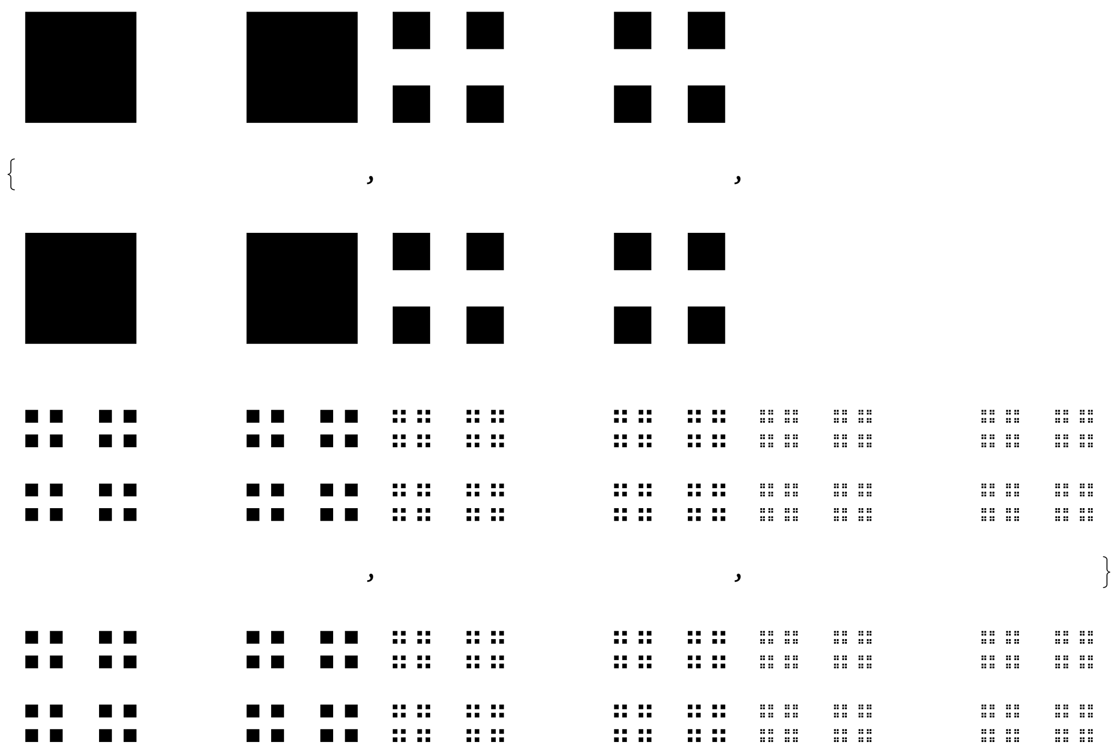
```
(* Generate the Cantor Set via an interative procedure*)

T0[x_] := x/3;
T2[x_] := 2/3 + x/3;
int[0] = {0, 1};
int[n_] := int[n] = Union[T0[int[n - 1]] , T2[int[n - 1]]]
(* Define rotation around the point (0.5,0.5) by 2 Pi th radians*)
R[v_, th_] :=
 {{Cos[2 Pi th], -Sin[2 Pi th]}, {Sin[2 Pi th], Cos[2 Pi th]}}.(v - {0.5, 0.5}) + {0.5, 0.5}

squares[n_, th_] := Flatten[Table[
   Polygon[{R[{int[n][[i]], int[n][[j]]}, th], R[{int[n][[i]], int[n][[j + 1]]}, th],
     R[{int[n][[i + 1]], int[n][[j + 1]]}, th], R[{int[n][[i + 1]], int[n][[j]]}, th]}],
   {i, 1, 2^(n + 1), 2}, {j, 1, 2^(n + 1), 2}]]

Table[Graphics[squares[i, 0]], {i, 1, 5}]
```



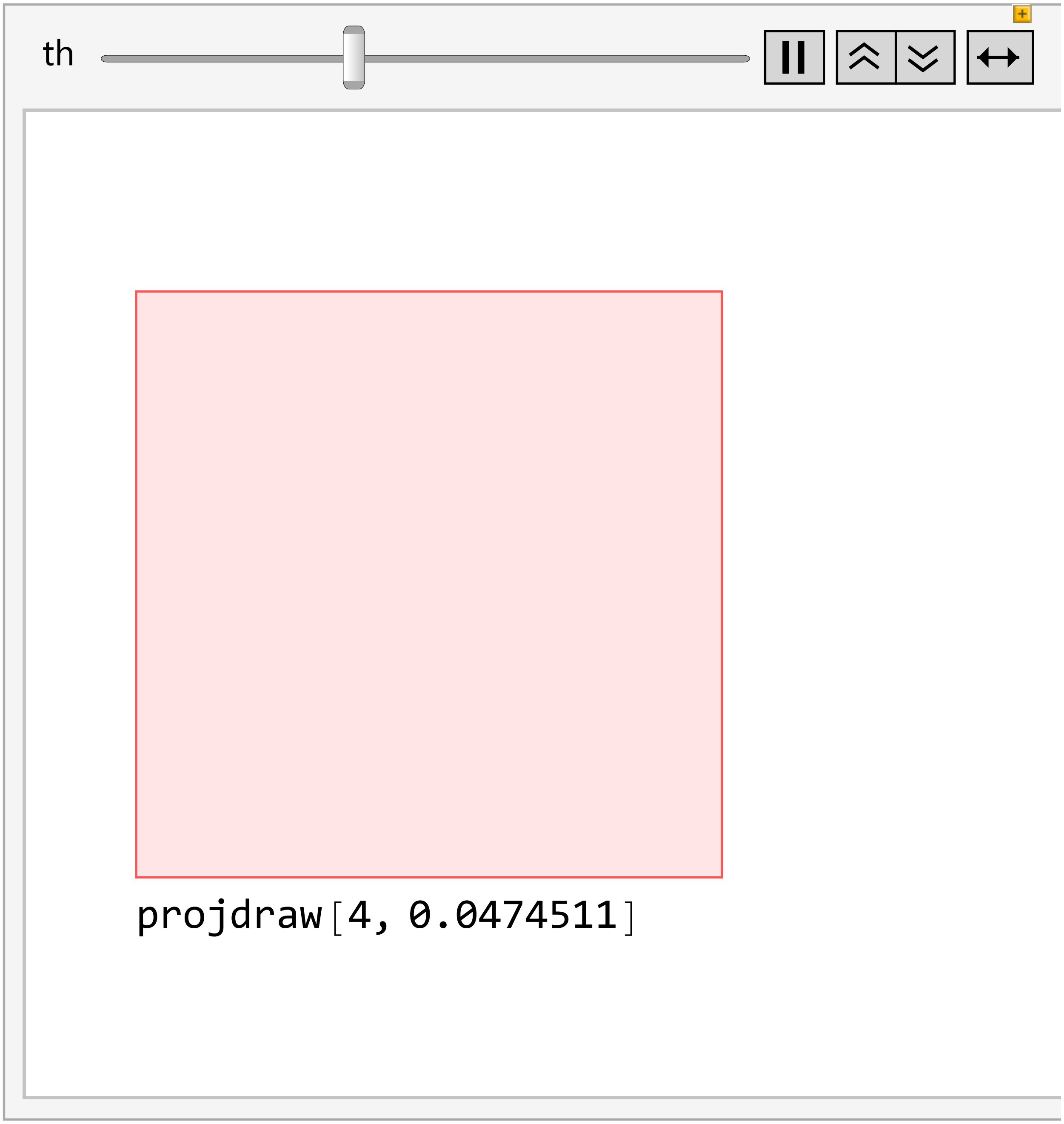
```
(* Here are the first 5 iterations of the Cantor Aerogel*)

(* The function lines computes the projection of the nth iterate of the Cantor aerogel
  on the x axis. Note that this approach only works for 0 ≤ 2 Pi theta ≤ Pi/4,
but the function we are interested in (i.e. the measure of this projection)
 is completely determined by its behaviour for 2Pi theta between 0 and Pi/4*)
```

```
lines[n_, th_] := lines[n, th] = Module[{dummy, plines, i, j},
   plines = Flatten[Table[Interval[{R[{int[n][[i]], int[n][[j + 1]]}, th][[1]],
        R[{int[n][[i + 1]], int[n][[j]]}, th][[1]]}],
      {i, 1, 2^(n + 1), 2}, {j, 1, 2^(n + 1), 2}]];
   dummy = plines[[1]];
   For[i = 2, i ≤ Length[plines], i++, dummy = IntervalUnion[dummy, plines[[i]]]];
   dummy
  ]

(* go through the output of lines and make them points in 2D space, then draw them.*)
projdraw[n_, th_] := Module[{TwoDlines, i}, TwoDlines = {};
  For[i = 1, i ≤ Length[lines[n, th]], i++, TwoDlines =
    Append[TwoDlines, {{lines[n, th][[i]][[1]], 0}, {lines[n, th][[i]][[2]], 0}}]];;
  Graphics[Line[TwoDlines], PlotRange →
    {{0.5 - Sqrt[2] / 2, 0.5 + Sqrt[2] / 2}, {-0.05, 0.05}}]]

projdraw[3, 0]
```

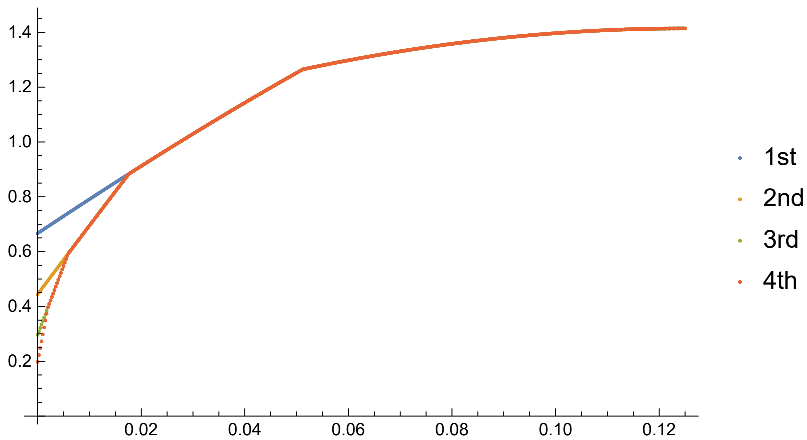


```
pic[n_] := Magnify[Animate[
   Column[{Graphics[squares[n, th], PlotRange → {{0.5 - Sqrt[2] / 2, 0.5 + Sqrt[2] / 2},
        {0.5 - Sqrt[2] / 2, 0.5 + Sqrt[2] / 2}}], projdraw[n, th]}],
   {th, 0, 0.125}, AnimationDirection → ForwardBackward, Alignment → Center], 2]
```

```
pic[4]
```



```
(* This behaviour is occuring because as we increase theta,
the projection of the 4th iteration of the Cantor Aerogel becomes
 indistinguishable from the projection of the 3rd, because the squares
 involved overlap one another vertically. Continuing to increase theta,
the 3rd iteration becomes indistinguishable from the 2md,
and so on, until we reach an angle of rotation of Pi/4,
at which point any iteration will be indistinguishable from the
 0th order iteration (i.e. a square of side length 1). This square,
rotated Pi/4 radians, has a vertical projection with measure sqrt(2),
which is exactly the limiting behaviour we see in the area calculation below*)

area[n_, th_] := Sum[lines[n, th][[j]].{-1, 1}, {j, 1, Length[lines[n, th]]}]

arealist[n_] := Table[{th, area[n, th]}, {th, 0, 0.125, 0.0005 / 2}]

ListPlot[{arealist[1], arealist[2], arealist[3], arealist[4]},
 PlotLegends → {"1st", "2nd", "3rd", "4th"}]
```




```
(* We can clearly see the phenomenon described above in this
 graph. The areas of the various projections start out distinct,
but merge over time, and converge to sqrt(2)*)
```