MXB361 Module 2: Prac 1

Computational Structures in DNA: space-filling curves

Topics:

- Generating a Hilbert curve (Section 1.1);
- Your tasks this week (Section 1.2);

1.1 Generating a Hilbert curve

The Hilbert curve is a space-filling curve, and as with other fractals it is generated recursively. To get started, divide your square region into 4 sub-regions, and draw an upside-down 'square' U with vertices at the centres of the 4 sub-regions. For the next iteration, divide each of the 4 sub-regions into 4 further sub-regions; the basic U-shape is constructed with sides half the previous length, and in addition the current figure is copied into each of the new 4 regions as follows: the top left-hand region contains a copy of the current figure (constructed with sides of half their previous length), the top right-hand region is a reflection (in the vertical axis) of the left-hand region, the bottom left-hand region has the current figure rotated 90 degrees clockwise while the bottom right-hand figure uses a 90 degree anti-clockwise rotation.

The first one, two, three and four iterations of a Hilbert curve are shown in the following figure (Fig 1).

1.2 Your tasks this week

This work is to be done in your allocated groups, and forms the foundation steps of the group project.

- Implement a Hilbert curve (`Hilbert(n)`) in MATLAB in two dimensions, as a function of n (the iteration level).
- Compute the length of the curve as a function of n.
- Compute the total number of turning points, including the start and end points, as a function of n.
- Create a distance matrix for all the above turning points, as a function of n.
- Construct the 4×4 distance matrix of Hilbert(1) and try to **reconstruct** the curve using geometric arguments.

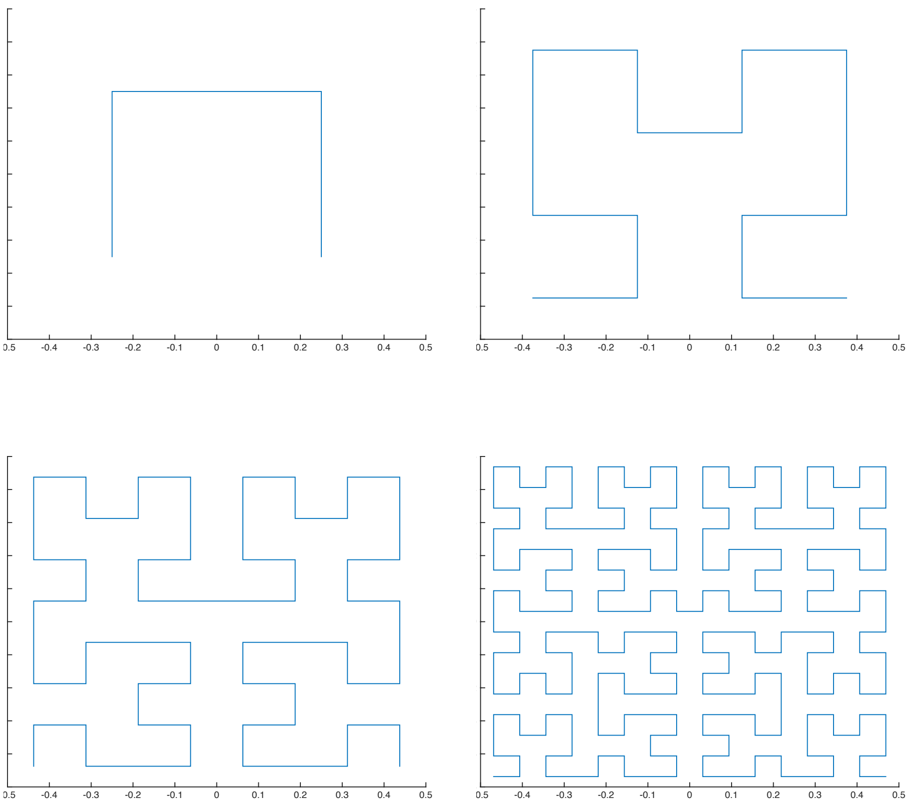

Figure 1: Hilbert Curves, with 1, 2, 3 and 4 iterations