Construct and reconstructing orbits from symbolic dynamics

The goal of this notebook is to put in evidence the algebraic operations of the **linear encoding of the cat map.** In the end we map between:
- the discrete set of coordinates of the phase space $x_{t,n}$ where t, n represent respectively the time and space evolution of the chaotic flow
- $m_{t,n}$ the symbolic encoding of the flow

The equation relating the two is given by **Eq.(3)** of our research article found on the arXiv https://arxiv.org/pdf/1912.02940.pdf

In particular, we look derive the finite matrix that maps from space of $x_{t,n}$ to space of symbols $m_{t,n}$ called A

Deriving m’s from x’s

Consider the system with periodicity in time $T = 5$ and periodicity in space $N_p = 3$

In[ ]:=
```
{T, Np} = {5, 3};
X = Array[Subscript[x, #1, #2] &, {T, Np}];
X // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \\ x_{4,1} & x_{4,2} & x_{4,3} \\ x_{5,1} & x_{5,2} & x_{5,3} \end{pmatrix}$$

Define the map A which maps from x’s to m’s

For that, we must derive the reformatted column vector of x called xbis, where all $x_{t,n}$ are re-organized as a column vector

In[•]:=
```
Xbis = ArrayReshape[X, {Np * T, 1}];
Xbis // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \\ x_{4,1} \\ x_{4,2} \\ x_{4,3} \\ x_{5,1} \\ x_{5,2} \\ x_{5,3} \end{pmatrix}$$

Then A is:

In[•]:=
```
Amap[T, Np, s] // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix}
s & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
-1 & s & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
-1 & -1 & s & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & s & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & -1 & s & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & -1 & s & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & s & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & s & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & s & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & s & -1 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & s & -1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & s & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & s & -1 & -1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & s & -1 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & s
\end{pmatrix}$$

And the resulting mbis vector, which we need to reformat to actually get the symbols $m_{t,n}$

mbis = A Xbis

mbis $\to m$

In[•]:=
```
mbis = Amap[T, Np, s].Xbis;
mbis // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} s\,x_{1,1}-x_{1,2}-x_{1,3}-x_{2,1}-x_{5,1} \\ -x_{1,1}+s\,x_{1,2}-x_{1,3}-x_{2,2}-x_{5,2} \\ -x_{1,1}-x_{1,2}+s\,x_{1,3}-x_{2,3}-x_{5,3} \\ -x_{1,1}+s\,x_{2,1}-x_{2,2}-x_{2,3}-x_{3,1} \\ -x_{1,2}-x_{2,1}+s\,x_{2,2}-x_{2,3}-x_{3,2} \\ -x_{1,3}-x_{2,1}-x_{2,2}+s\,x_{2,3}-x_{3,3} \\ -x_{2,1}+s\,x_{3,1}-x_{3,2}-x_{3,3}-x_{4,1} \\ -x_{2,2}-x_{3,1}+s\,x_{3,2}-x_{3,3}-x_{4,2} \\ -x_{2,3}-x_{3,1}-x_{3,2}+s\,x_{3,3}-x_{4,3} \\ -x_{3,1}+s\,x_{4,1}-x_{4,2}-x_{4,3}-x_{5,1} \\ -x_{3,2}-x_{4,1}+s\,x_{4,2}-x_{4,3}-x_{5,2} \\ -x_{3,3}-x_{4,1}-x_{4,2}+s\,x_{4,3}-x_{5,3} \\ -x_{1,1}-x_{4,1}+s\,x_{5,1}-x_{5,2}-x_{5,3} \\ -x_{1,2}-x_{4,2}-x_{5,1}+s\,x_{5,2}-x_{5,3} \\ -x_{1,3}-x_{4,3}-x_{5,1}-x_{5,2}+s\,x_{5,3} \end{pmatrix}$$

In[•]:=
```
m = ArrayReshape[mbis, {T, Np}];
m // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} s\,x_{1,1}-x_{1,2}-x_{1,3}-x_{2,1}-x_{5,1} & -x_{1,1}+s\,x_{1,2}-x_{1,3}-x_{2,2}-x_{5,2} & -x_{1,1}-x_{1,2}+s\,x_{1,3}-x_{2,3}-x_{5,3} \\ -x_{1,1}+s\,x_{2,1}-x_{2,2}-x_{2,3}-x_{3,1} & -x_{1,2}-x_{2,1}+s\,x_{2,2}-x_{2,3}-x_{3,2} & -x_{1,3}-x_{2,1}-x_{2,2}+s\,x_{2,3}-x_{3,3} \\ -x_{2,1}+s\,x_{3,1}-x_{3,2}-x_{3,3}-x_{4,1} & -x_{2,2}-x_{3,1}+s\,x_{3,2}-x_{3,3}-x_{4,2} & -x_{2,3}-x_{3,1}-x_{3,2}+s\,x_{3,3}-x_{4,3} \\ -x_{3,1}+s\,x_{4,1}-x_{4,2}-x_{4,3}-x_{5,1} & -x_{3,2}-x_{4,1}+s\,x_{4,2}-x_{4,3}-x_{5,2} & -x_{3,3}-x_{4,1}-x_{4,2}+s\,x_{4,3}-x_{5,3} \\ -x_{1,1}-x_{4,1}+s\,x_{5,1}-x_{5,2}-x_{5,3} & -x_{1,2}-x_{4,2}-x_{5,1}+s\,x_{5,2}-x_{5,3} & -x_{1,3}-x_{4,3}-x_{5,1}-x_{5,2}+s\,x_{5,3} \end{pmatrix}$$

Defining Functions

Reconstruct orbits

Our goal is mostly to **reconstruct orbits** in phase space from the symbolic dynamics. To that end, we follow the converse of the process written above:

m → mbis → xbis using xbis = A^{-1} mbis. Then xbis → x

Generating different sequence of symbols

One key feature of this project is to see how symbolic dynamics m_{tn} that share common sequences produce trajectories in phase that overlap. The main example studied is the one of **figure 6 on the article**, where the core (blue region) of two symbolic dynamics shadow each other. We write multiple user-defined functions, each with a precise goal (generating symbols, coloring of the matrices of symbols and plotting) in order to draw and plot the respective trajectories of these sequences, in phase space.

Common core shared: diffoutside

Inputs two different matrices of symbols, a range of integers, the position and the size of the share block
Outputs the two matrices with a shared rectangular-shaped sequence of symbols

Swapping 1 block of symbols: swapping1block

Inputs 1 matrix of symbols, a range of integers, the positions and the size of the swaps
Outputs two identical matrices except for two rectangular-shaped sequences of symbols swapped

Swapping 2 blocks of symbols: swapping2blocks

Inputs 1 matrix of symbols, a range of integers, the positions and the size of the swaps and the positions of the bordering shared blocks
Outputs two identical matrices except for two rectangular-shaped sequences of symbols swapped, surrounded by a shared thick border of symbols

Swapping 3 blocks symbols between 3 sequences: swapping3blocks

Does the same as swapping2blocks except the output is: three identical matrices with three rectangular-shaped sequences of symbols swapped, surrounded by a shared thick border of symbols

Swapping 2 diamond-shaped blocks of symbols between 2 sequences: swapping2diamonds

Does the same as swapping2blocks except the shape of the swapped symbols and thick borders are diamond.

Swapping 3 diamond-shaped cores between 3 sequences: swapping3diamonds

Does the same as swapping3blocks except the shape of the swapped symbols and thick borders are diamond.

Coloring 1 block of symbols: coloring1block and coloring1blockwithfont

Inputs matrix of symbols and position and size of a block that needs to be colored
Outputs same matrix with a block of colored symbols. The second function does the same with a special font for those same symbols.

Coloring 1 inner and 1 outer blocks of symbols: coloring2blocks

Inputs matrix of symbols and position and size of a block and a thick border that need to be colored
Outputs same matrix with colored symbols

Coloring 1 inner and 1 outer diamond-shaped blocks of symbols: coloring2diamonds

Does the same as coloring2blocks, except for diamond-shaped blocks symbols rather than rectangular

Replacing symbols with color code: replacingsymbolswithsquares

Inputs a matrix of integers.

Outputs a matrix of colors, where each color corresponds to an integer

Drawing borders around color-coded symbols: drawingsquareborders

Inputs a matrix of colors and draws a rectangular contour around a region

Drawing diamond-shaped borders around color-coded symbols: drawingdiamondborders

Does the same drawingsquareborders except for diamond-shaped borders

Plotting 2 sequences in (q, p) phase space with zoom-in: plotting2

Inputs two trajectories in phase space (q_1, p_1) and (q_2, p_2)

Outputs the discrete display of these trajectories in two separate plots: one ranging from [0, 1] (the entire phase space) and the other zoomed-in on a particular region of interest.

The two set of discrete points are plotted with green / red disks of different size in order to visualize how overlapping the trajectories are in phase space.

Plotting 3 sequences in (q,p) phase space with zoom-in: plotting3

Inputs two trajectories in phase space (q_1, p_1), (q_2, p_2) and (q_3, p_3)

Does the same as plotting2 except with 3 sets of points and 3 different different disks of color green, red and blue.

Plotting 2 sequences in (q,p) phase space with zoom-in with special coloring for shared sequences of symbols: specialplotting2

Does the same as plotting2 except the coordinates $(q_{\text{tn}}, p_{\text{tn}})$ corresponding to the symbolic dynamics that overshadow one another are plotted with a different coloring

Other functions