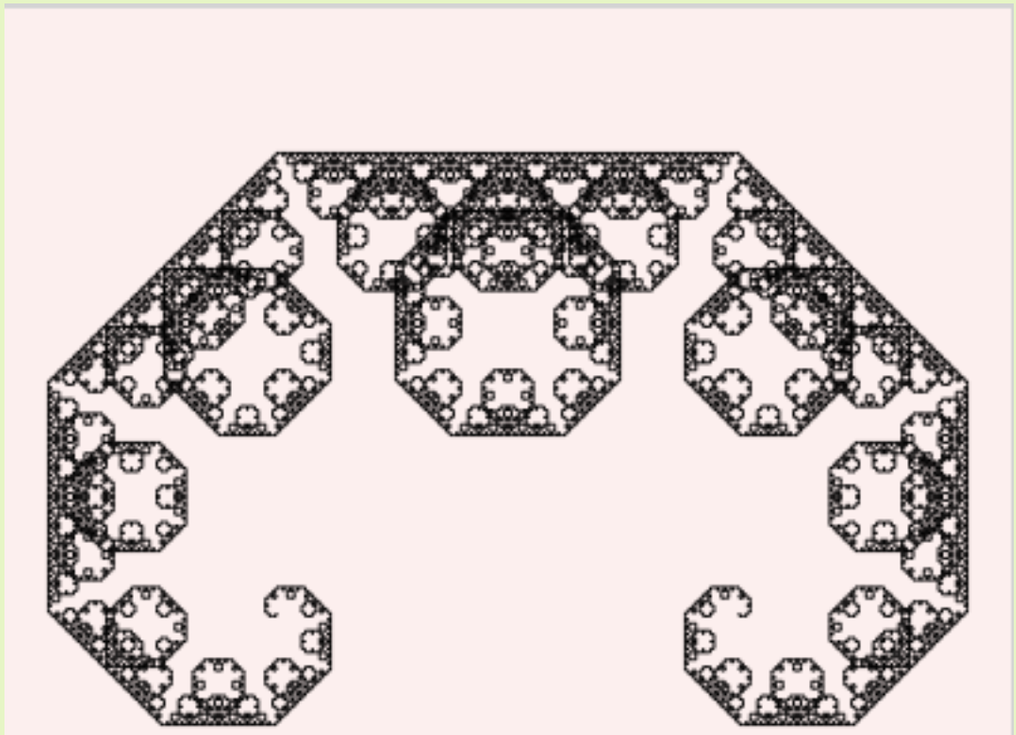
課題1

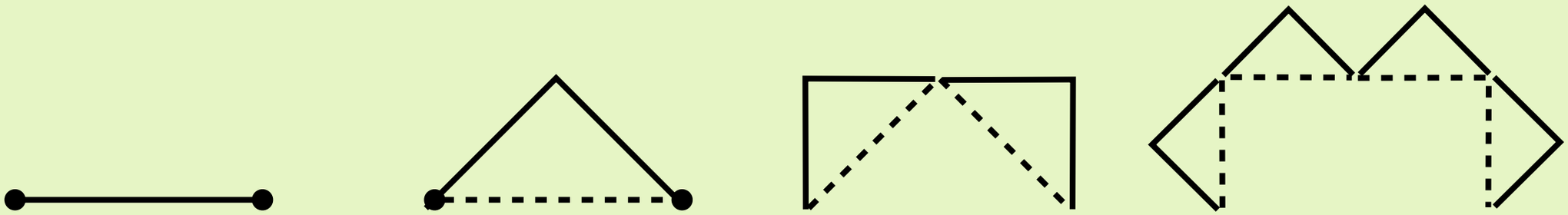
以下の図のようなLévy C曲線を描画するプログラムを作れ．Lévy C曲線の定義はつぎのスライドに示す．以下の図は深さ14の曲線になる．深さnの曲線の描画関数を `kadai1(n)` として定義せよ．ただし，初期点を`(0.27, 0.4), (0.73, 0.4)`とせよ．



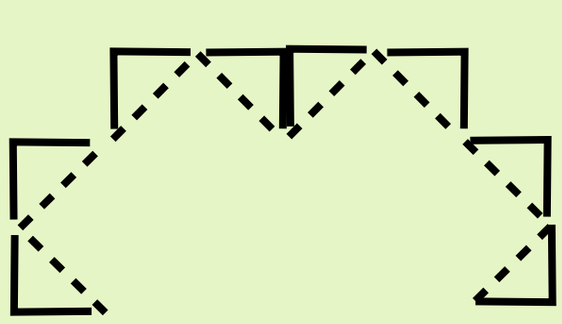
深さ14のLévy C 曲線

Lévy C 曲線の定義

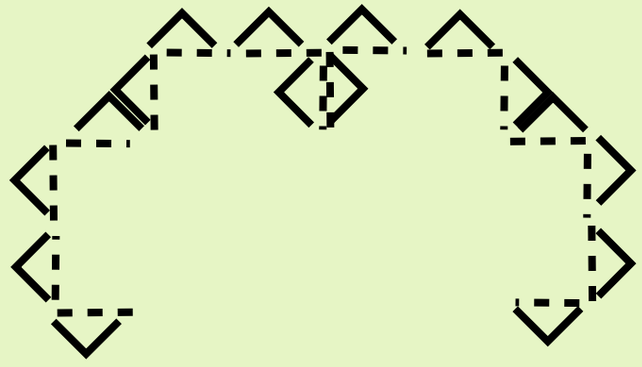
- Lévy C曲線は以下のように段階的に定義される.



d = 0　　d = 1　　d = 2　　d = 3



d = 4



d = 5

…

解答例1 (1)

まず以下に示す関数を定義する.

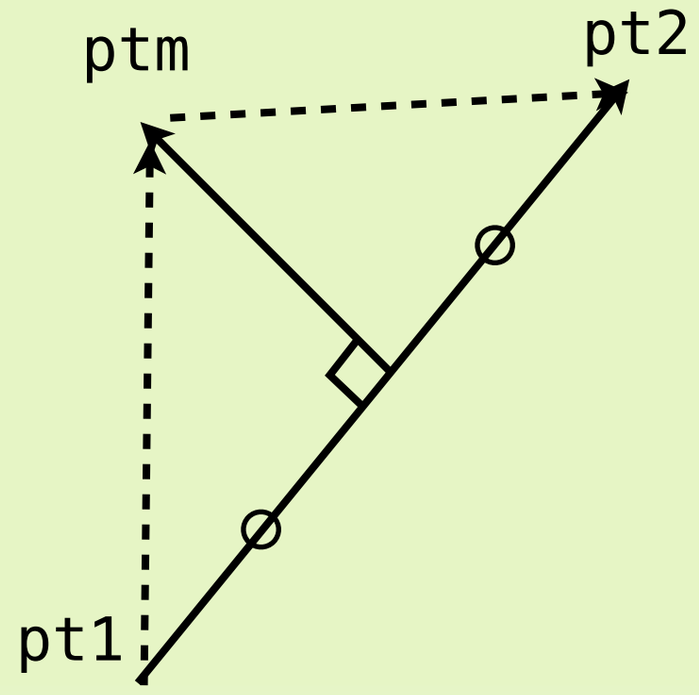
rotate_90	与えられたベクトルを90度回転させたベクトルを返す
vec_minus	ベクトルの引き算の結果を返す.
vec_plus	ベクトルの足し算
half	長さが半分のベクトルを返す

```
function rotate_90(pt){
   var [x, y] = pt
   return [y, -x]
}

function vec_minus(pt1, pt2){
   var [x1, y1] = pt1
   var [x2, y2] = pt2
   return [x2 - x1, y2 - y1]
}

function vec_plus(pt1, pt2){
   var [x1, y1] = pt1
   var [x2, y2] = pt2
   return [x1 + x2, y1 + y2]
}

function half(pt){
   var [x, y] = pt;
   return [x / 2, y / 2]
}
```

解答例１ (2)

あとは Lévy C曲線の定義にしたがって，プログラムを書けば良い.
基本的に pt1 -> pt2 を直接つなぐ代わりに pt1->ptm->pt2 と繋げば良く，ptmの計算が問題となる.

```
function levy_c(pt1, pt2, depth){
   if (depth == 0){
      draw_line(pt1, pt2)
   } else {
      var ptm =
        vec_plus(
          rotate_90(half(vec_minus(pt2, pt1))),
          half(vec_plus(pt1, pt2)))
      levy_c(pt1, ptm, depth - 1)
      levy_c(ptm, pt2, depth - 1)
   }
}

function kadai1(n){
   levy_c([0.27, 0.4], [0.73, 0.4], n)
}
```



課題２

以下の連立方程式の解の一つをNewton-Raphson法を用いて解くプログラム `kadai2(a)` を作れ．この関数は，xとyの値を並べたベクトル `[x, y]` を返すようにせよ．

$$\begin{cases} y &= \sin x, \\ y^2 &= ax \end{cases}$$

解答例２ (1)

まず，与えられた方程式から以下のように２つの関数を定義する

$$\begin{cases} f_1(x,y) &= y - \sin x \\ f_2(x,y) &= y^2 - ax \end{cases}$$

これより，Newton-Raphson法の反復公式はつぎのようになる.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} -\cos x & 1 \\ -a & 2y \end{pmatrix}^{-1} \begin{pmatrix} f_1(x,y) \\ f_2(x,y) \end{pmatrix}$$

これをプログラム化すれば，つぎのスライドのようになる.

解答例２ (2)

ここでは$|f_1^2+f_2^2| < \varepsilon$となったとき停止するようにする．初期値は(1, 1)としているが，適当に変えれば，出力される値も変化する．

原点に近い解は，(a, a)に近くなる．ただし，aが0.75くらいの値よりも大きくなると，(0, 0)以外の解はなくなってしまう（要確認）．

```
function kadai2(a){
    function f1(x, y){return y - Math.sin(x);}
    function f2(x, y){return y * y - a * x;}
    var x = 1.0;
    var y = 1.0;
    var EPS = 1.0e-10
    while (true){
        var ff1 = f1(x, y);
        var ff2 = f2(x, y);
        if (ff1 * ff1 + ff2 * ff2 < EPS) break;
        var a11 = -Math.cos(x);
        var a12 = 1;
        var a21 = -a;
        var a22 = 2 * y;
        var det = a11 * a22 - a12 * a21;
        x = x - (a22 * ff1 - a12 * ff2) / det;
        y = y - (-a21 * ff1 + a11 * ff2) / det;
    }
    return [x, y]
}
puts(kadai2(0.1))
```

```
0.10033636446368767,0.10016809558281038
```