D2 The Koch Snowflake

The Koch snowflake is obtained by starting with an equilateral triangle and repeatedly doing the following operation: on each edge add an equilateral triangle in the center with side-length one-third the length of the edge, pointing outwards. The Koch snowflake has finite area. But it has infinite perimeter.

D2.1 Drawing the Snowflake

The basic drawing tool in MATLAB is the `plot` function. The simplest form of this takes two vectors, giving the $x$- and $y$-coordinates respectively of a sequence of points, and `plot` joins them up with line segments.

We will draw an approximation to the snowflake by specifying the vertices of the polygon. The main loop takes the current list and produces the new list. The main operation here is to take two consecutive points on the old list and calculate the additional points for the new list.

Given points $A = (a_1, a_2)$ and $B = (b_1, b_2)$, we need to find points $U$, $V$ and $W$ for the following picture:

A bit of geometry yields that

$$U = \left(\frac{2a_1 + b_1}{3}, \frac{2a_2 + b_2}{3}\right)$$
and $W$ similarly. The point $V$ is obtained from $U$ by going a distance one-third of $|AB|$ at an angle of 60° to the current heading. The current angle or heading $\theta$ can be deduced by noting that $\tan \theta = (b_2 - a_2)/(b_1 - a_1)$. MATLAB has a function `atan2` that produces an angle in the full 0 to $2\pi$ range given the change in $y$ and the change in $x$. So we get code like:

% assuming A and B are 2-entry vectors
U = (2*A+B)/3;
W = (A+2*B)/3;
distAB = sqrt( sum( (B-A).^2 ) );
angAB = atan2( B(2)-A(2), B(1)-A(1) );
V = U + (distAB/3) * [ cos(angAB+pi/3), sin(angAB+pi/3) ];

The overall code for the snowflake stores a sequence of points in a table/matrix. Each row has two values, the $x$- and $y$-coordinates. To access a point we need the whole row: we write $P(i,:)$ to get row $i$. Also, $\text{size}(P,1)$ is the number of rows in $P$. Each iteration we add three points between every successive pair of existing points.

`triangleKoch.m`