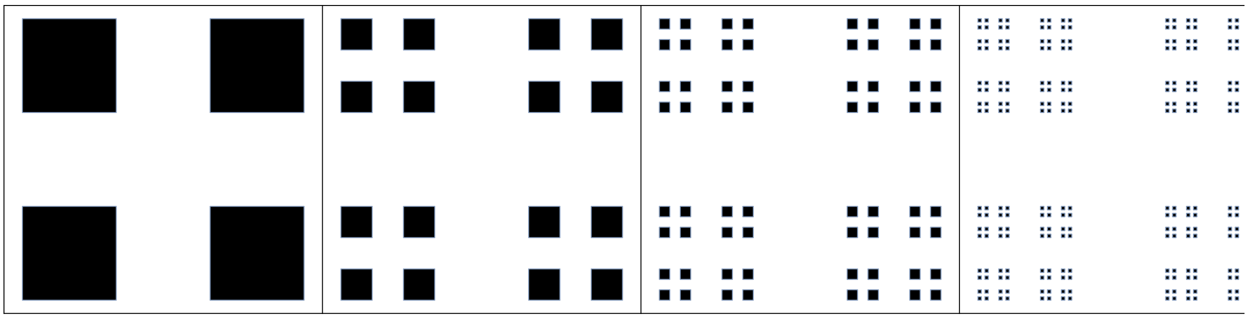
MAT1750: Shadows of the Cantor Square

Charlene Chu - November 28, 2017

Problem

```
GraphicsGrid[{Table[CantorMesh[i, 2, BaseStyle → Black], {i, 1, 6}]},
 Frame → All, ImageSize → 1000]
```



Draw the above approximations of the Cantor square C^2. Then rotate C^2 by an angle θ and project the rotation vertically, draw that projection (while explaining why it is what you think it is), and compute its measure. Perhaps also use Manipulate[..., {θ, 0, $\pi/2$}]?

Solution

We could use CantorMesh to draw the approximations of the Cantor square, but we cannot extract information (data points) from this MeshRegion object. So, instead, we will create a function *f*[*n*, *θ*] that draws it along with its rotation by *θ* and its projection (shadow), and displays the measure.

Before we begin, we here are some helper functions that we will use. CantorSequence[*n*] returns a list of 0s and 1s, where 1 represents the interval we keep and 0 represents the interval we remove when constructing the *n*-th iteration of Cantor set. Using this sequence, we can obtain the a list of the intervals of the *n*-th iteration of the Cantor set using CantorIntervals[*n*]. These intervals will give us all the information we need to draw the approximations along with its rotation by *θ* and vertical projection, as well compute its measure.

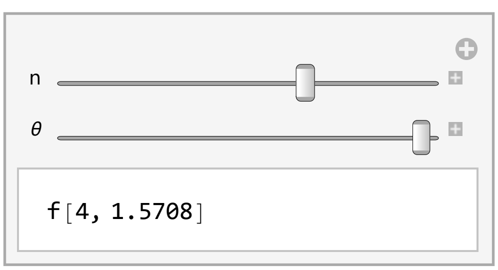
```
CantorSequence[0] = {1};
CantorSequence[n_Integer] := Module[{s = {}},
   Do[s = If[CantorSequence[n - 1][[i]] == 1, Join[s, {1, 0, 1}], Join[s, {0, 0, 0}]],
    {i, Length[CantorSequence[n - 1]]}];
   s
  ];
```

```
CantorIntervals[n_Integer] := Module[{l, output = {}},
   l = Table[k/3^n, {k, 0, 3^n}][[Flatten[Position[CantorSequence[n], 1]]]];
   Do[AppendTo[output, {i, i + 1/3^n}], {i, l}]; output
  ];
f[n_, θ_] := Module[{t, p, rp, i, l, m, o, r, s, px, py, mt},
   t = CantorIntervals[n];

   p = Join @@ Table[{{t[[k, 1]], t[[j, 1]]}, {t[[k, 1]], t[[j, 2]]},
       {t[[k, 2]], t[[j, 2]]}, {t[[k, 2]], t[[j, 1]]}}, {k, 2^n}, {j, 2^n}];
   rp  =  (({Re[#], Im[#]} &) /@ # &) /@ (Apply[Complex, p, {2}] * Exp[ⅈ θ]);

   i = Interval @@ ({Min[#], Max[#]} & /@ rp[[All, All, 1]]);
   l = (({#, 0} &) /@ # &) /@ (List @@ i);
   m = RegionMeasure[i];

   o = Polygon[p];
   r = Polygon[rp];
   s = Line[l];

   px = Flatten[rp[[All, All, 1]]];
   py = Flatten[rp[[All, All, 2]]];
   mt = Style[Text[m , {(Min[px] + Max[px])/2, (Min[py] + Max[py])/2} ], Bold];

   Graphics[{Pink, Opacity[.2], o, Opacity[.5], r, Black, Thick, Opacity[1], s, mt}]
  ];
```

Example 1

Here is an interactive example using Manipulate.

```
Manipulate[f[n, θ], {n, 0, 6, 1}, {θ, 0, π/2}]
```



Example 2

Here is a poster of some of the combinations n and θ from the above interactive example.

```
GraphicsGrid[Table[f[n, θ], {n, 0, 5, 1}, {θ, {0, π/6, π/5, π/4, π/3, π/2}}],
 ImageSize → 1000, Frame → All]
```