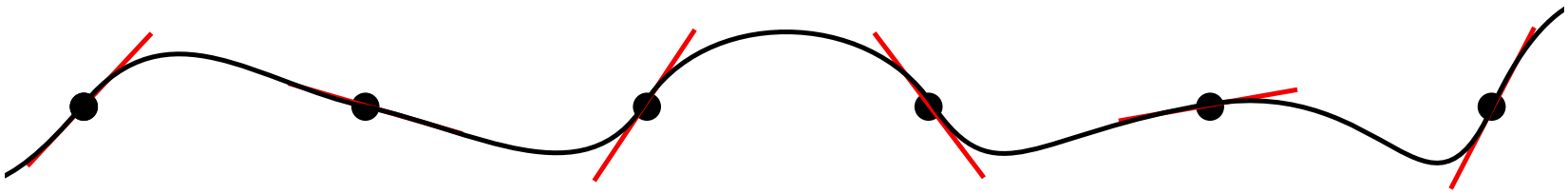
Pick deterministic gradients on grid, interpolate using Hermite polynomial.

Perlin Noise - internals



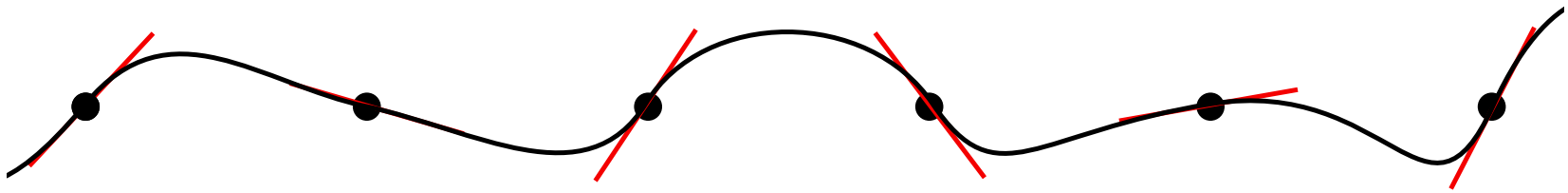
Pick deterministic gradients on grid, interpolate using Hermite polynomial.

Perlin Noise - internals



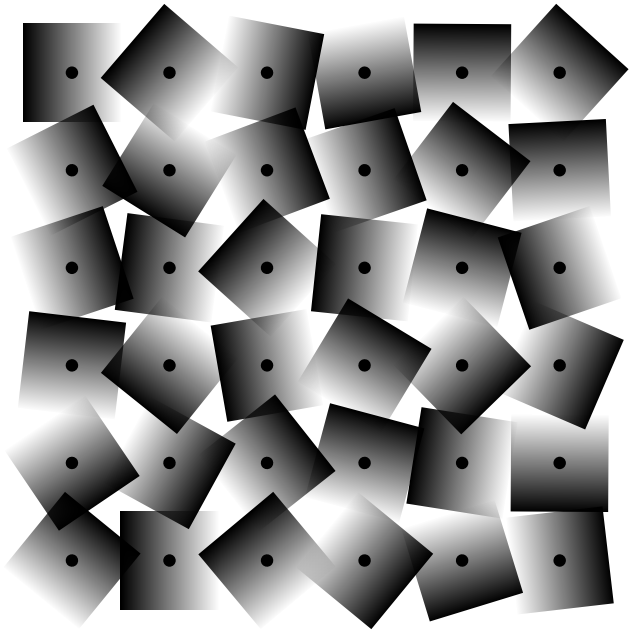
Pick deterministic gradients on grid, interpolate using Hermite polynomial.

Perlin Noise - internals

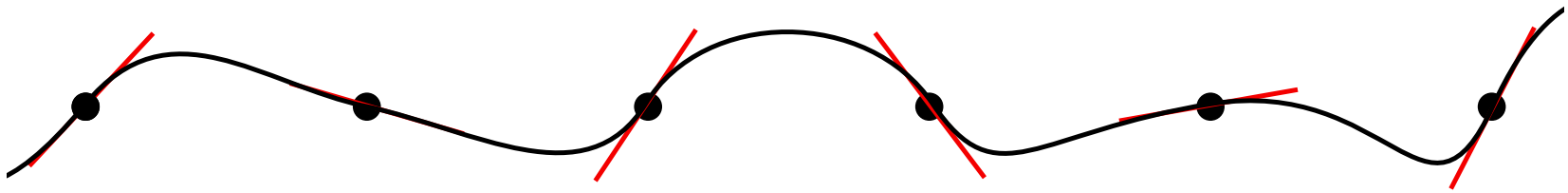


Pick deterministic gradients on grid, interpolate using Hermite polynomial.

Original Perlin Noise (1982):

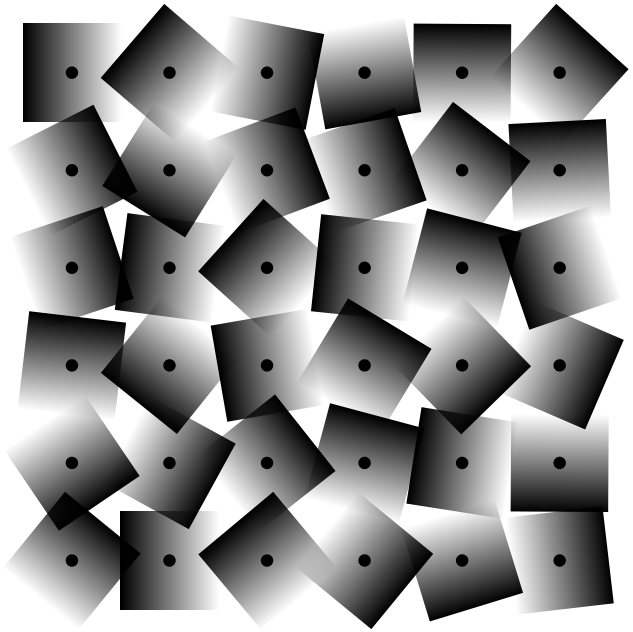


Perlin Noise - internals

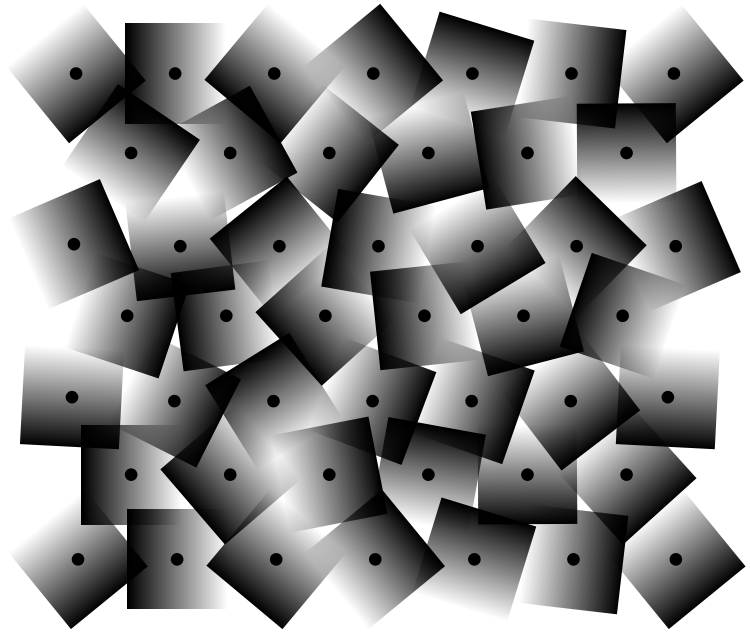


Pick deterministic gradients on grid, interpolate using Hermite polynomial.

Original Perlin Noise (1982):



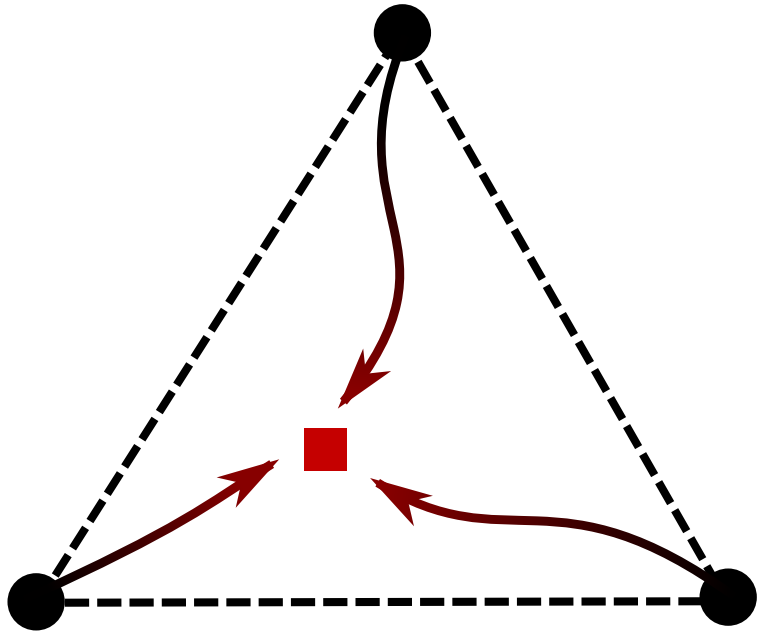
Simplex Noise (2001):



Implementation Details

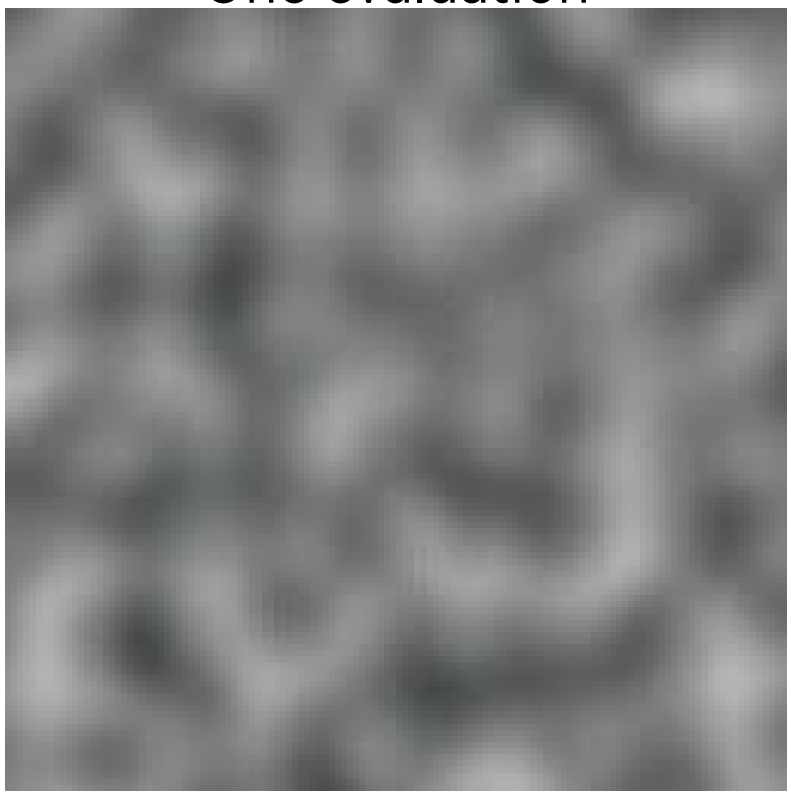
Advantage:

- Interpolation is completely local within one grid cell.
- Grid values are not stored, but calculated on-demand by a (cheap) hash function.
- Completely independent evaluation, trivially parallel.
- Not just suitable for GPUs - GPUs were designed for this.
- Spatial size is not inherently limited, except by floating point accuracy.



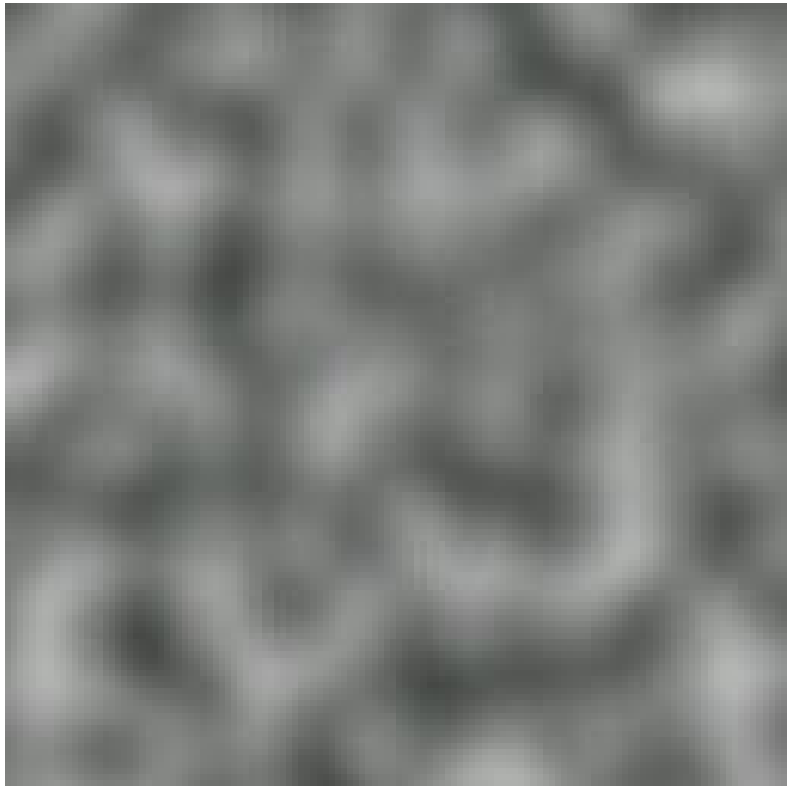
Noise Octaves

One evaluation

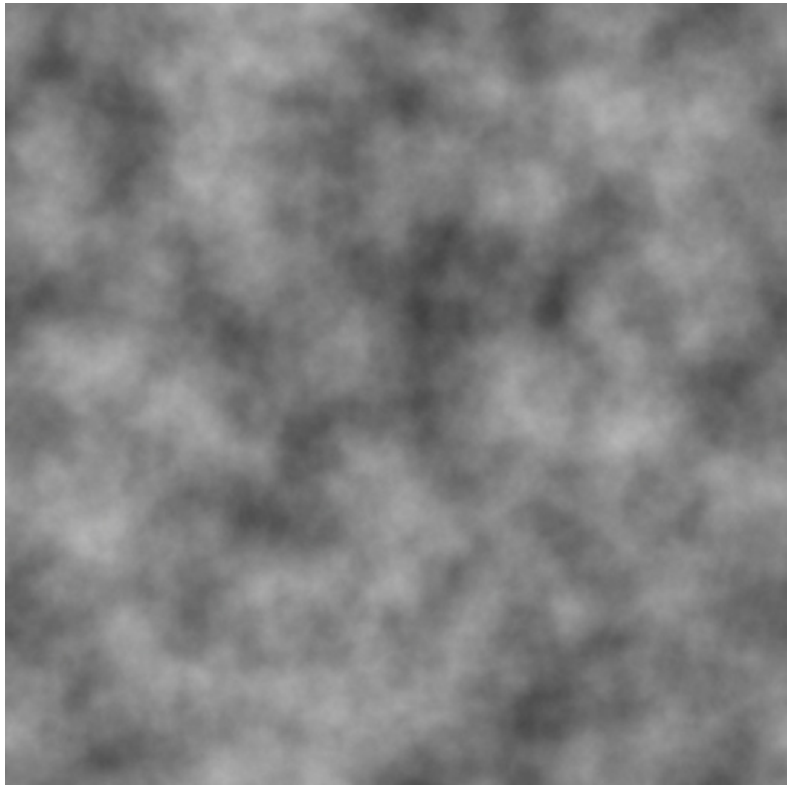


Noise Octaves

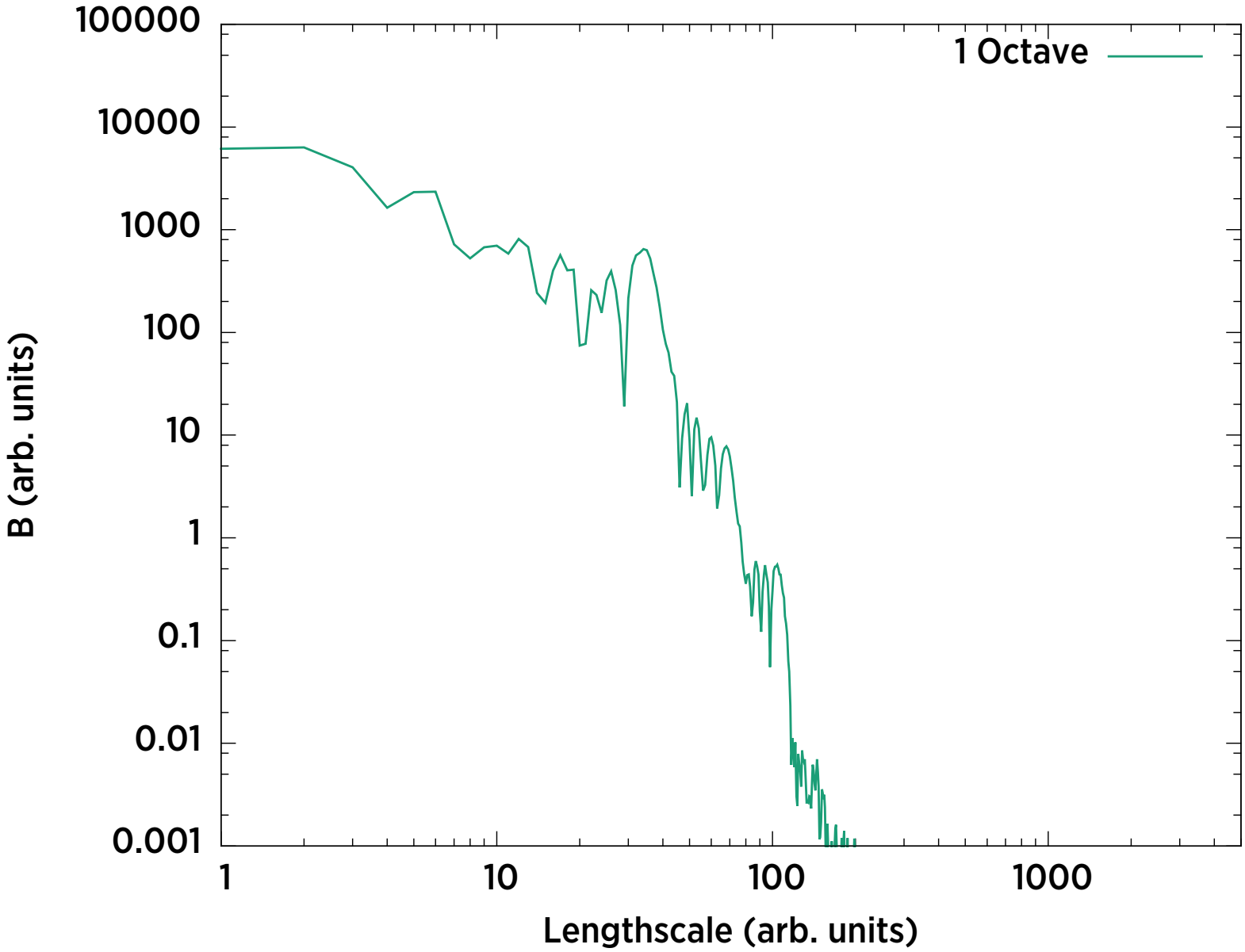
One evaluation



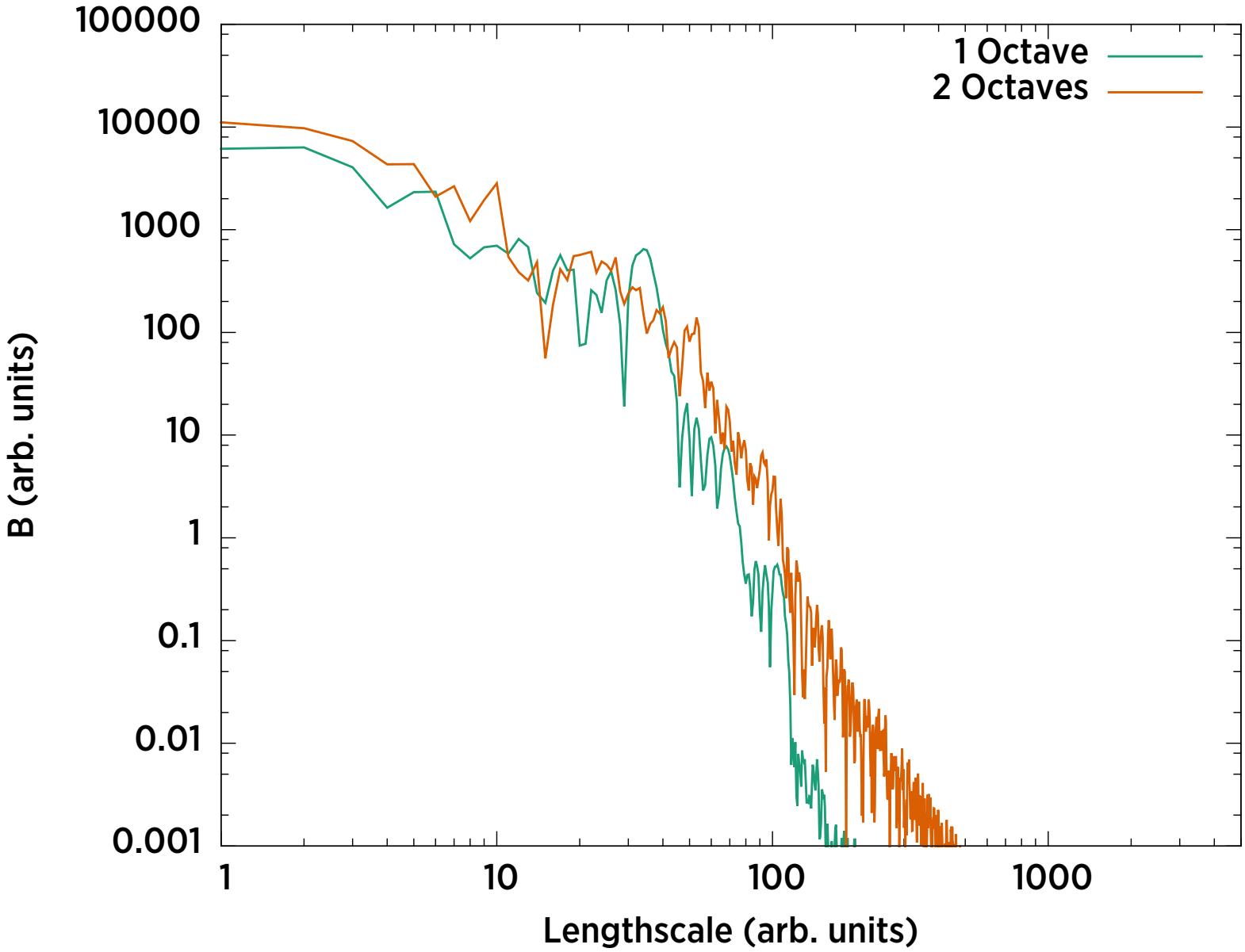
Multiple octaves, stacked



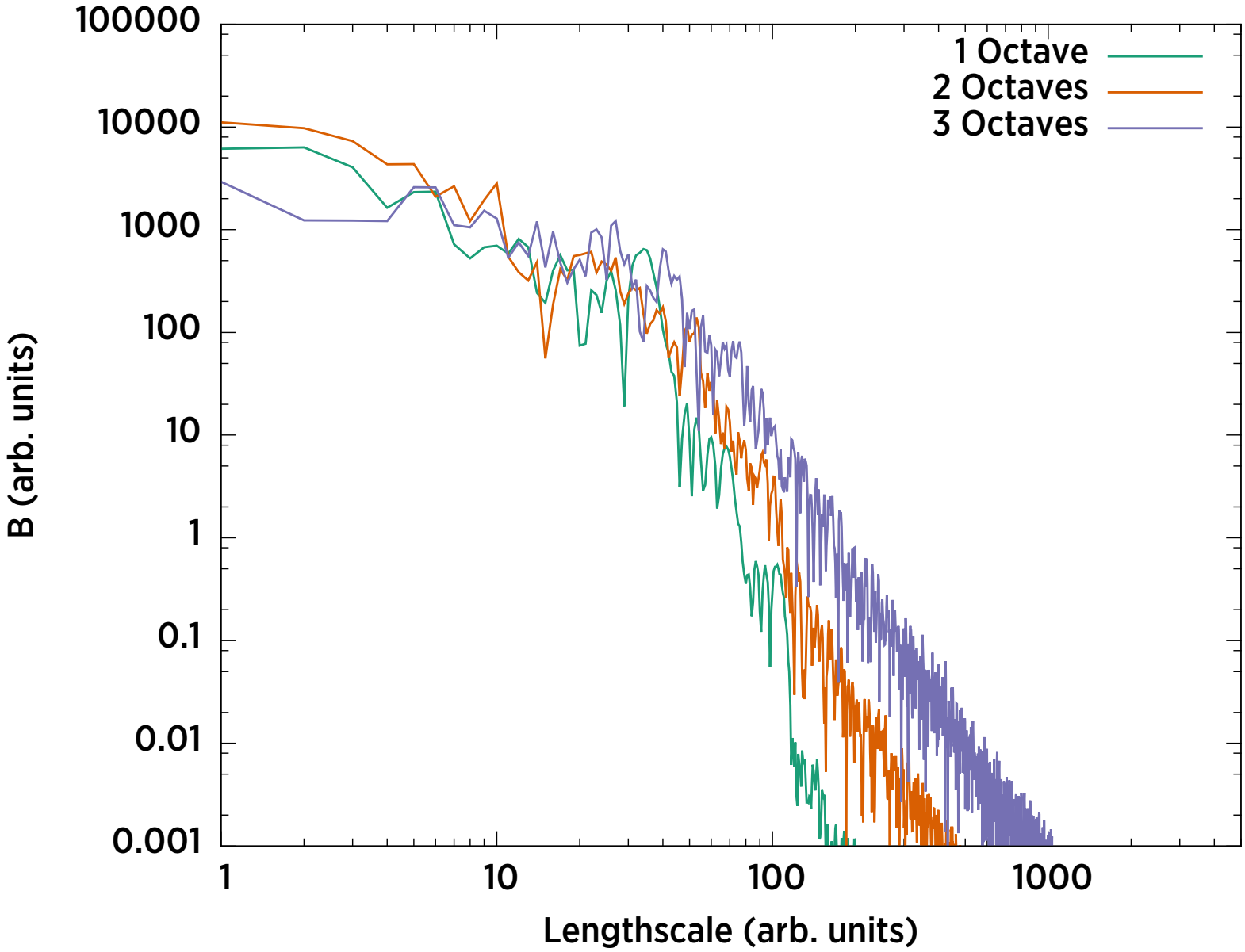
Spectrum


1 Octave
B (arb. units)
100000
10000
1000
100
10
1
0.1
0.01
0.001
1
10
100
1000
Lengthscale (arb. units)

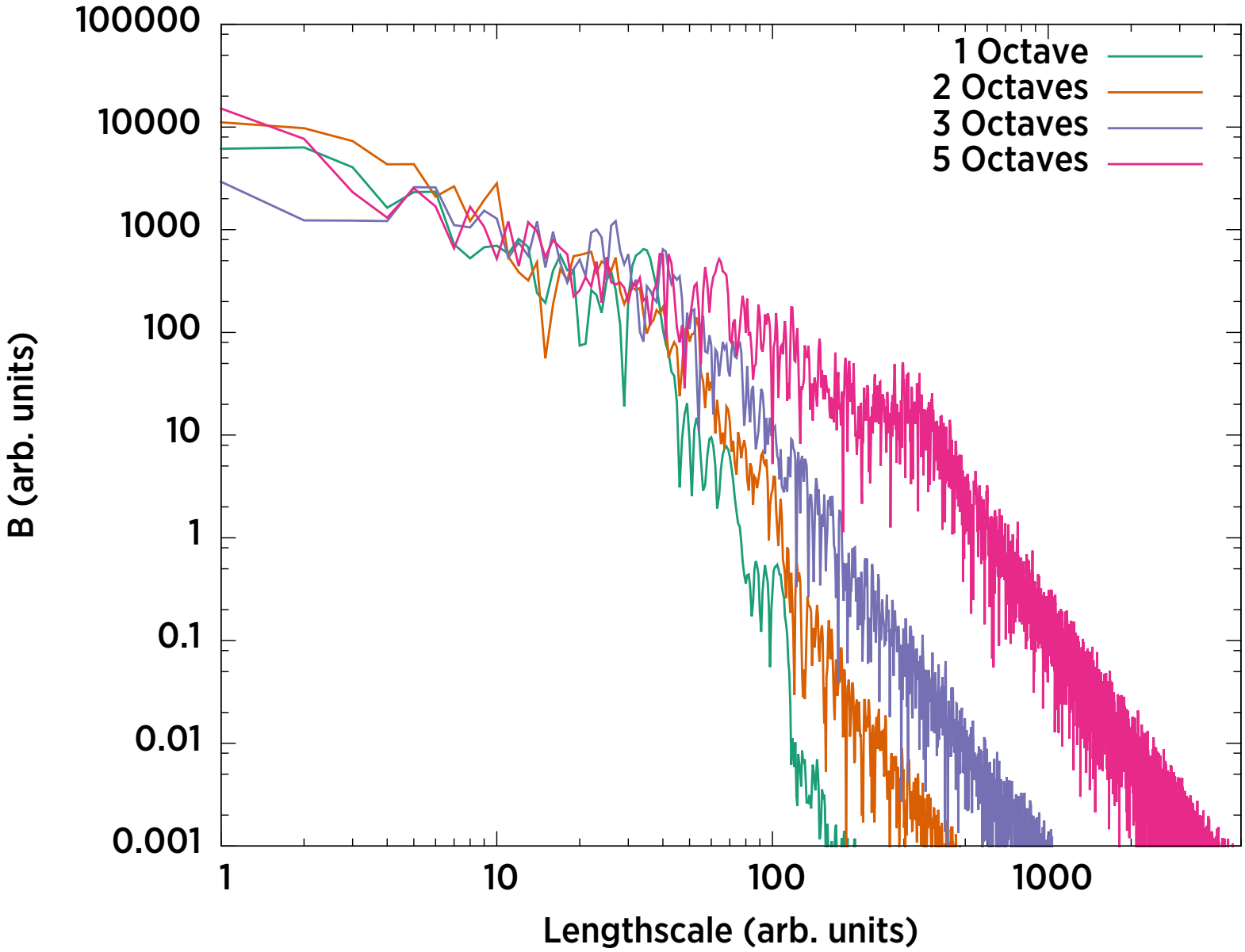
Spectrum


100000
10000
1000
100
10
1
0.1
0.01
0.001
B (arb. units)
1
10
100
1000
Lengthscale (arb. units)
1 Octave
2 Octaves

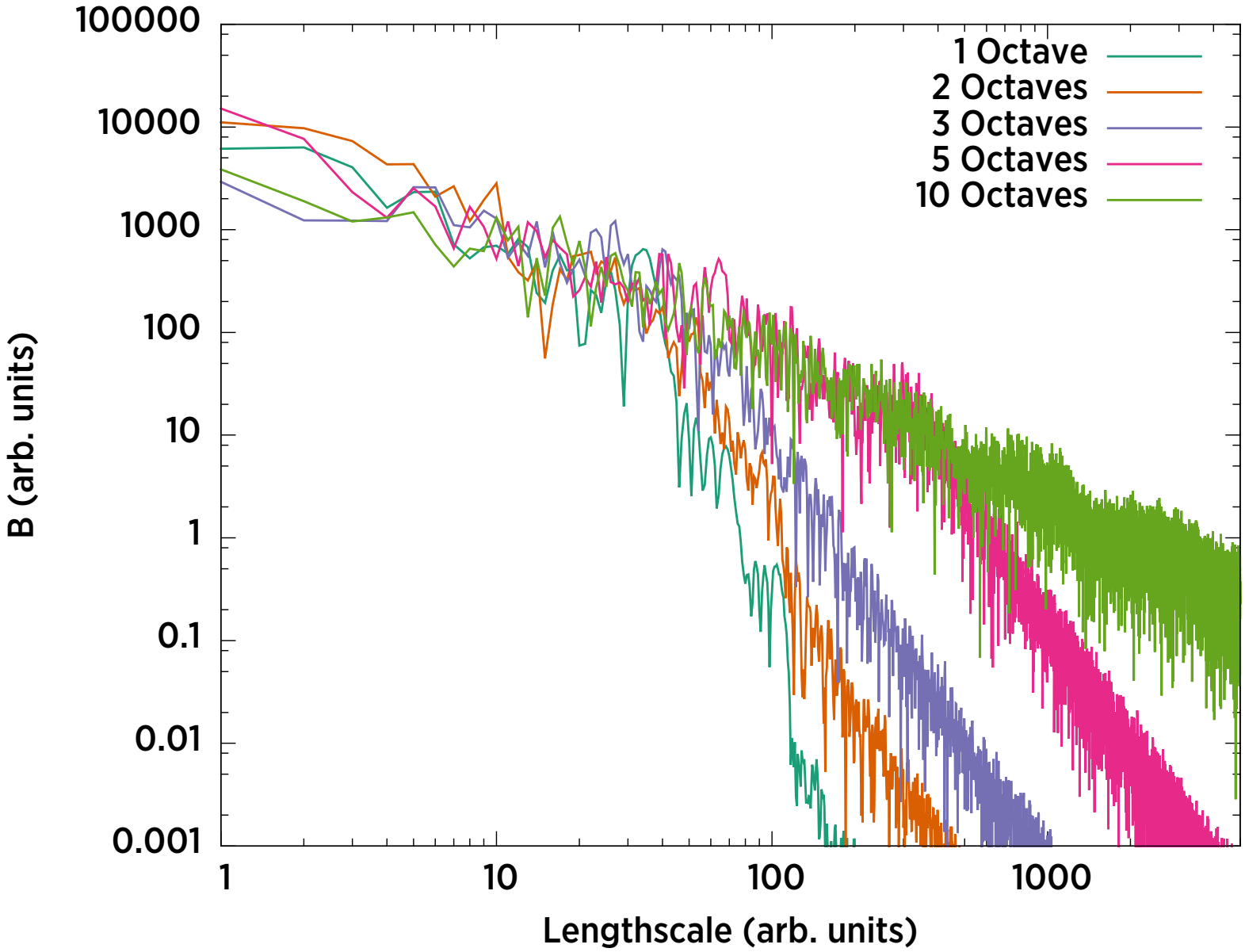
Spectrum


100000
10000
1000
100
10
1
0.1
0.01
0.001
B (arb. units)
1
10
100
1000
Lengthscale (arb. units)
1 Octave
2 Octaves
3 Octaves

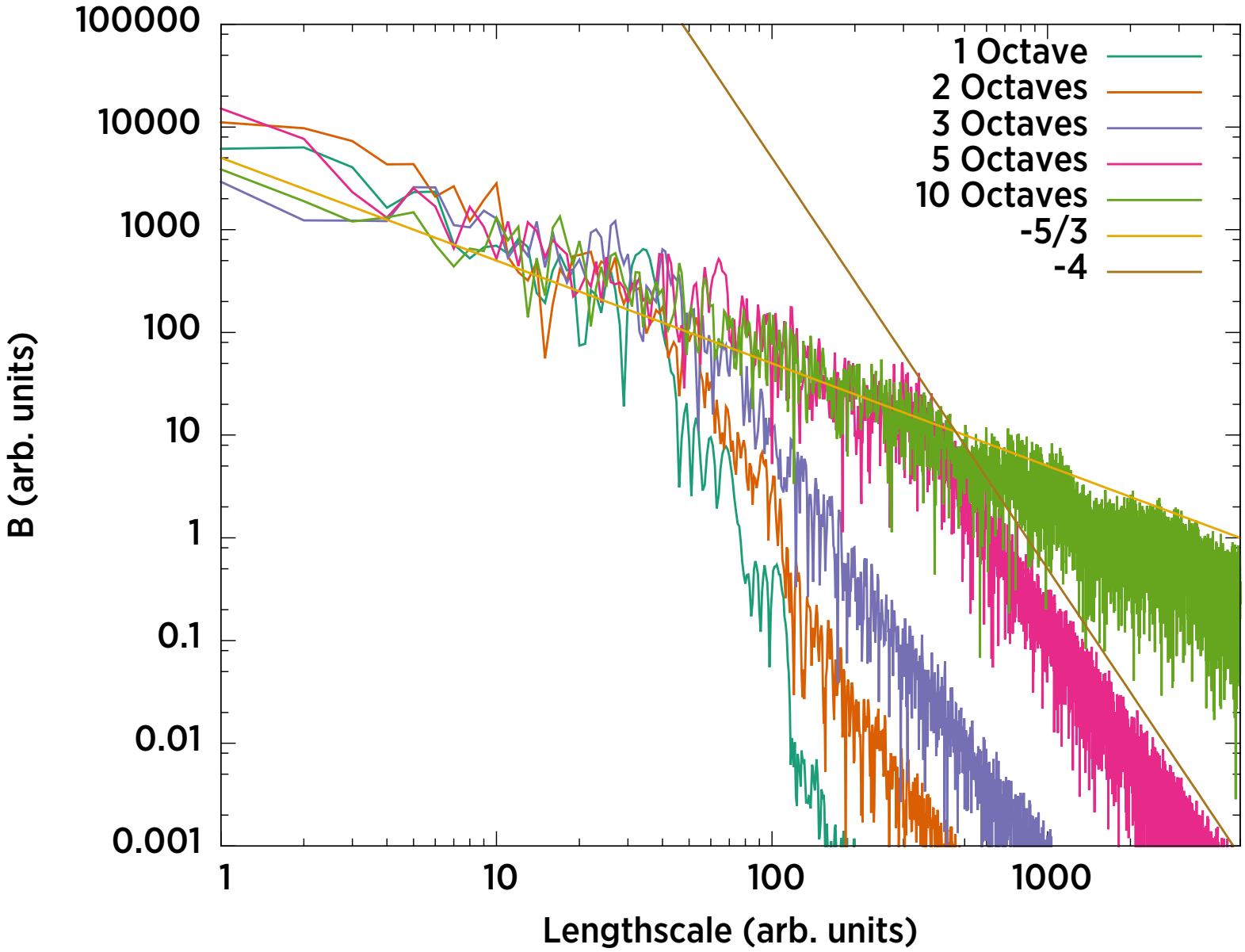
Spectrum


100000
10000
1000
100
10
1
0.1
0.01
0.001
B (arb. units)
1
10
100
1000
Lengthscale (arb. units)
1 Octave
2 Octaves
3 Octaves
5 Octaves

Spectrum


100000
10000
1000
100
10
1
0.1
0.01
0.001
B (arb. units)
1
10
100
1000
Lengthscale (arb. units)
1 Octave
2 Octaves
3 Octaves
5 Octaves
10 Octaves

Spectrum


100000
10000
1000
100
10
1
0.1
0.01
0.001
B (arb. units)
1
10
100
1000
Lengthscale (arb. units)
1 Octave
2 Octaves
3 Octaves
5 Octaves
10 Octaves
-5/3
-4

From Noise to Turbulent Fields

But these are still scalar noise functions! How do we get B-Fields from it?

From Noise to Turbulent Fields

But these are still scalar noise functions! How do we get B-Fields from it?

Dumb approach:

- 3 independent evaluations for 3 components
- Not divergence free

From Noise to Turbulent Fields

But these are still scalar noise functions! How do we get B-Fields from it?

Dumb approach:

- 3 independent evaluations for 3 components
- Not divergence free

Smarter:

- 3 independent evaluations for A
- $\vec{B} = \nabla \times \vec{A}$
- $\Rightarrow \nabla \cdot \vec{B} = \nabla \cdot (\nabla \times \vec{A}) = 0$

From Noise to Turbulent Fields

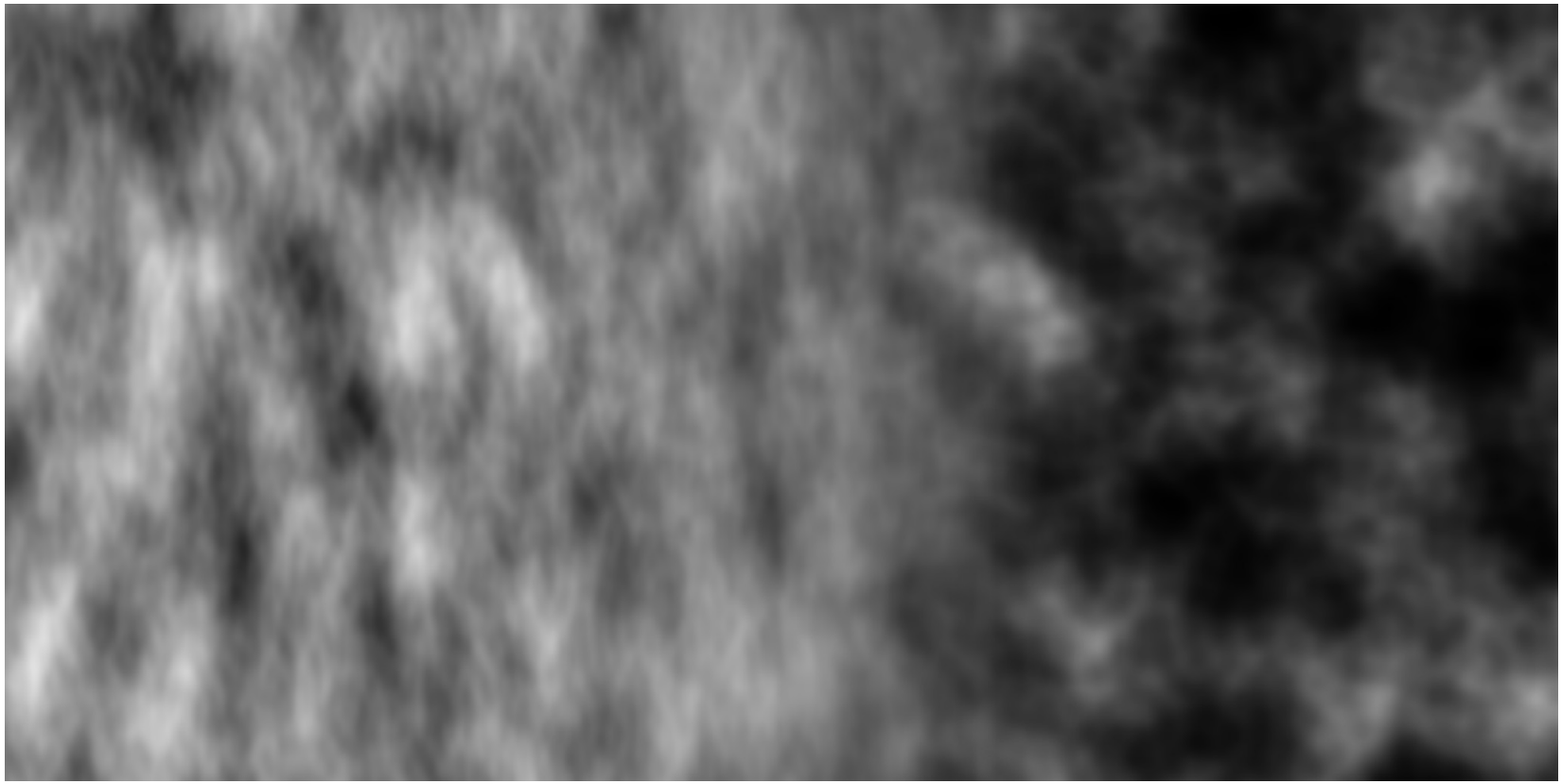
$$\vec{B}(\vec{x}) = \nabla \times \vec{A}(\vec{x})$$

- Isotropic in flat coordinate space.
- Anisotropy can be obtained by distorting coordinates.

From Noise to Turbulent Fields

$$\vec{B}(\vec{x}) = \nabla \times \vec{A}(\vec{x})$$

- Isotropic in flat coordinate space.
- Anisotropy can be obtained by distorting coordinates.



From Noise to Turbulent Fields

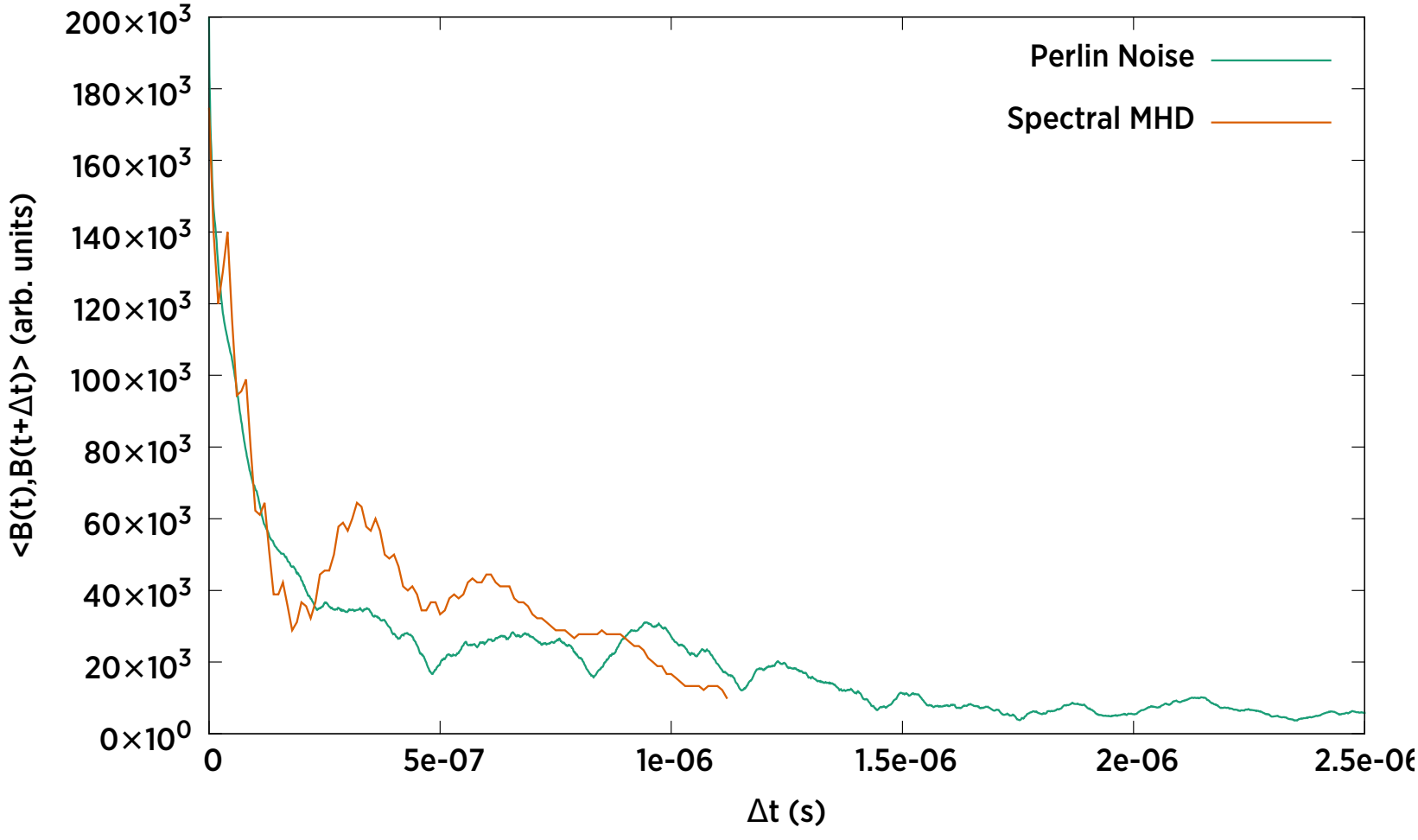
Weak Turbulence

- 3D noise advected along the background field.
- Or: One resting, one moving at $+v_A$, one moving at $-v_A$

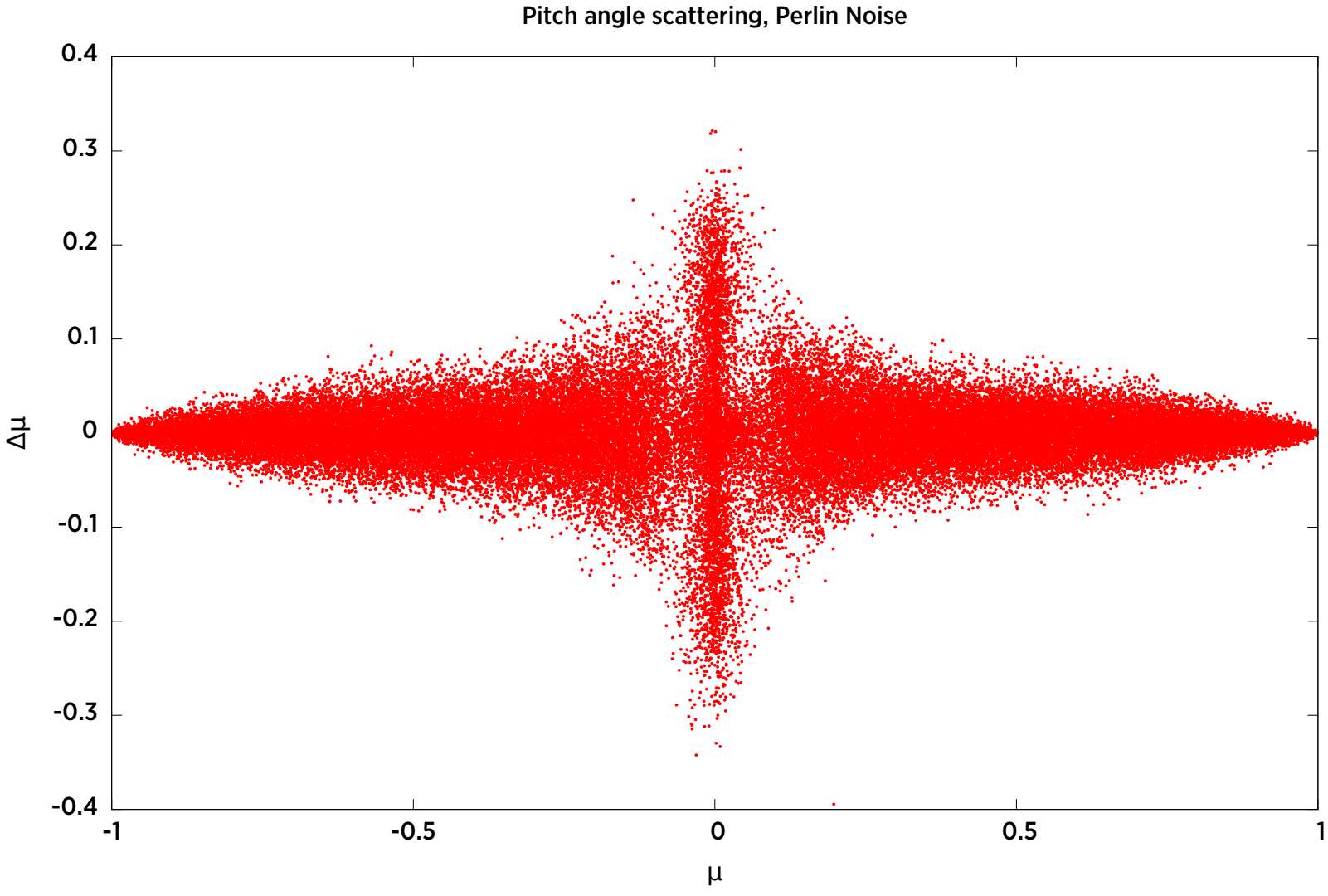
Strong Turbulence

- 4D noise, with 4th coordinate increasing with time.

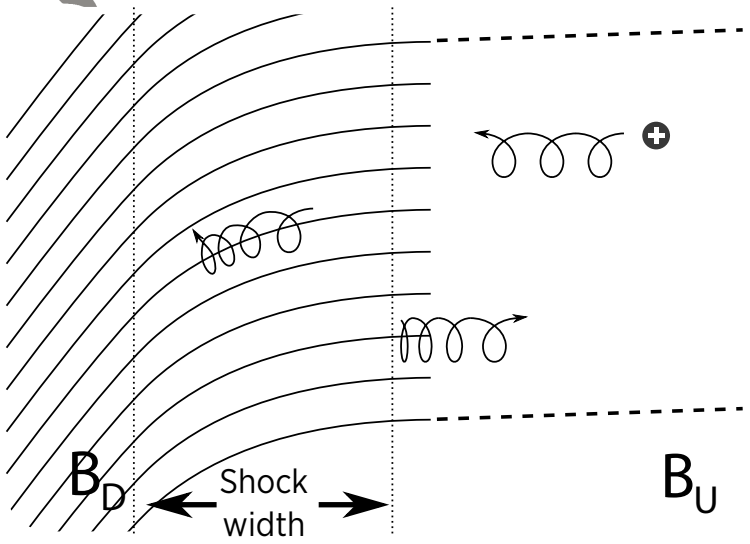
Field Autocorrelation



Pitch Angle Scattering

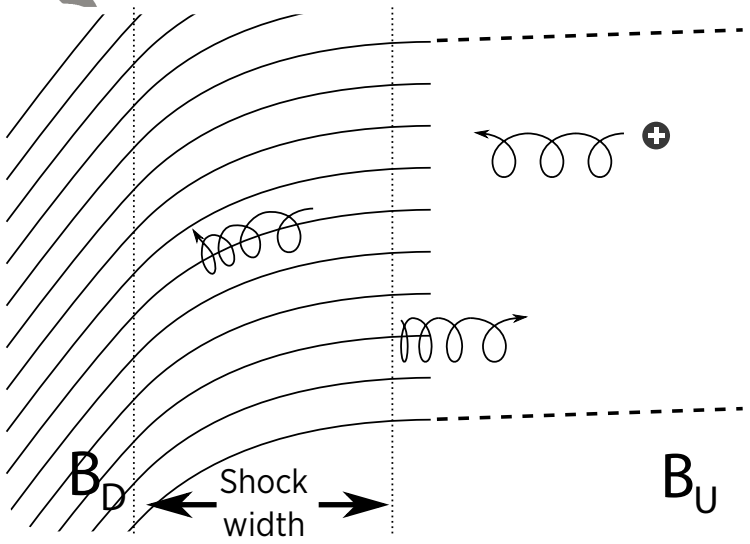


Application

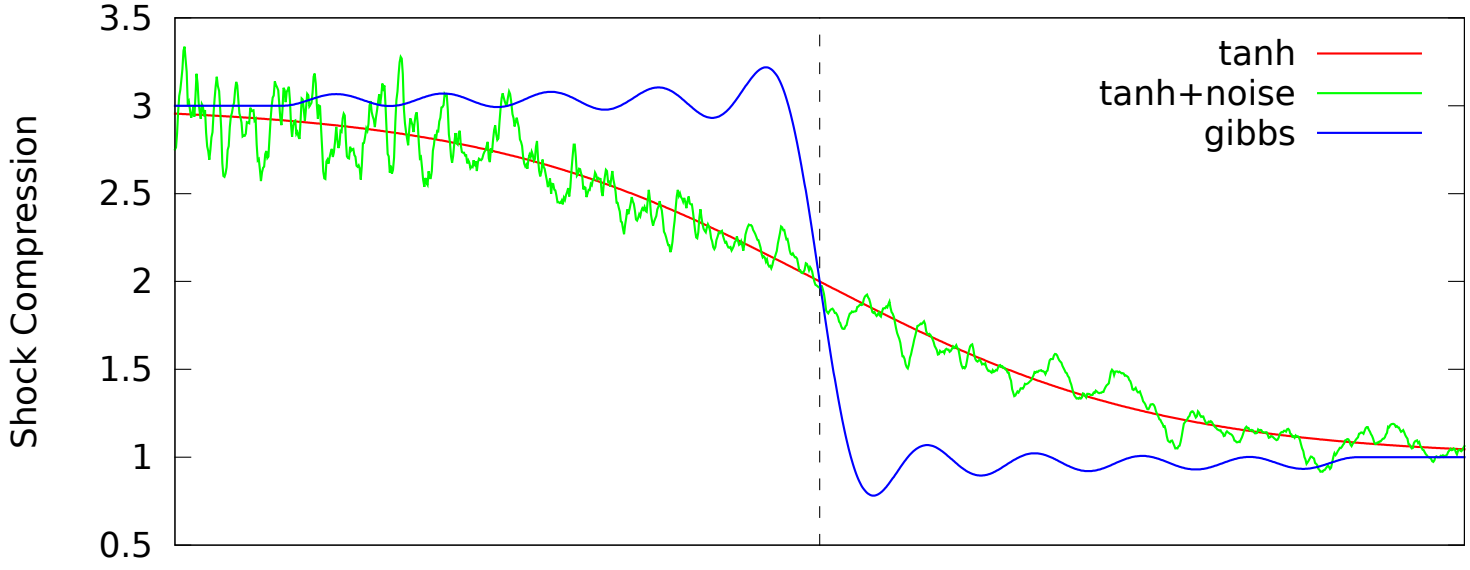


- Test particle simulations of particle accel. at shocks
- Kinetic scale structures
- Perlin Noise based turbulence

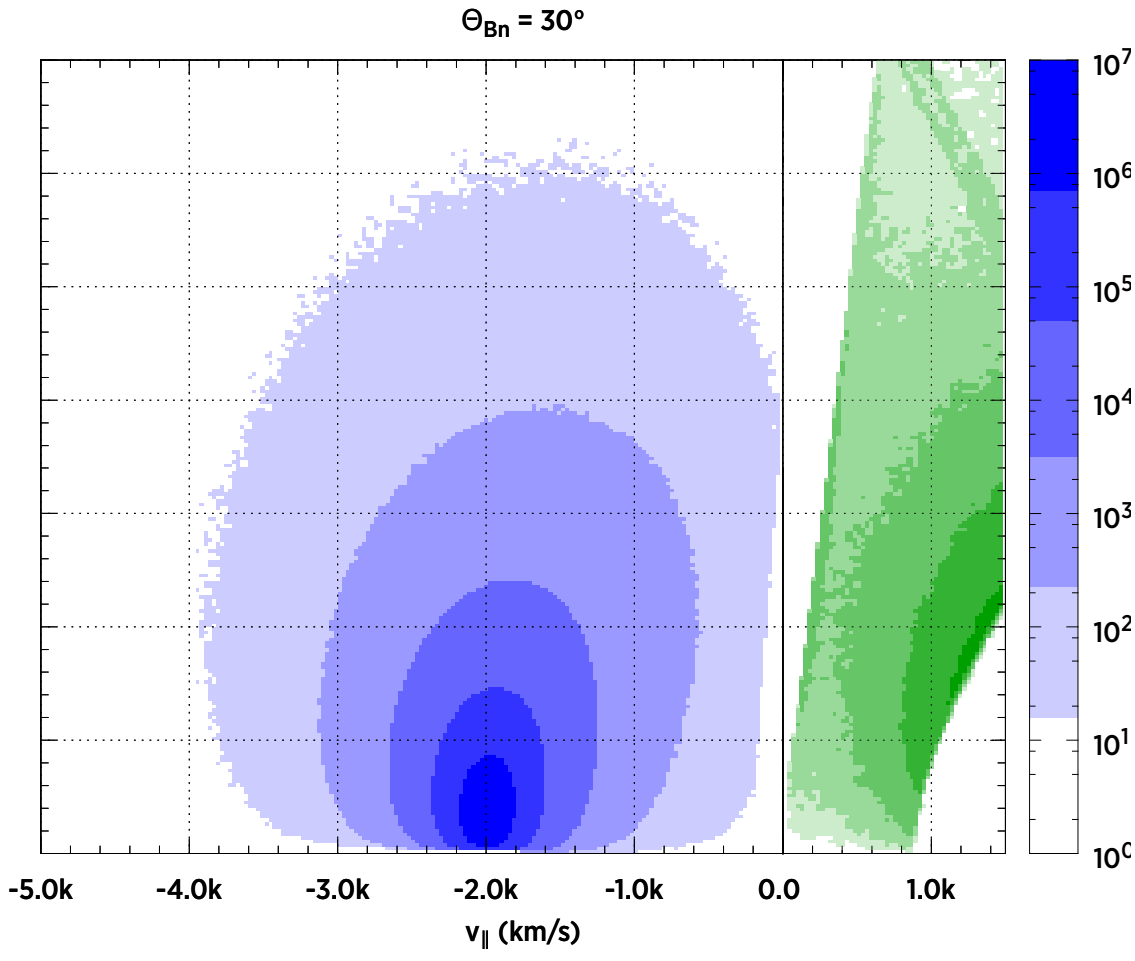
Application



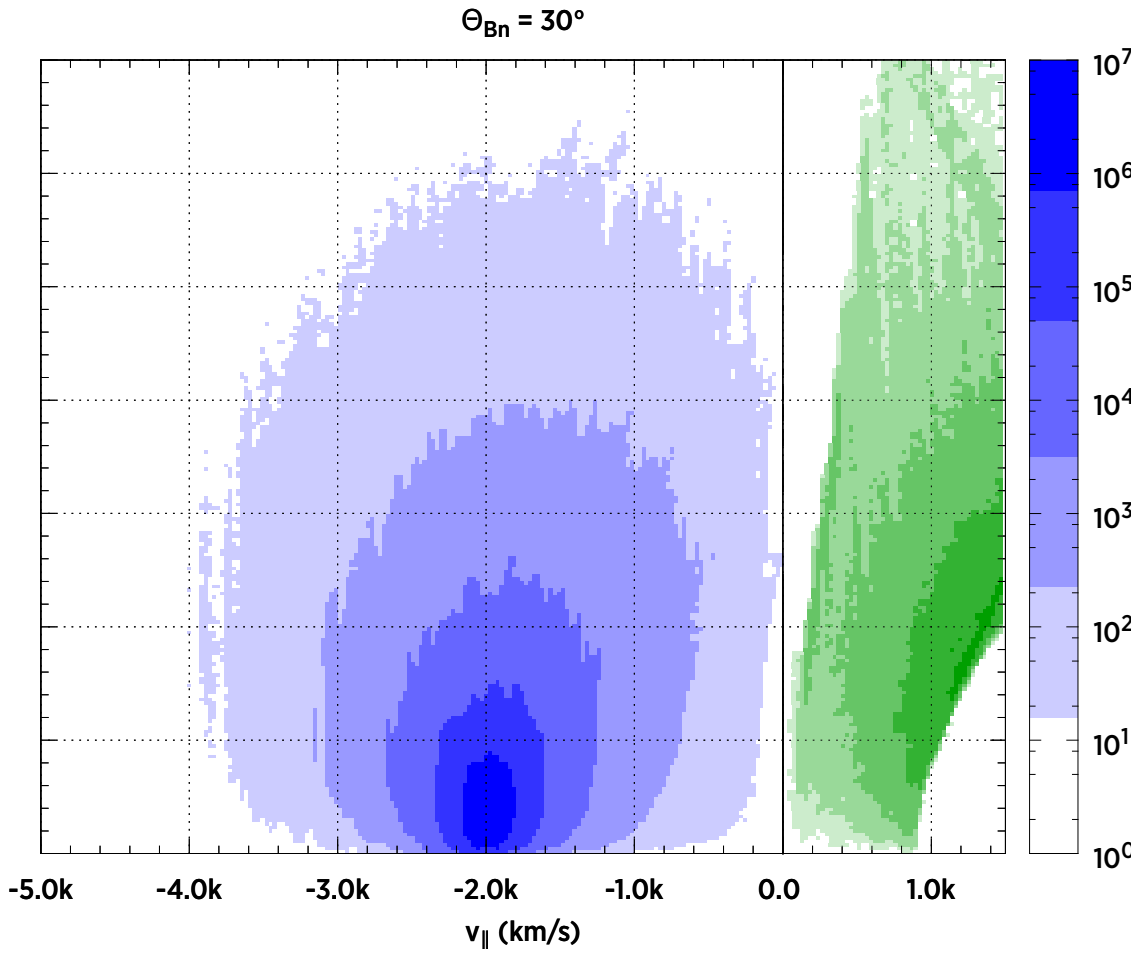
- Test particle simulations of particle accel. at shocks
- Kinetic scale structures
- Perlin Noise based turbulence



Preliminary results



Preliminary results



Conclusion

Perlin Noise

- Perlin Noise is a suitable way to create something like turbulence.
- Versatile and computationally inexpensive.

Outlook

- Further quantitative comparison to actual turbulence.
- Application to test-particle simulations of different scenarios.