Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Dipl.-Inf. G. Buse, M.Sc. K. Rahnema

SS 2012
Sheet 12
11.07.2012

Algorithms of Scientific Computing

Arithmetization of Space-Filling Curves

If the number $t \in [0, 1[$ is given with the basis four, i.e.

$$t = 0_4.q_1q_2q_3q_4\ldots,$$

then the mapping $h(t)$ of the Hilbert curve can be written as

$$h(0_4.q_1q_2q_3q_4\ldots) = H_{q_1} \circ H_{q_2} \circ H_{q_3} \circ H_{q_4} \circ \cdots \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

with the operators

$$H_0 := \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad H_1 := \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix}$$

$$H_2 := \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad H_3 := \begin{pmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}.$$

Excercise 1: Calculation of h for (in)finite fractions

Calulate the values $h\left(\frac{1}{8}\right)$ and $h\left(\frac{1}{3}\right)$.

Excercise 2: Arithmetization of the Peano Curve

Derive an arithmetization of the Peano curve (see figure 1a), similar to the arithmetization of the Hilbert curve. Hence, we are looking for a representation

$$p(0.q_1q_2q_3q_4\ldots) = P_{q_1} \circ P_{q_2} \circ P_{q_3} \circ P_{q_4} \circ \cdots \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

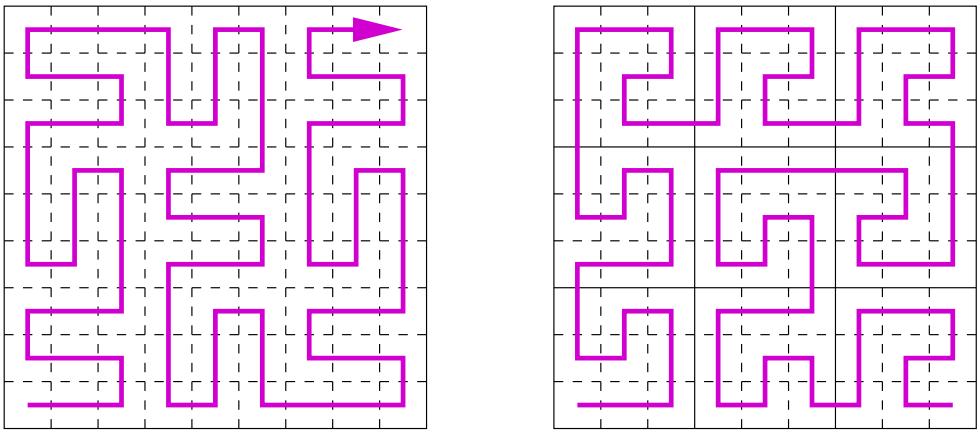
constructed on an appropriate basis and with appropriate operators P_0, P_1, ...

Develop an algorithm that computes the Peano function $p(t)$.

Use the Peano function to plot the approximating polygon of the Peano curve (with Python).

Additional Excercise:

Do the same for the meander-style Peano curve.



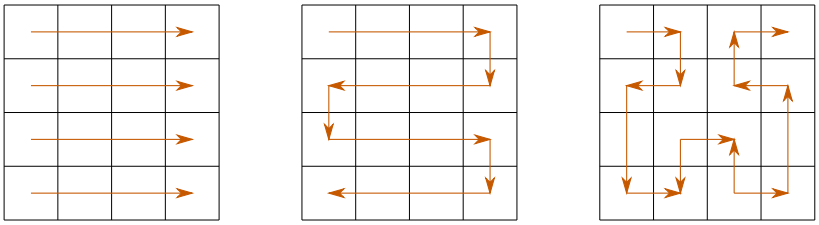
(a) Peano Curve (b) Meander-style Peano Curve

Abbildung 1: Two space-filling curves of the Peano type

Excercise 3: Cache Efficiency

One field of application for space-filling curves is the cache efficient traversal of grids. Due to the increasing gap between processor and memory speed this is an more and more important issue. In this last excercise we will examine, if we can obtain a better efficiency by exploiting the spatial locality of space-filling curves.

In the figures below we see three traversal schemes over a four-by-four grid. Assume that we traverse the grid cells in the given order and we always need all four adjacent vertices for one cell to process it. Assume further that our cache can store up to seven vertices (cells do not contain any data, thus they don't need to be stored). If the cache is full and another vertex is loaded, it replaces the least recently used vertex from the cache. If there are more than one vertices that can be replaced, the upper or more left one is preferred.



Try manually, how many cache-misses occur during the three grid traversals.

Discuss whether/how the results change if you change the size of the cache, and/or grid.