

F1_vs_F2 — Help & Reference

Concepts • Parameters • Parser functions

All parameters and functions are deterministic. With identical seeds and config JSON, results are exactly reproducible.

Version: 2025-09-25

Project: F1_vs_F2 Web Playground

1) Concepts — How Images Are Generated

Pipeline. Sampling space → evaluate f1/f2 → generation mode → projection → isotropic fit → draw markers.

Sampling (X–Y space)

Points are taken either on a **grid** (regular lattice) or **randomly** (uniform in the chosen ranges). Grid has a **phase**: *on_lines* (aligned on grid lines) or *between_lines* (shifted by half a step).

Functions

Two expressions — **f1(x,y)** and **f2(x,y)** — are evaluated for every sampled point. Their outputs feed the next stage per the chosen **Generation mode**.

Generation modes

- **F1_VS_F2**: $u=f1$, $v=f2$ — canonical phase-space view.
- **F1_VS_X1**: $u=f1$, $v=x$ — banded patterns.
- **F1F2_VS_F1**: $u=f1+f2$, $v=f1$ — sheared carpets.
- **INDEX_VS_F1**: $u=index$, $v=f1$ — temporal ribbons.

Projection

Converts (u,v) to drawable (x,y): **Rectilinear** (identity), **Polar** ($u \rightarrow \text{angle}$, $v \rightarrow \text{radius}$), **Aitoff/Hammer/Lambert/Mollweide** (cartographic warps). After projection, points are **isotropically fitted** into a square canvas — circles stay circles.

Styling & compositing

Markers (pixel/point/circle/square/star) with **spot size**, **alpha**, **foreground/background** colors. Overdraw builds tone and texture.

Seeds & reproducibility

points_seed controls sampling; **style_seed** affects RANDOM choices; **func_seed** drives parser RNG (*rand/uniform/gauss*), placeholders (p1...p8), and salted *hash/noise*.

Tip: toggle “Exact render always” for final checks and exports; keep it off for rapid exploration.

2) Parameters — Effects and Guidance

Functions

- **f1(x,y), f2(x,y)**: core creative expressions. Nonlinearities add bands; discontinuities add edges and moiré.
- **Function seed**: seeds parser randomness and placeholders; change to explore stable variants.

Space

- **Sampling**: grid exposes symmetries; random feels granular/organic.
- **Grid phase**: on_lines vs between_lines (half-step shift) — reduces aliasing.
- **Grid step (on/between)**: smaller step → more points & finer interference; min is 0.001.
- **X/Y ranges**: narrow ranges zoom into micro-structure; wide ranges show global shape.
- **Points (random)**: density when sampling randomly.

Render

- **Projection**: rectilinear (neutral); polar (rings/spokes); cartographic (continents/voids).
- **Marker**: pixel/point/circle/square/star — from raw stippling to decorative geometry.
- **Spot size**: footprint per point; large sizes merge into masses.
- **Rotation**: rotates (u,v) pre-projection; breaks axial alignment.
- **Alpha**: low → tonal smoothing; high → crisp dots.
- **Foreground/Background**: palette; high contrast emphasizes structure.
- **DPI**: 144–300 for screen, 450–600 for print.
- **Exact render always**: disable preview downsampling.

Export & Presets

Save/Load preset (.json) preserves full state; Download PNG exports current canvas.

3) Parser Functions — Reference & Creative Tips

Variables: **x**, **y**. Placeholders: **p1...p8** (stable per Function seed). All functions are pure and deterministic.

Basic math & constants

sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, log, log10, exp, sqrt, abs, floor, ceil, pow, pi, e, tau.

Angles & polar helpers

atan2(y,x), angle(x,y), hypot(x,y)/length(x,y), r2(x,y), polar_r(x,y), polar_theta(x,y).

Use for radial masks, spokes/rings, rotation-invariant patterns.

Shaping / clipping

clamp(x,a,b), saturate(x), relu(x), lrelu(x,a), softplus(x), softsign(x), sign(x)/sgn(x), sigmoid(x)/logistic(x).

Control dynamic range; build plateaus and soft thresholds.

Interpolation & thresholds

lerp(a,b,t)/mix(a,b,t), step(edge,x), smoothstep(a,b,x).

Morph fields; soft vs hard boundaries.

Periodic & tiling

fract(x), mod(x,y) (true modulo), wrap(x,a,b), tri(x), saw(x).

Create lattices, combs, moiré; combine with angle/radius for polar tiling.

Distance blends (exotic, powerful)

smoothmin(a,b,k), smoothmax(a,b,k).

Smoothly union/intersect shape fields; higher k → harder edge.

Waves & transforms

sinc(x), gamma(x,g), map(x,a,b,c,d).

Diffraction-like ripples; tone mapping; quick rescaling.

Stochastic (function-seeded)

rand(), uniform(a,b), gauss(mu,sigma), hash(x,y), noise(x,y).

All driven by Function seed for reproducibility. Use noise/hash for grain; uniform/gauss for randomized coefficients.

Creative recipes

- Interference carpet: $\sin(x) + \sin(y^2)$ mixed via smoothstep.
- Radial gate: $\text{saturate}(1 - \text{polar}_r(x,y))$ as a mask to confine detail.
- Tiled polar: $\text{wrap}(\text{angle}(x,y), -\pi, \pi)$ with $\text{tri}(r2(x,y)) \rightarrow$ flower-like spokes.
- Smooth union: $\text{smoothmin}(f1, f2, k)$ for creamy borders.
- Soft threshold: $\text{smoothstep}(t1, t2, f1)$ as a weight for $\text{lerp}(a, b, \cdot)$.