Dror Bar-Natan: Talks: Fields-0911:

Dror Bar-Natan: Academic Pensieve: 2009-11:

Hilbert's 13th Problem - Program

Pensieve Header: Programs for the Kolmogorov-Arnol'd solution of Hilbert's 13th problem.

```
SetOptions[Rasterize,
  RasterSize → 900, ImageSize → 900
 ];
SetOptions[Plot,
  AspectRatio → 1, Exclusions → None, ColorFunction → Hue, ColorFunctionScaling → False
 ];
SetOptions[Plot3D,
  PlotPoints → 150, ColorFunction → Hue, ColorFunctionScaling → False,
  ViewPoint → {-1, -2, 1}, ImageSize → 900, Exclusions → None
 ];
SetOptions[DensityPlot, PlotPoints → 150,
  ColorFunction → Hue, ColorFunctionScaling → False];
λ = N[Sqrt[5] - 1] / 2;
phi0[x_] := x;
Phi[ϕ0_, n_, mu_ , ff_ ] := Phi[ϕ0, n, mu, ff, 0];
Phi[
    ϕ0_, n_,
    mu_ (* "slope" *),
    ff_ (* "flat fraction" *),
    s_ (* "shift" *)
   ][x_] := Module[
   {xs = x - s / n, x0, y0, x1, y1, nu},
   y0 = ϕ0[x0 = Floor[n * xs] / n];
   y1 = ϕ0[x1 = x0 + 1 / n];
   If[(1 - ff) / 2 < (xs - x0) * n < (1 + ff) / 2, Return[(y0 + y1) / 2 + mu * (xs - (x0 + x1) / 2)]];
   nu = ((y1 - y0) / (x1 - x0) - mu * ff) / (1 - ff);
   If[(xs - x0) * n < 1 / 2, Return[y0 + nu * (xs - x0)]];
   y1 + nu * (xs - x1)
  ];
G[f_, Phi[phi0_, n_, mu_, ff_, s_]] := Module[
   {phi, as, bs, ms, v, extra},
   If[s == 0, extra = 0, extra = 1];
   phi = Phi[phi0, n, mu, ff, s];
   as = phi /@ ((s + Range[1 - extra, n] - (1 + ff) / 2) / n);
   bs = phi /@ ((s + Range[1 - extra, n] - (1 - ff) / 2) / n);
   ms = (s + Range[1 - extra, n] - 1 / 2) / n;
   Interpolation[
    Sort[Flatten[Table[
        v = f[ms[[i]], ms[[j]]];
        If[mu == 0,
         Point[as[[i]] + λ * as[[j]], v],
         {Point[as[[i]] + λ * as[[j]], v], Point[bs[[i]] + λ * bs[[j]], v]}
        ],
        {i, n + extra}, {j, n + extra}
       ]]] /. Point → List, InterpolationOrder → 1
   ]
  ];
f[x_, y_] := N[
   Re[Zeta[x + I (13 + 4 y)]] / 3
  ];
```

```
Phi[val_, opts___Rule] := Module[
  {
   phi0 = ϕ0 /. {opts} /. ϕ0 → Identity,
   n = Subdivisions /. {opts} /. Subdivisions → 5,
   mu = Slope /. {opts} /. Slope → 0,
   ff = FillFactor /. {opts} /. FillFactor → 0.8,
   shift = Shift /. {opts} /. Shift → 0,
   x0, x1, x2, x3, x4, y0, y1, y2, y3, y4, phi
  },
  y0 = phi0 /@ (x0 = (Range[0, n] - shift) / n);
  x1 = x0 + (1 - ff) / (2 n); x2 = x0 + 1 / (2 n); x3 = x0 + (1 + ff) / (2 n);
  y4 = phi0 /@ (x4 = x0 + 1 / n);
  y2 = (y0 + y4) / 2;
  y1 = y2 - mu * ff / (2 n); y3 = y2 + mu * ff / (2 n);
  phi = Interpolation[
    Thread[{Join[x0, x1, x2, x3], Join[y0, y1, y2, y3]}],
    InterpolationOrder → 1
   ];
  val /. {
    ϕ → phi, X0 → x0, X1 → x1, X2 → x2,
    X3 → x3, X4 → x4, Y0 → y0, Y1 → y1, Y2 → y2, Y3 → y3, Y4 → y4
   }
 ]
```