Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
M.Sc. A. Mo-Hellenbrand, M.Sc. J.-M. Gallard

SS 2017
Worksheet 11
19.07.2017

Algorithms of Scientific Computing

Arithmetization of Space-Filling Curves and Cache Efficiency

Exercise 1: Arithmetization of the Hilbert Curve

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

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

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

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

b) In the lecture you learned that there is no continuous bijection between $[0,1]$ and $[0,1]^2$. So there have to be points x, y with $x \neq y$ so that $h(x) = h(y)$. How many points are mapped onto $\left(\frac{1}{2}, \frac{1}{2}\right)^T$? How many points are mapped onto $(0,0)^T$?

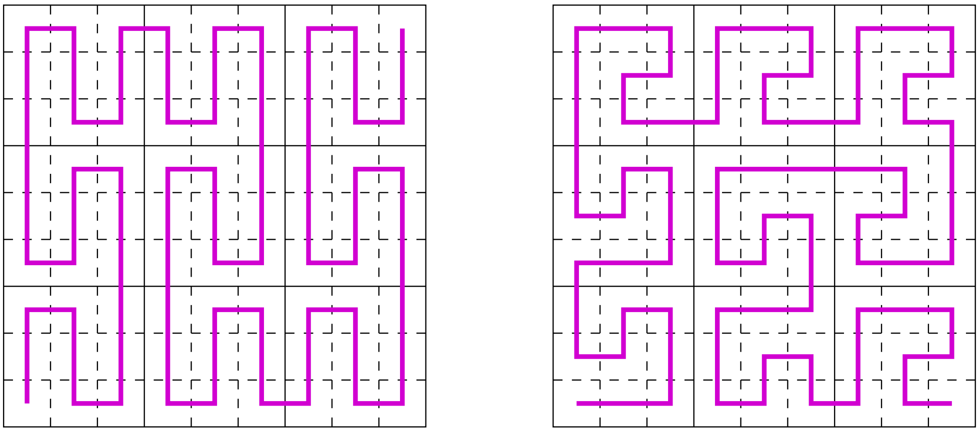
Exercise 2: Arithmetization of the Peano Curve

Derive an arithmetization of the Peano curve (see Figure 1a), analogous to the arithmetization of the Hilbert curve. So given a number $t = 0_9.q_1q_2q_3q_4\ldots$, we are looking for a representation

$$p(0_9.q_1q_2q_3q_4\ldots) = P_{q_1} \circ P_{q_2} \circ P_{q_3} \circ P_{q_4} \circ \cdots \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

with appropriate operators $P_0, P_1, \ldots$.

Develop an algorithm that computes the Peano function $p(t)$. A code skeleton is attached. Use the Peano function to plot the approximating polygon of the Peano curve.



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

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

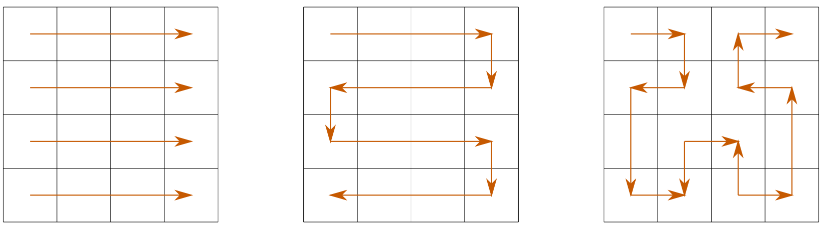
Additional Exercise:

Do the same for the meander-style Peano curve.

Exercise 3: Cache Efficiency

One field of application for space-filling curves is the cache efficient traversal of grids. Due to the widening gap between processor and memory speed this is an more and more important issue. In this last exercise 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. They are loaded in a Z pattern (top left first, then top right, bottom left and bottom right last). 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.



(a) row-wise (b) zig-zag (c) Hilbert

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