Dr. Tobias Neckel
Dr. Florian Rupp

Dynamical Systems & Scientific Computing: Homework Assignments

4.1 [⏲] Fast Poisson Solver

Using the discrete sine transform (DST) for the Poisson problem

$$\Delta u(x) = f(x), \qquad x \in \Omega := [0,1]^2 \tag{1}$$
$$u(x) = 0, \qquad x \in \Gamma := \partial\Omega \tag{2}$$

as described in exercise 01, a fast alternative solver for the resulting linear system of equations can be constructed.

a) 1D Problem

In the one-dimensional case, the discretisation of Eq. (1) with finite differences results in the following linear system of equations

$$u_{n+1} - 2u_n + u_{n-1} = f_n, \quad n = 1, \ldots, N-1 \ , \tag{3}$$

where $u_1, \ldots, u_{N-1}$ represent the unknowns and f_n is defined as $f_n := h^2 \cdot f(x_n)$ using the mesh size h. Note that due to the Dirichlet zero boundary conditions $u_0 = u_N = 0$, no equations for index 0 and N exist and no contribution in the discretised right hand side f_1 and f_{N-1} appear.
Now, we use the (inverse) discrete sine transform (DST) of u and f

$$u_n = 2 \sum_{k=1}^{N-1} U_k \sin \frac{\pi n k}{N}, \qquad f_n = 2 \sum_{k=1}^{N-1} F_k \sin \frac{\pi n k}{N}$$

and insert these terms into the system of equations (3).

i) Show that the DST coefficients U_k depend on the F_k in the following way

$$U_k = \frac{F_k}{2 \cos \frac{\pi k}{N} - 2} \qquad \text{for} \quad k = 1, \ldots, N-1 \ . \tag{4}$$

Hence, the U_k can be retrieved directly from the F_k, without solving a linear system of equations!

ii) Formulate an algorithm which solves the system of equations (3) efficiently by using the dependency (4) and the fast sine transform(s).

b) 2D Problem

In the two-dimensional case, the FD discretisation of Eq. (1) results in the following linear system of equations

$$u_{n,m+1} + u_{n+1,m} - 4u_{n,m} + u_{n-1,m} + u_{n,m-1} = f_{n,m}, \quad n,m = 1, \ldots, N-1 \ , \tag{5}$$

again with homogeneous Dirichlet boundary conditions and the scaling $f_{n,m} := h^2 \cdot f(x_n, y_m)$ using the mesh size $h_x = h_y = h$.
Similar to 1D, we use the 2D (inverse) discrete sine transform (DST) of u and f

$$u_{n,m} = 2 \sum_{k=1}^{N-1} \sum_{l=1}^{N-1} U_{k,l} \sin \frac{\pi n k}{N} \sin \frac{\pi m l}{N}, \qquad f_{n,m} = 2 \sum_{k=1}^{N-1} \sum_{l=1}^{N-1} F_{k,l} \sin \frac{\pi n k}{N} \sin \frac{\pi m l}{N} \tag{6}$$

and insert these terms into the linear system of equations (5).

i) Show that the DST coefficients $U_{k,l}$ depend on the $F_{k,l}$ in the following way

$$U_{k,l} = \frac{F_{k,l}}{2\cos\frac{\pi k}{N} + 2\cos\frac{\pi l}{N} - 4} \qquad \text{for} \quad k,l = 1,\ldots,N-1\ . \tag{7}$$

Hence, the $U_{k,l}$ can be retrieved directly from the $F_{k,l}$, without solving a linear system of equations!

ii) Formulate a 2D algorithm which solves the system of equations (5) efficiently by using the dependency (7) and the fast sine transform(s).

4.2 [☼] The Hilbert Curve

The type of grammars which has been created in Excercise 1, can be used to derive an algorithm, which defines the travel direction of the curve in a global coordinate system.

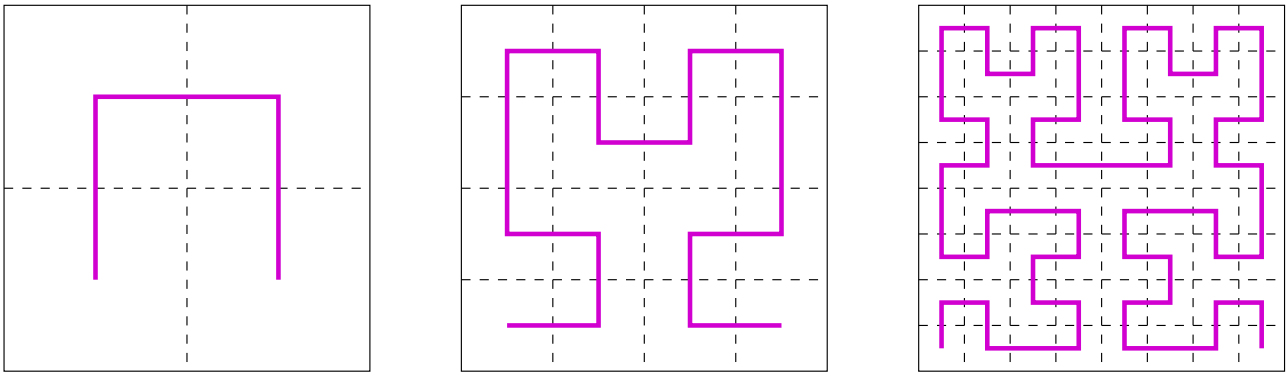

Abbildung 1: Construction of the Hilbert Curve.

In this Excercise we will derive a "real" turtle graphics algorithm for the Hilbert Curve, that only uses the following commands:

- Go one step ahead.
- Turn the travel direction by 90 ° to the right.
- Turn the travel direction by 90 ° to the left.

a) Try to find an algorithm for which the turtle turns at most once after doing a step (so it shouldn't turn more than once by 90 ° at the same spot).

b) Implement the grammar in a matlab program. Assume that the 2D domain is $[0,1]^2$ and use the depth p of the cell tree as input parameter to stop the recursion. Make sure to visualise your results to cross-check the correct construction of the discrete iterates of the curve.

Hint: You can for example consider where the curve enters and exits a sub-square. Try to think like the turtle: The next sub-square is always in front of you...

Classification: ☼ easy, ◷ easy with longer calculations, ☆ a little bit difficult, ⬚ challenging.