The Dynamics of Complex Systems

Phase Space Reconstruction
Recurrence Quantification Analysis

Fred Hasselman
f.hasselman@bsi.ru.nl

Maarten Wijnants
m.wijnants@pwo.ru.nl

Behavioural Science Institute
Radboud University Nijmegen

Test a hypothesis by creating surrogate data and bootstrapping your favourite nonlinear measure

It is possible to create more sophisticated hypotheses by generating surrogate data based on the spectrum / distribution of the observed data.

H_0: Signal is generated by	Surrogate type	Method
Independent random numbers	I. Random temporal structure with same distribution as observed data	Randomise samples without replacement
Gaussian linear stochastic process	II. Random spectral phases	Multiply Fourier transform of the sample by random phases and convert back to time domain
Rescaled Gaussian linear stochastic process	III. Rescaled to a Gaussian distribution, phase randomised and rescaled to the empirical distribution	- Amplitude Adjusted Fourier Transform (AAFT) (biased!) - Itratively Refined Surrogates
Correlated linear stochastic process with time dependent mean and variance	IV. Preserve correlation structure and nonstationarity of mean and variance	General constrained randomisation by a cost function minimised by simulated annealing
Same dynamical system at different initial conditions	V. Exact copy of phase space, different initial conditions	Twin surrogates: Generate from recurrence matrix

Ok!
Now I know it is likely my time series data represent the dynamics of a complex system…

scale-free / fractal
highly correlated / interdependent
nonlinear / maybe chaotic
result of multiplicative interactions

now what?
(help!)

Takens' (1981) Embedding Theorem tells us that a (strange) attractor can be recovered ("reconstructed") from observations of a single component process of a complex interaction-dominant system.

Takens, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand and L.-S. Young (Eds.) Dynamical Systems and Turbulence. *Lecture Notes in Mathematics vol. 898*, 366–381, Springer-Verlag.

How to study interaction-dominant systems

As you know in a **coupled system** the time evolution of one variable depends on other variables of the system. This implies that one variable contains information about the other variables (of course depending upon the strength of coupling and maybe the type of interaction)

So given the Lorenz system …

$$dX/dt = \delta \cdot (Y - X)$$

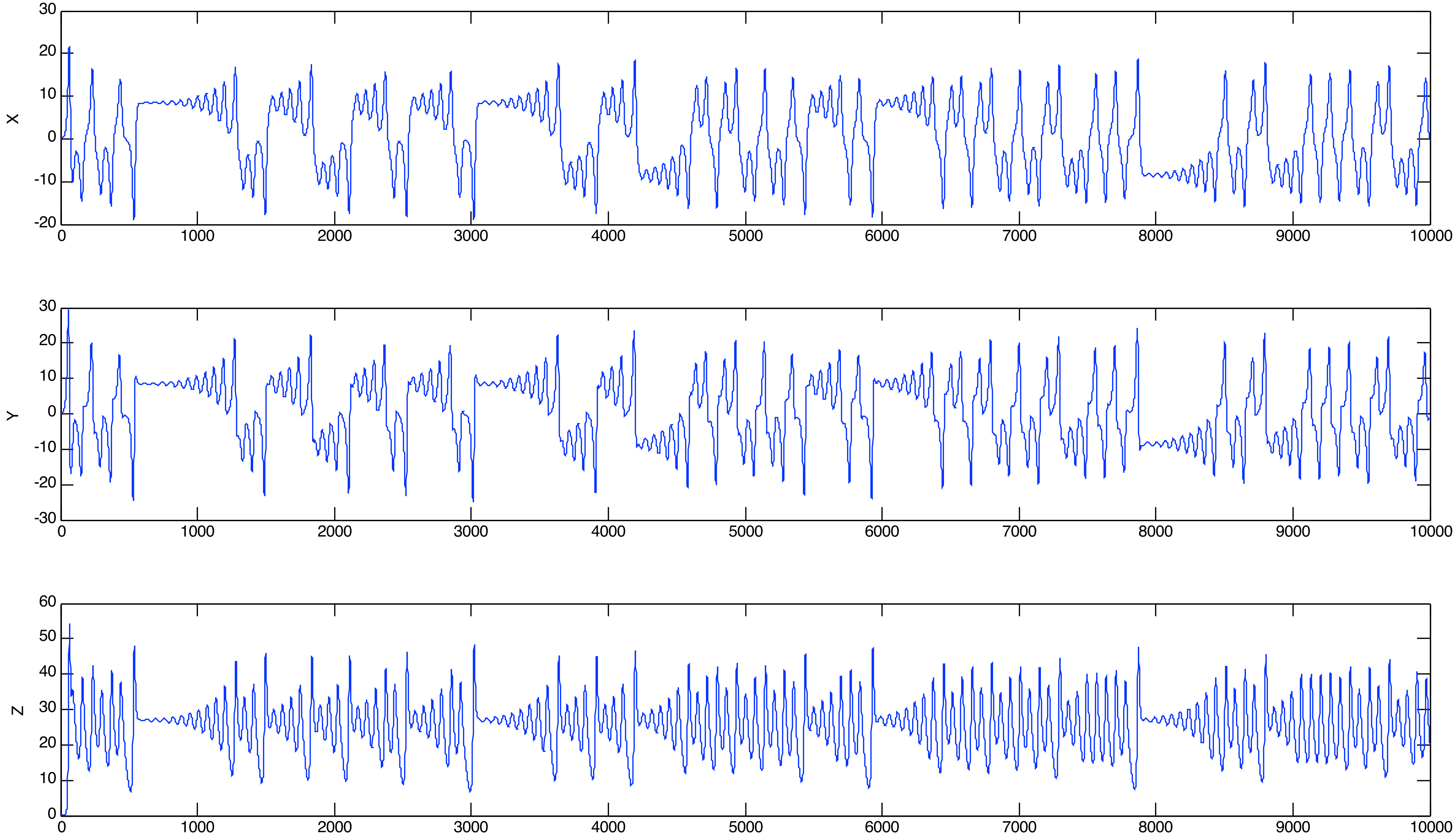
$$dY/dt = r \cdot X - Y - X \cdot Z$$

$$dZ/dt = X \cdot Y - b \cdot Z$$

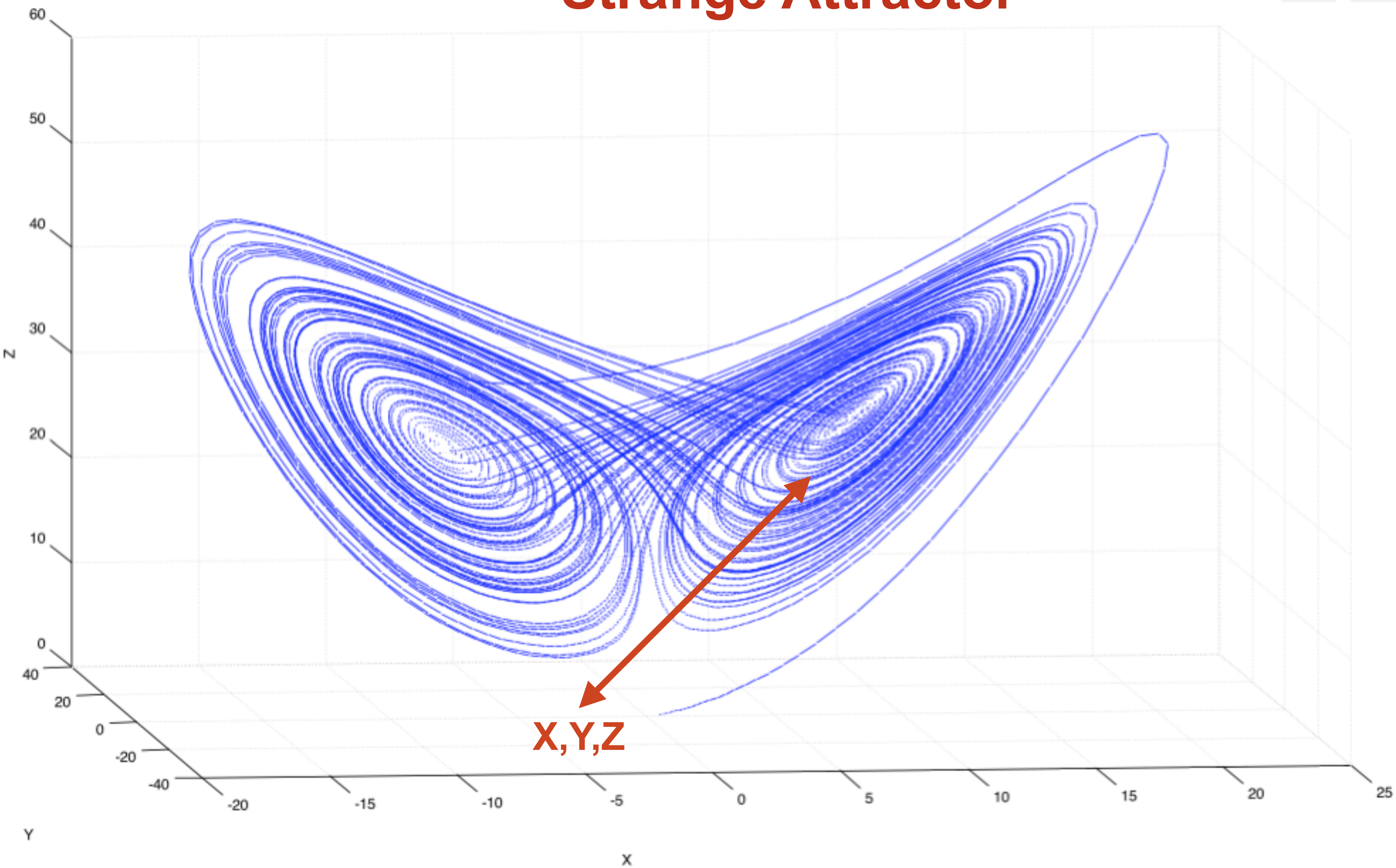
Takens' theorem suggests that we should be able to reconstruct the higly chaotic "butterfly" attractor by just using X(t) [or Y(t) or Z(t)] …

Takens, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand and L.-S. Young (Eds.) Dynamical Systems and Turbulence. *Lecture Notes in Mathematics vol. 898*, 366–381, Springer-Verlag.

Lorenz system – Time series of X, Y and Z

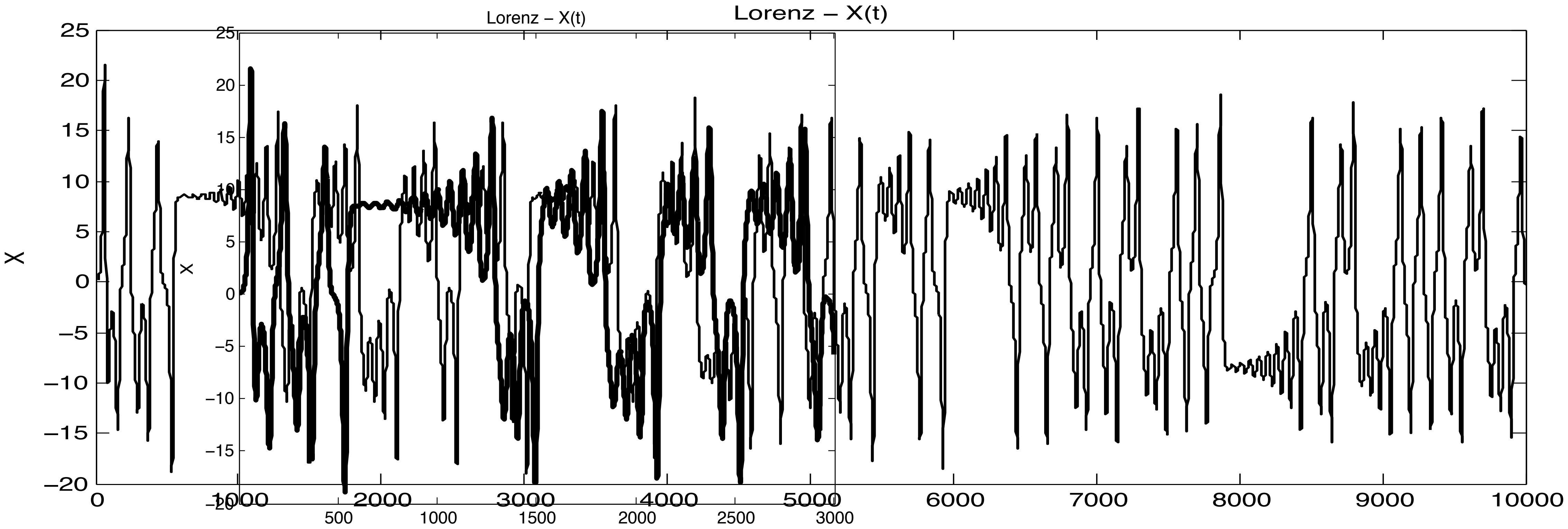


Lorenz system – X,Y,Z State space
Strange Attractor



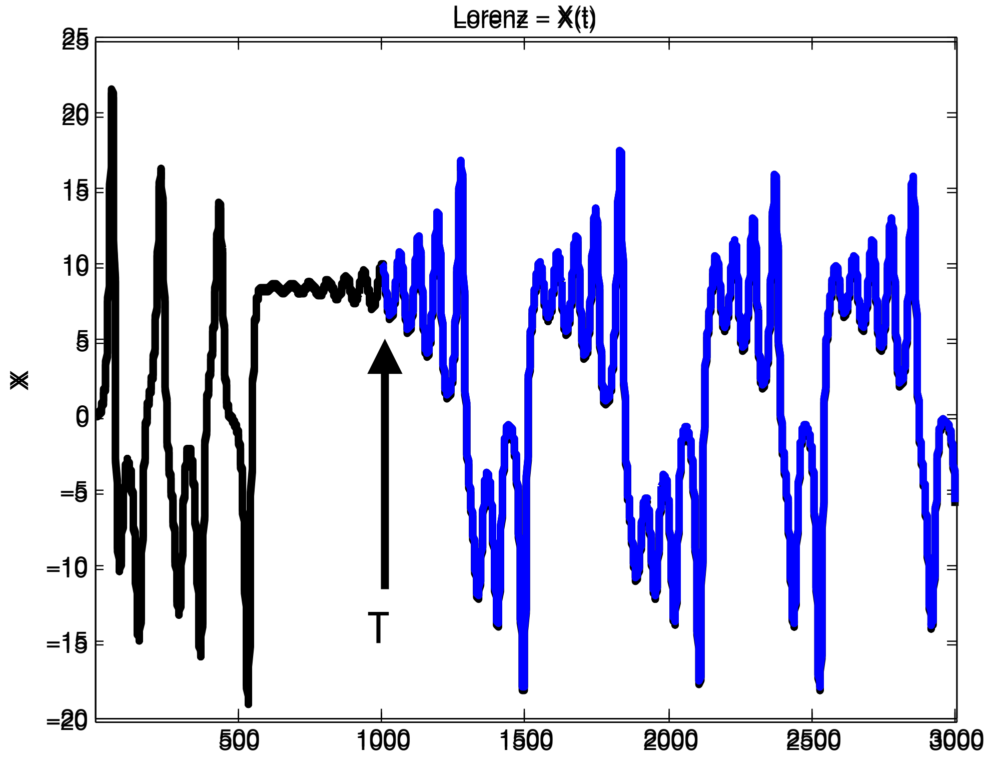
Creating surrogate dimensions using the method of delays

X (t)



Creating surrogate dimensions using the method of delays

X (t)

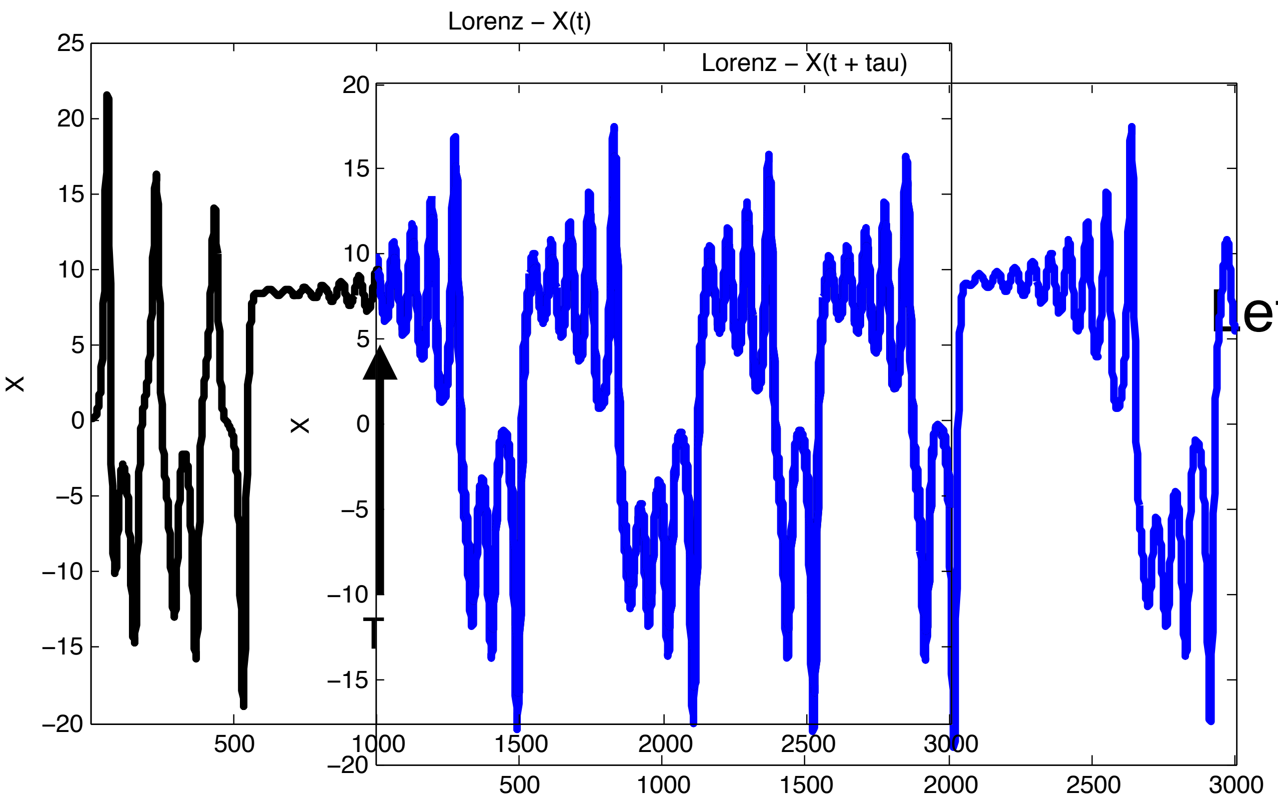


Let's take our embedding delay or lag to be:

T = 1000

Creating surrogate dimensions using the method of delays

X (t)



Let's take our embedding delay or lag to be:

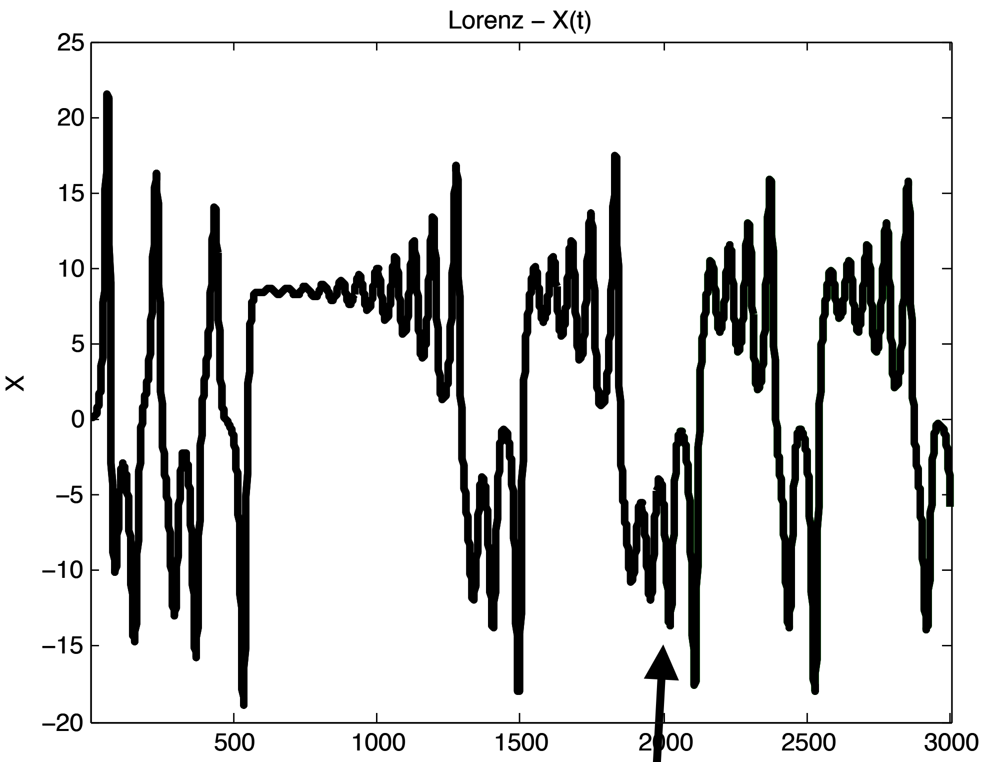
T = 1000

X (t + τ)

Data point 1 + T [X(t) = 1001] becomes data point 1 for this dimension

Creating surrogate dimensions using the method of delays

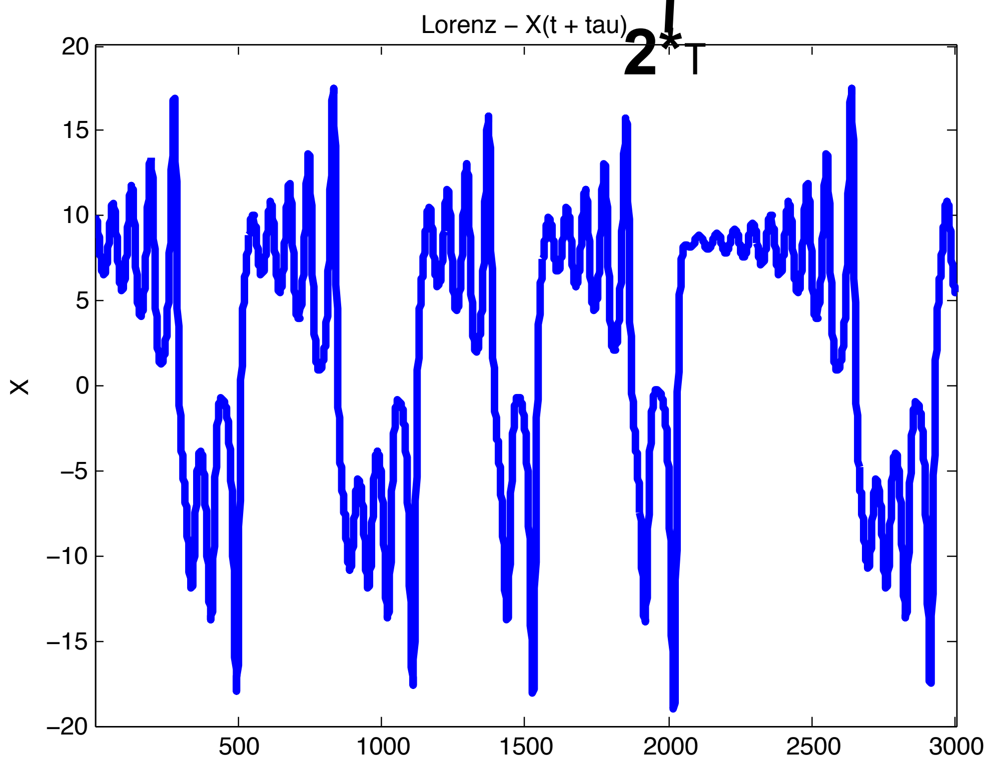
X (t)



Let's take our embedding delay or lag to be:

T = 1000

X (t + T)



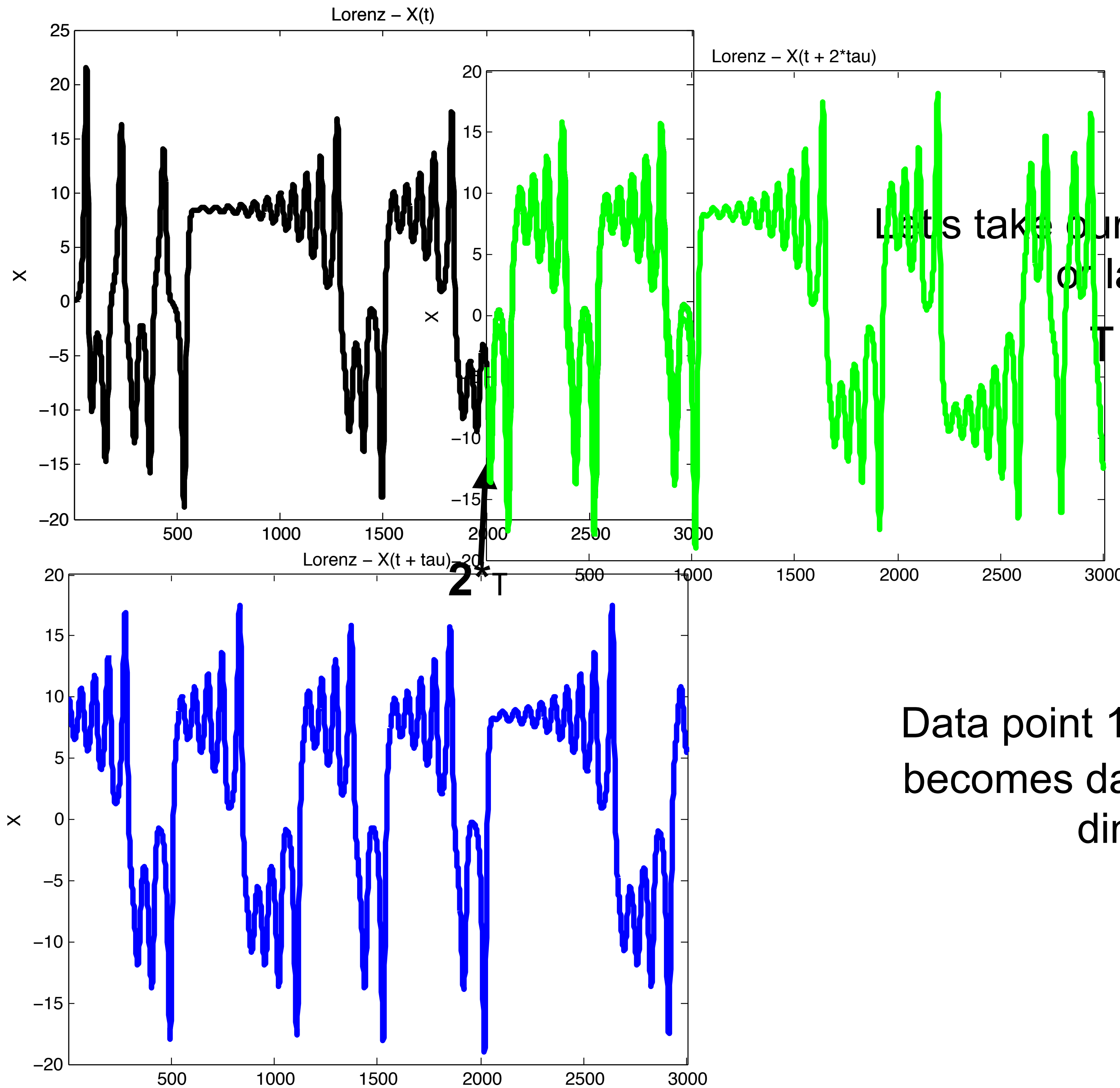
Data point 1 + T [X(t) = 1001] becomes data point 1 for this dimension

Creating surrogate dimensions using the method of delays

X (t)

X (t + τ)

X (t + 2*τ)



Let's take our embedding delay or lag to be:

τ = 1000

Data point 1 + τ [X(t) = 1001] becomes data point 1 for this dimension

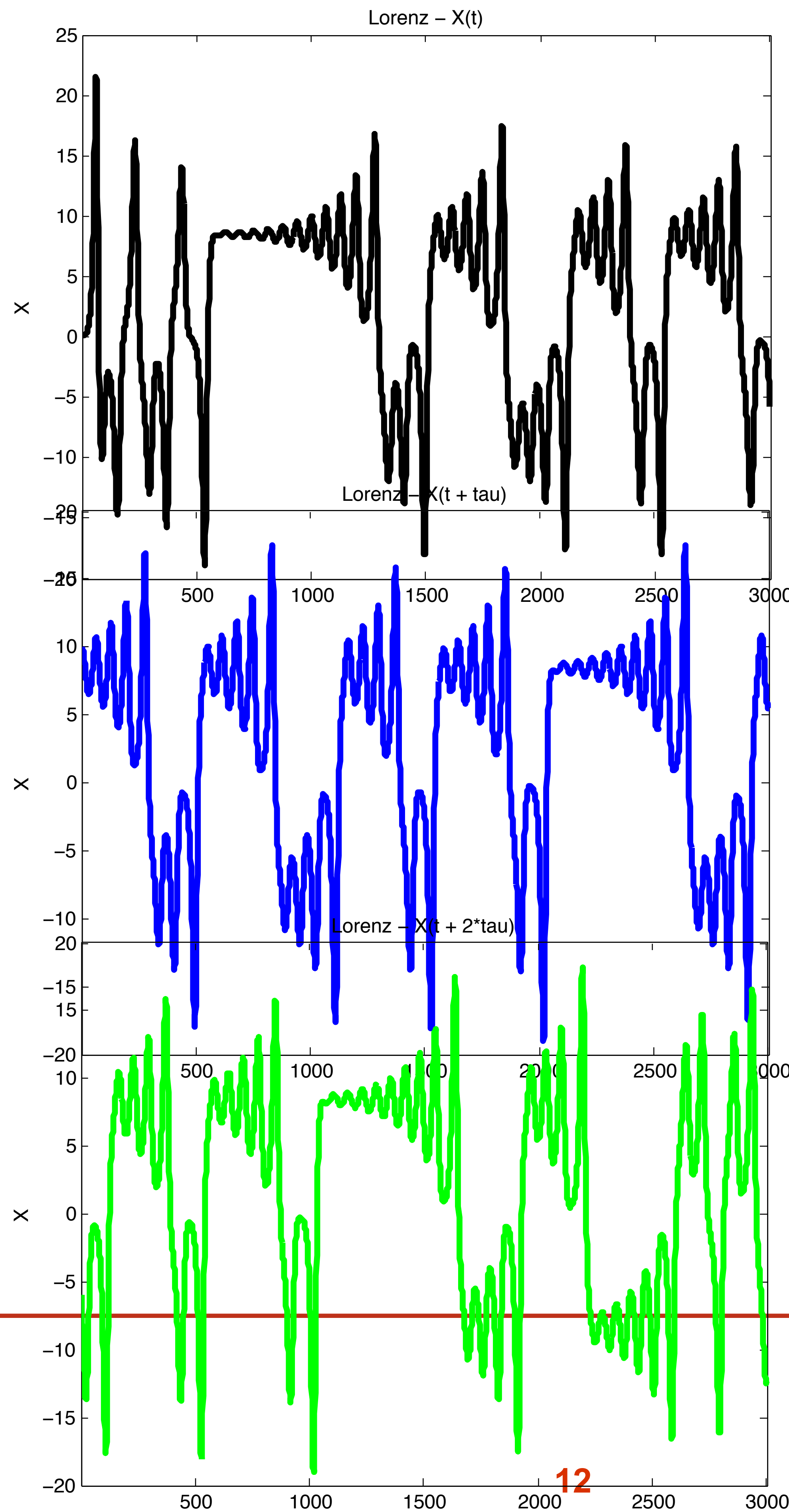
Data point 1 + 2*τ [X(t) = 2001] becomes data point 1 for this dimension

Creating surrogate dimensions using the method of delays

X (t)

X (t + τ)

X (t + 2τ)



The embedding lag reflects the point in the time series at which we are getting **new information** about the system…

In theory any lag can be used, everything is interacting...

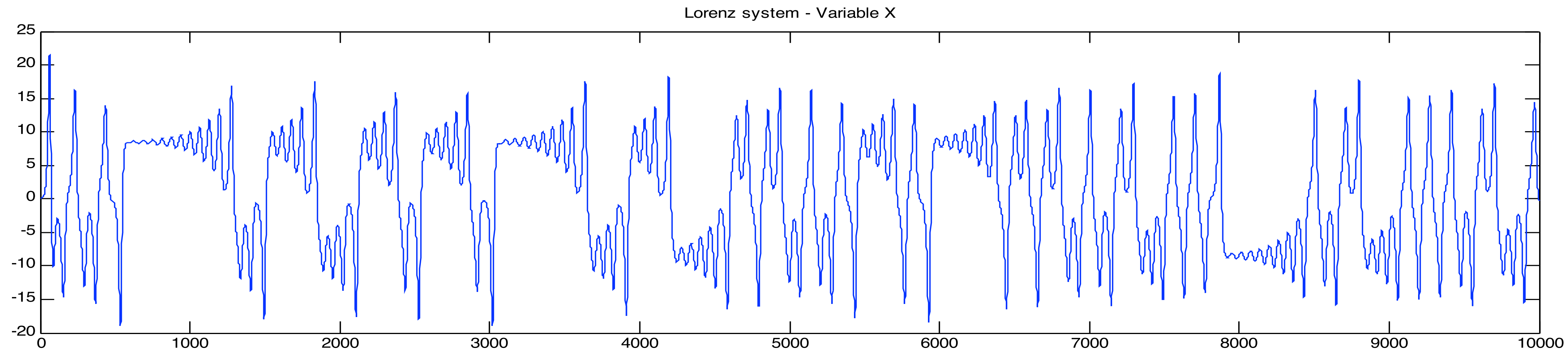
We are looking for the lag which is optimal, gives us maximal new information about the temporal structure in the data…

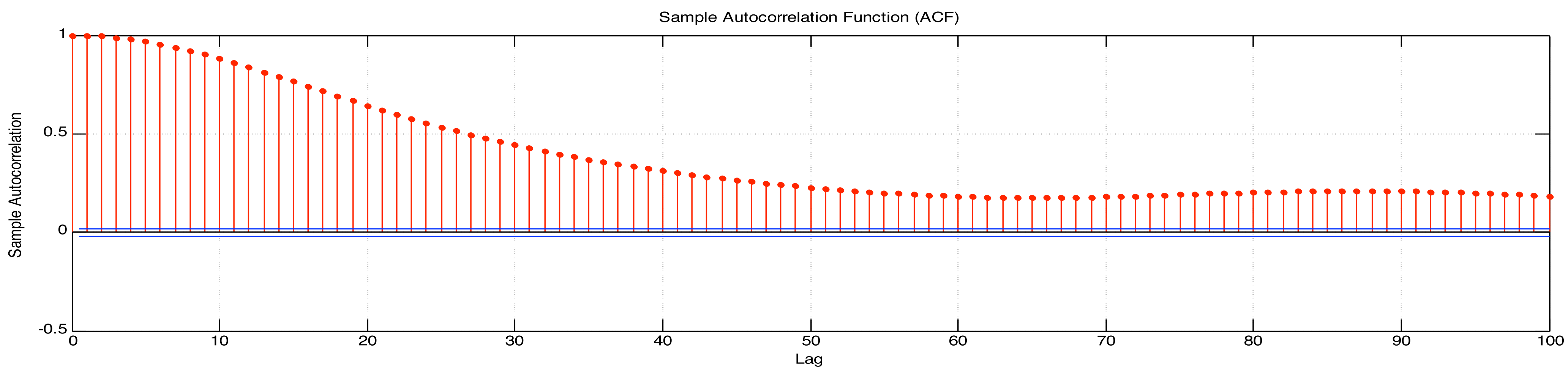
Intuitively:
Where the autocorrelation is zero

We are creating a return plot to examine the systems' state space!

How to determine embedding lag?

- We saw that the autocorrelation function is not very helpful when you are dealing with long range correlations in the data.



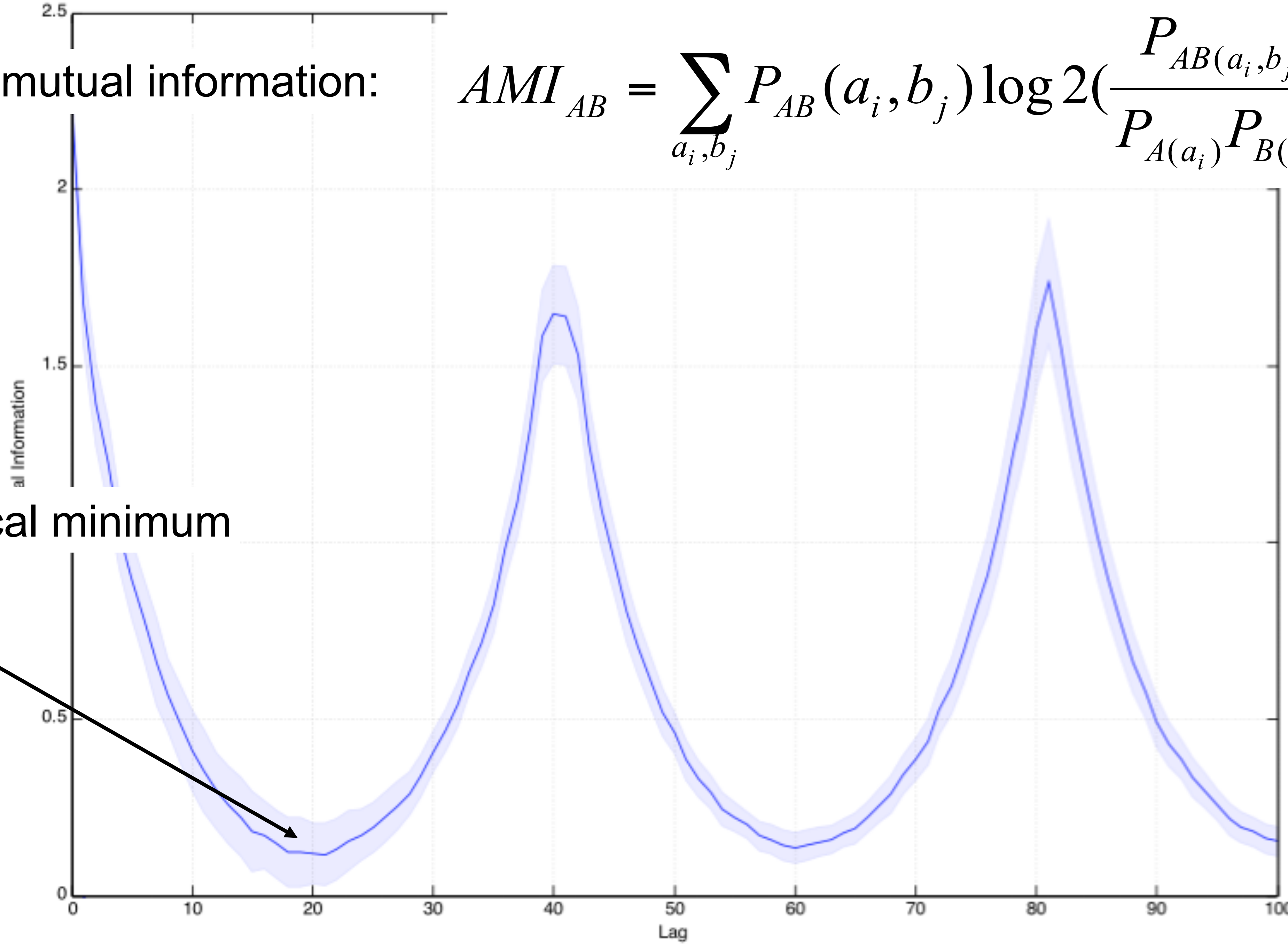


Lorenz system – Determine embedding lag

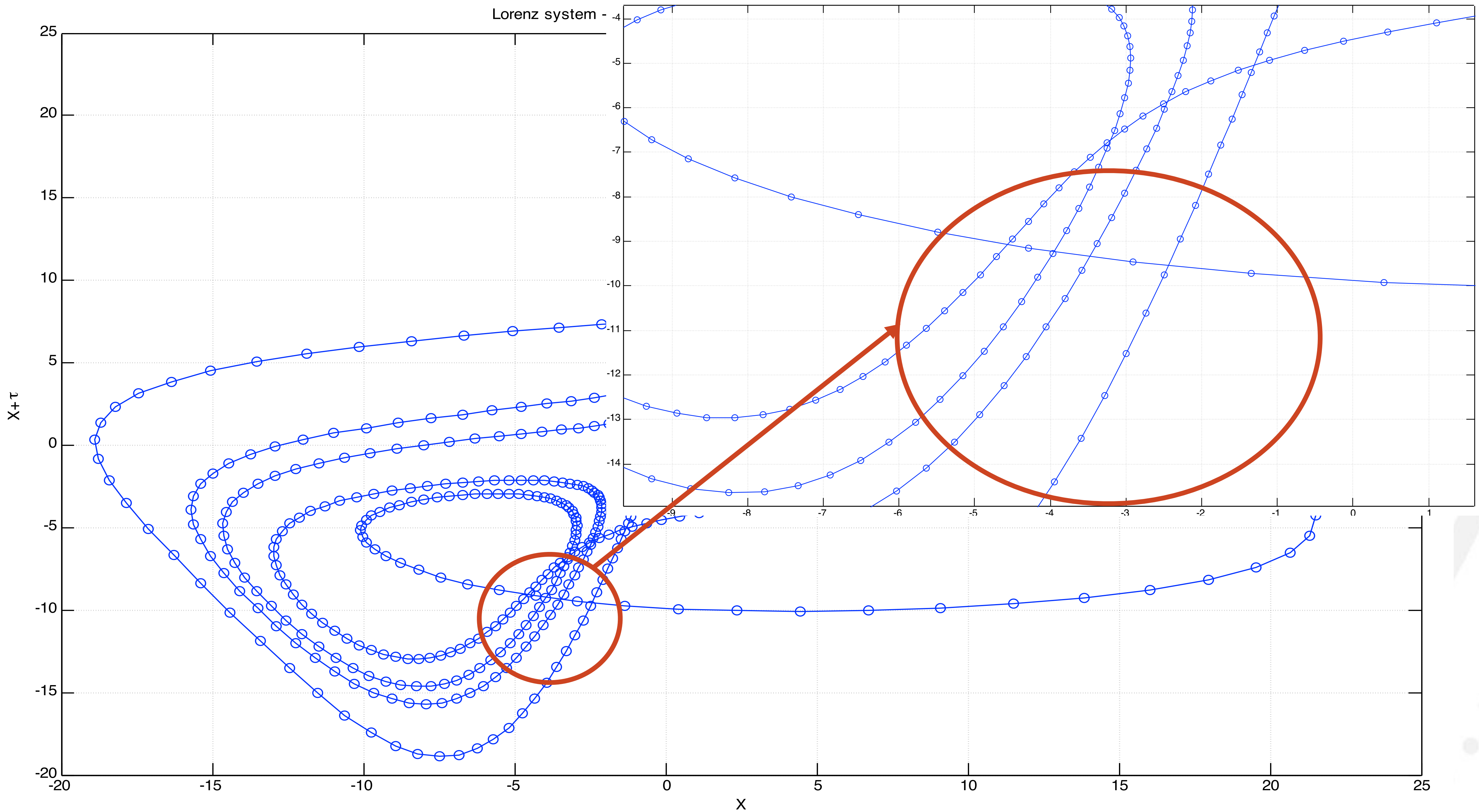
Average mutual information:

$$AMI_{AB} = \sum_{a_i, b_j} P_{AB}(a_i, b_j) \log 2\left(\frac{P_{AB(a_i, b_j)}}{P_{A(a_i)} P_{B(b_j)}}\right)$$

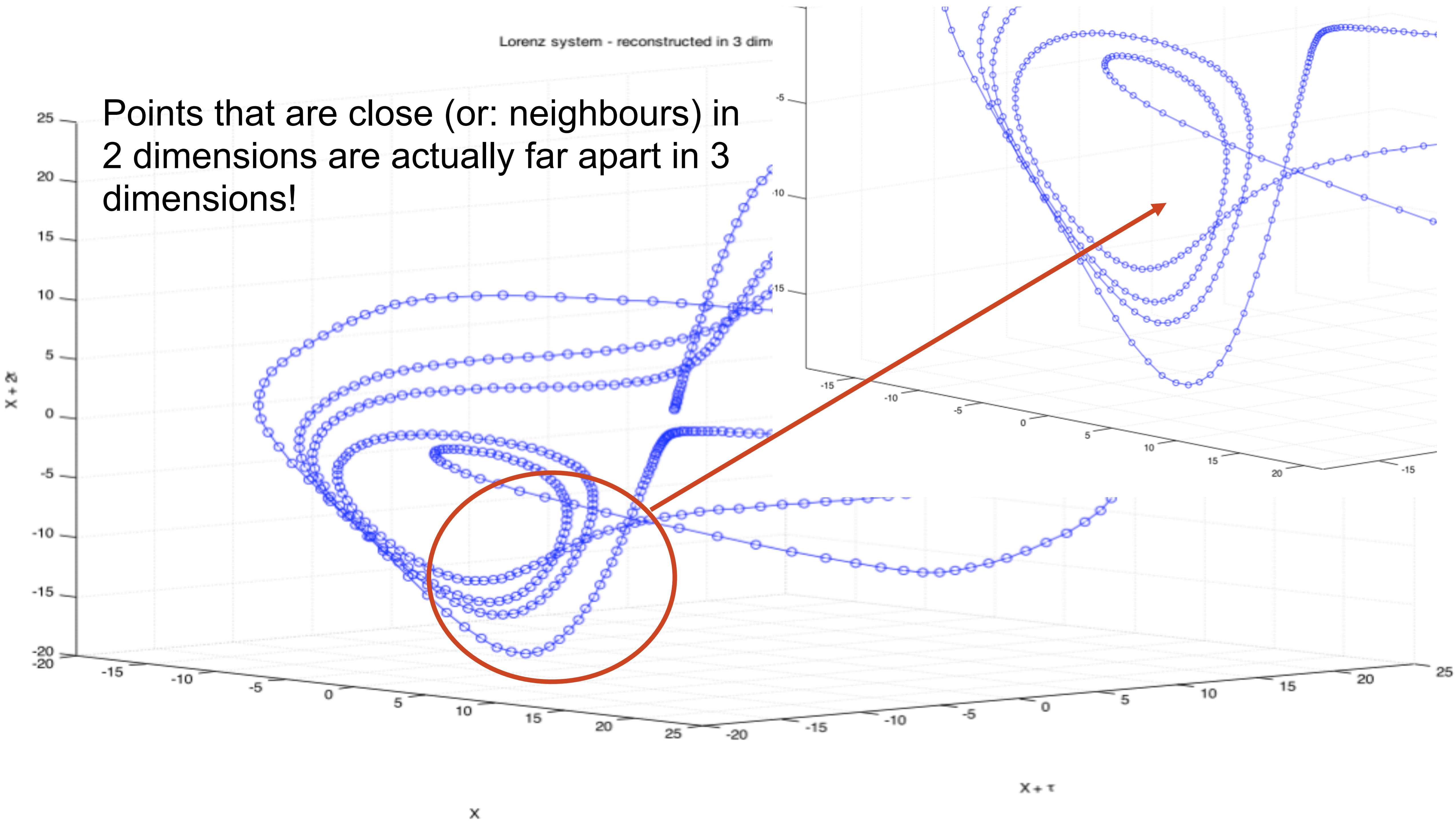
Use first local minimum



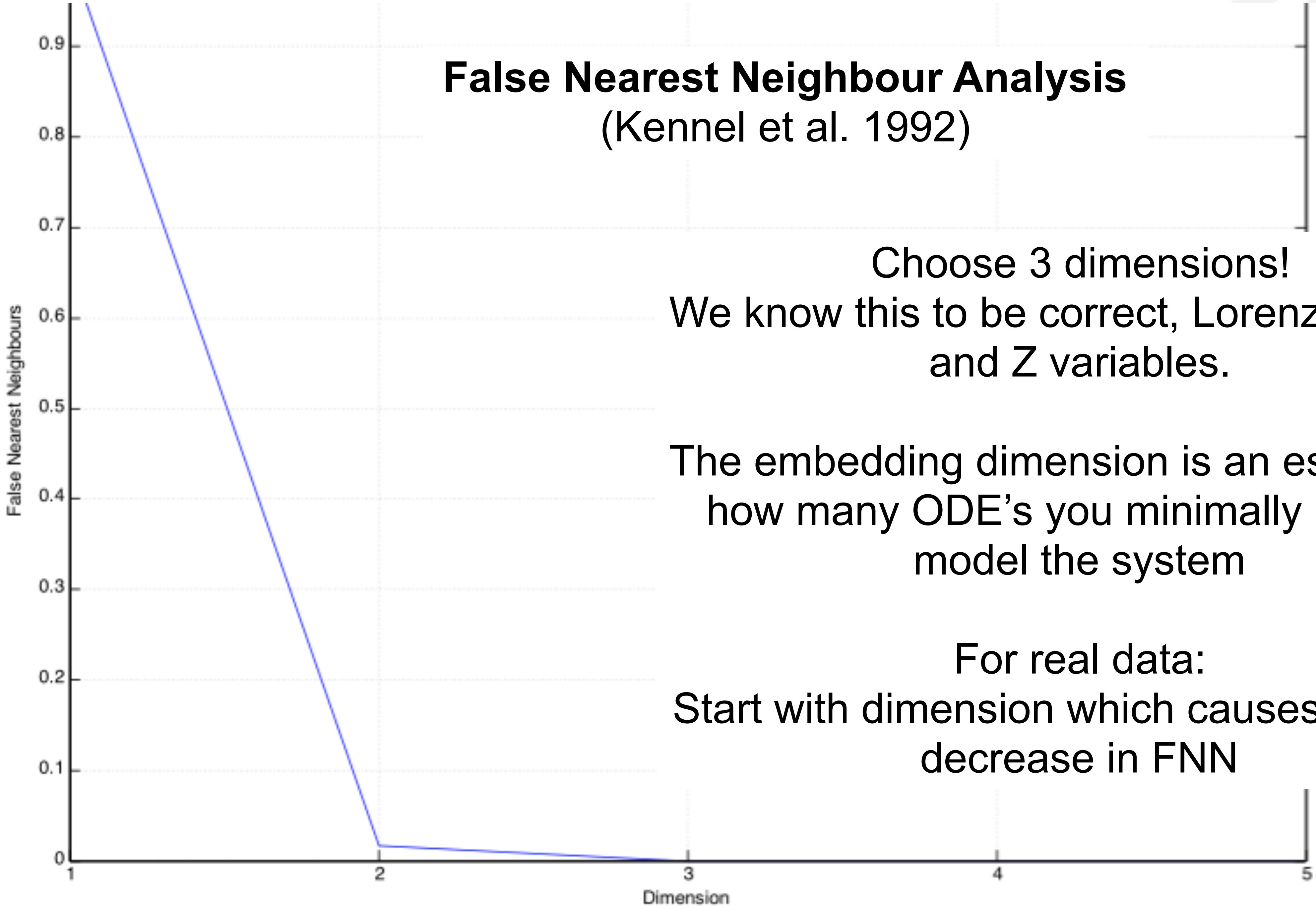
How many dimensions? Determine *embedding dimension* (m)



Lorenz system – Determine embedding dimension



Lorenz system – Determine embedding dimensions



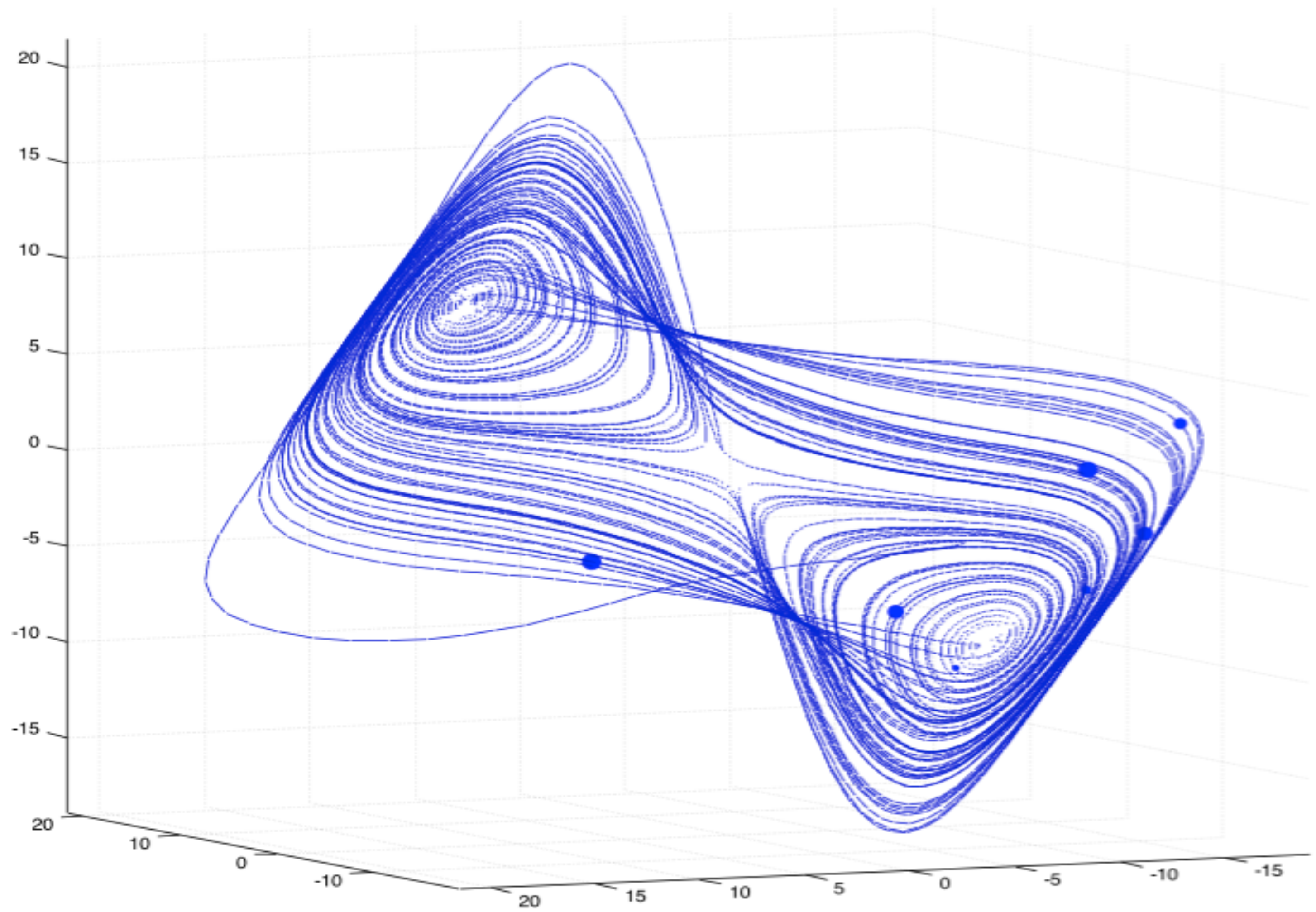
False Nearest Neighbour Analysis
(Kennel et al. 1992)

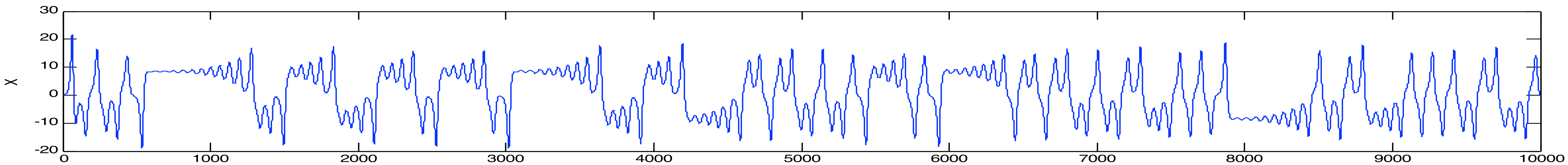
Choose 3 dimensions!
We know this to be correct, Lorenz has X, Y and Z variables.

The embedding dimension is an estimate of how many ODE's you minimally need to model the system

For real data:
Start with dimension which causes greatest decrease in FNN

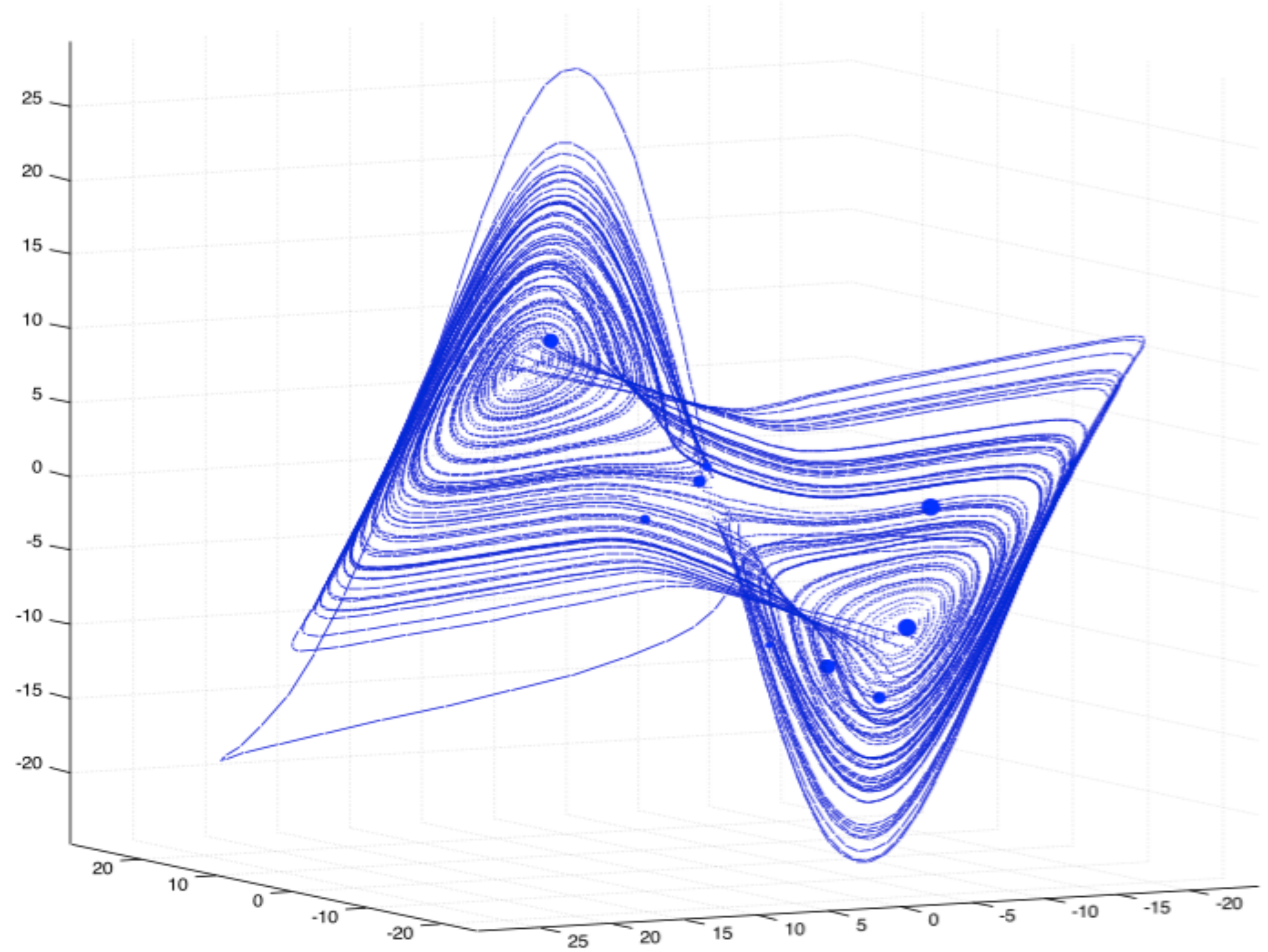
Lorenz system – Reconstruct phase space using X

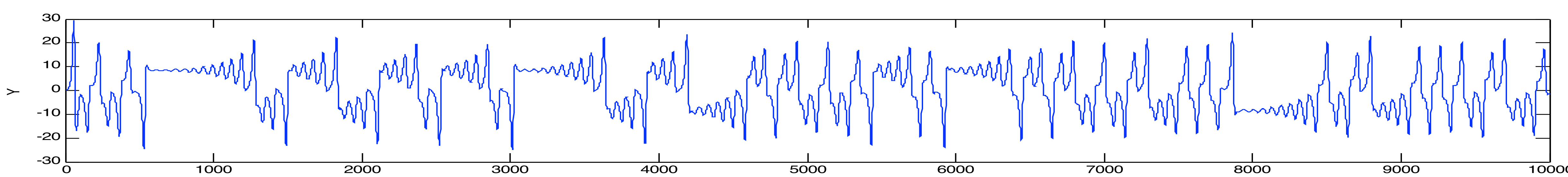




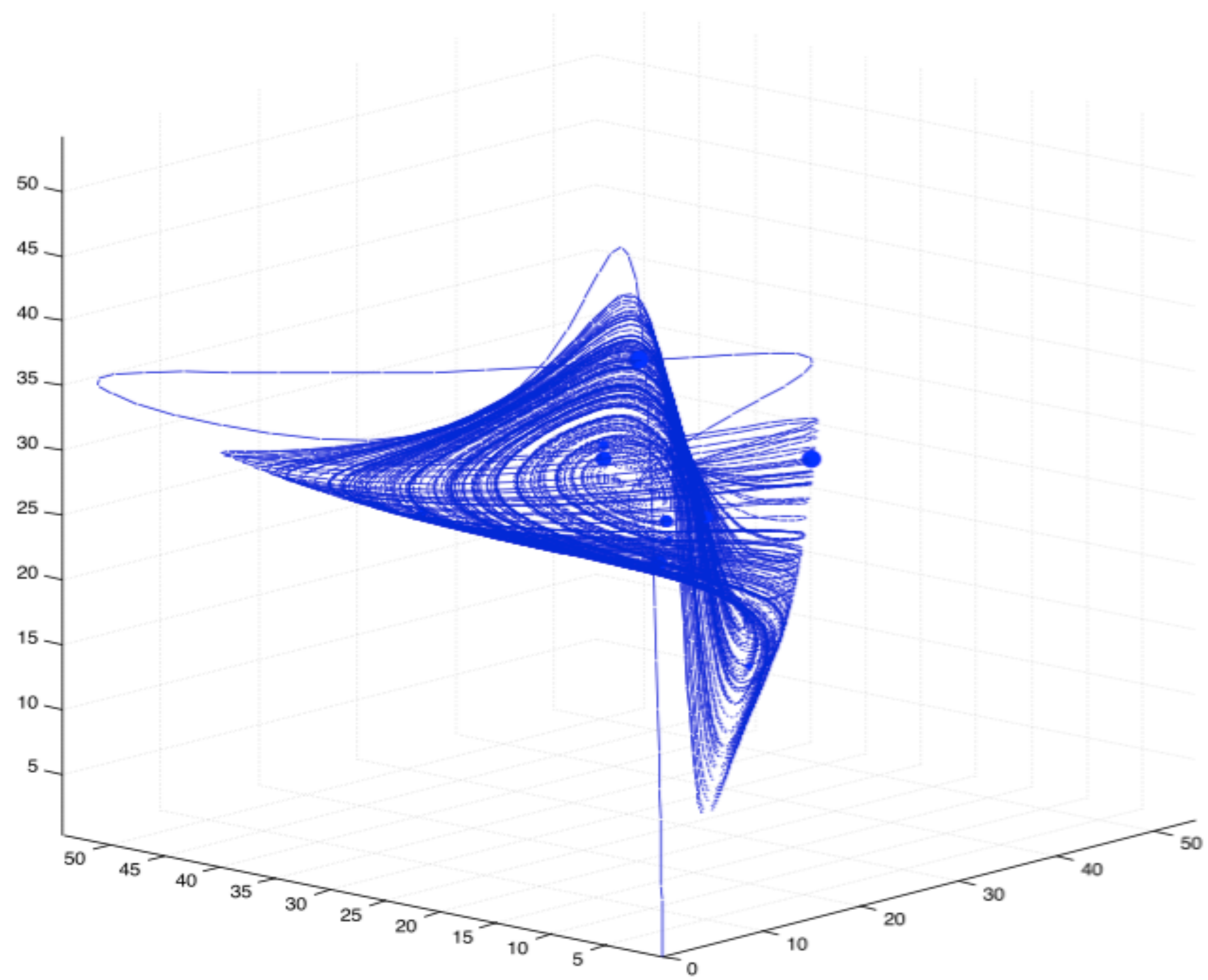
Behavioural Science Institute
Radboud University Nijmegen

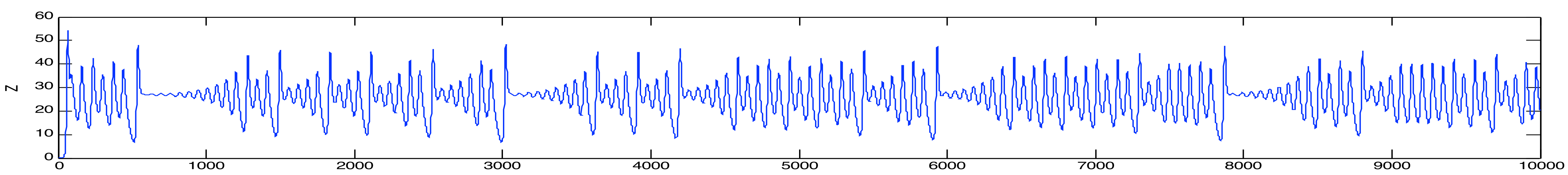
Lorenz system – Reconstruct phase space using Y





Lorenz system – Reconstruct phase space using Z





https://youtu.be/6i57udsPKms

Isn't that amazing?

- Take a moment to realise what we just did:

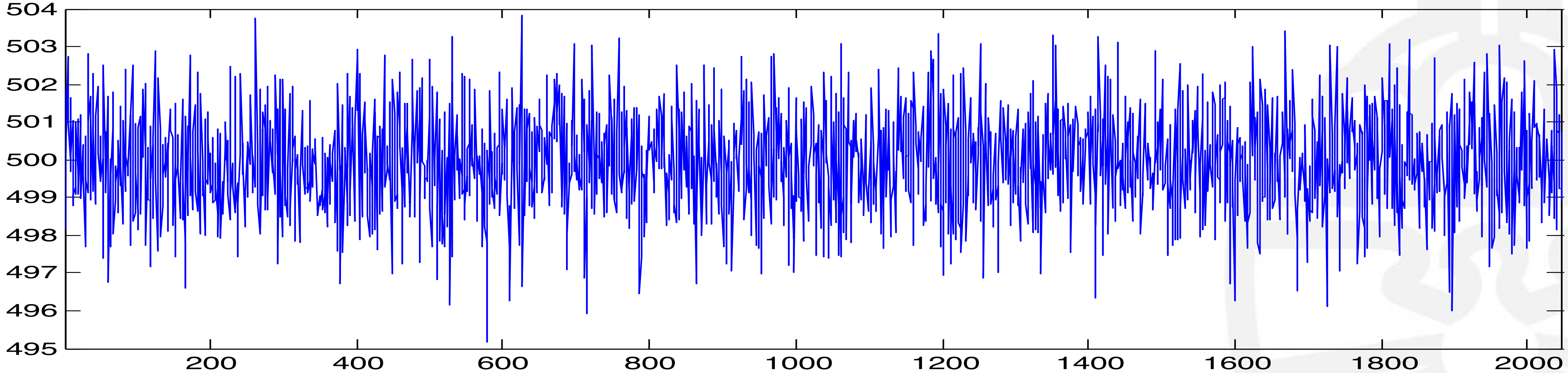
- The state space (defined by X,Y and Z) of a complex, nonlinear chaotic system was reconstructed to a phase space (lag plot) of 3 surrogate dimensions X, $X_{t+\tau}$, $X_{t+2*\tau}$

- **You only need to measure one variable of a system!!**
  … because "everything is interacting"…
  *We **exploit (and need)** the dependencies in the data!*

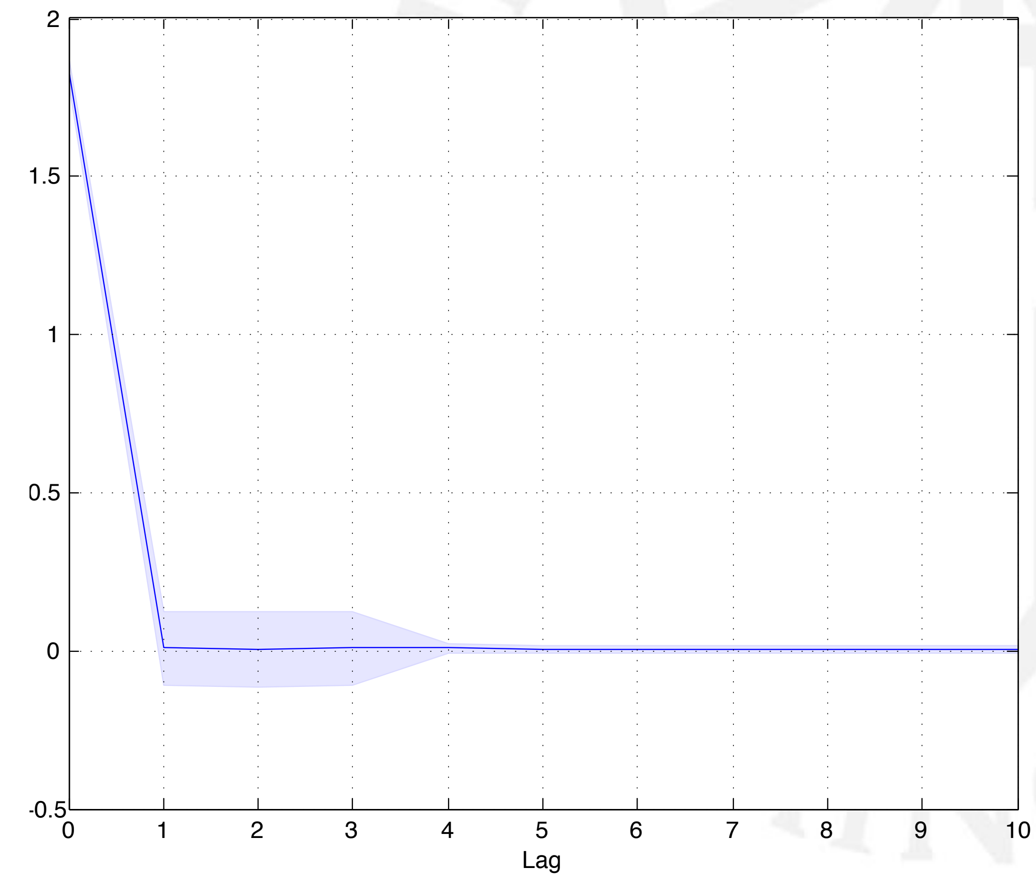
The length of your data set needs to be long enough to create the surrogate dimension.

- The reconstruction process **does not make many assumptions about the data.** You can also try to reconstruct a phase space from a random variable. (What will happen?)

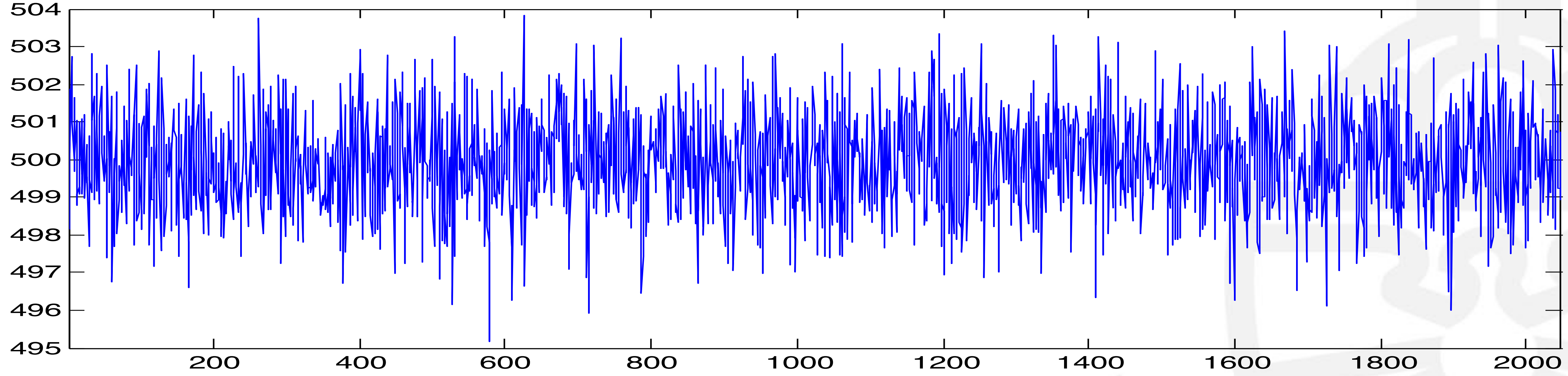
Suppose we have measured a true IID variable



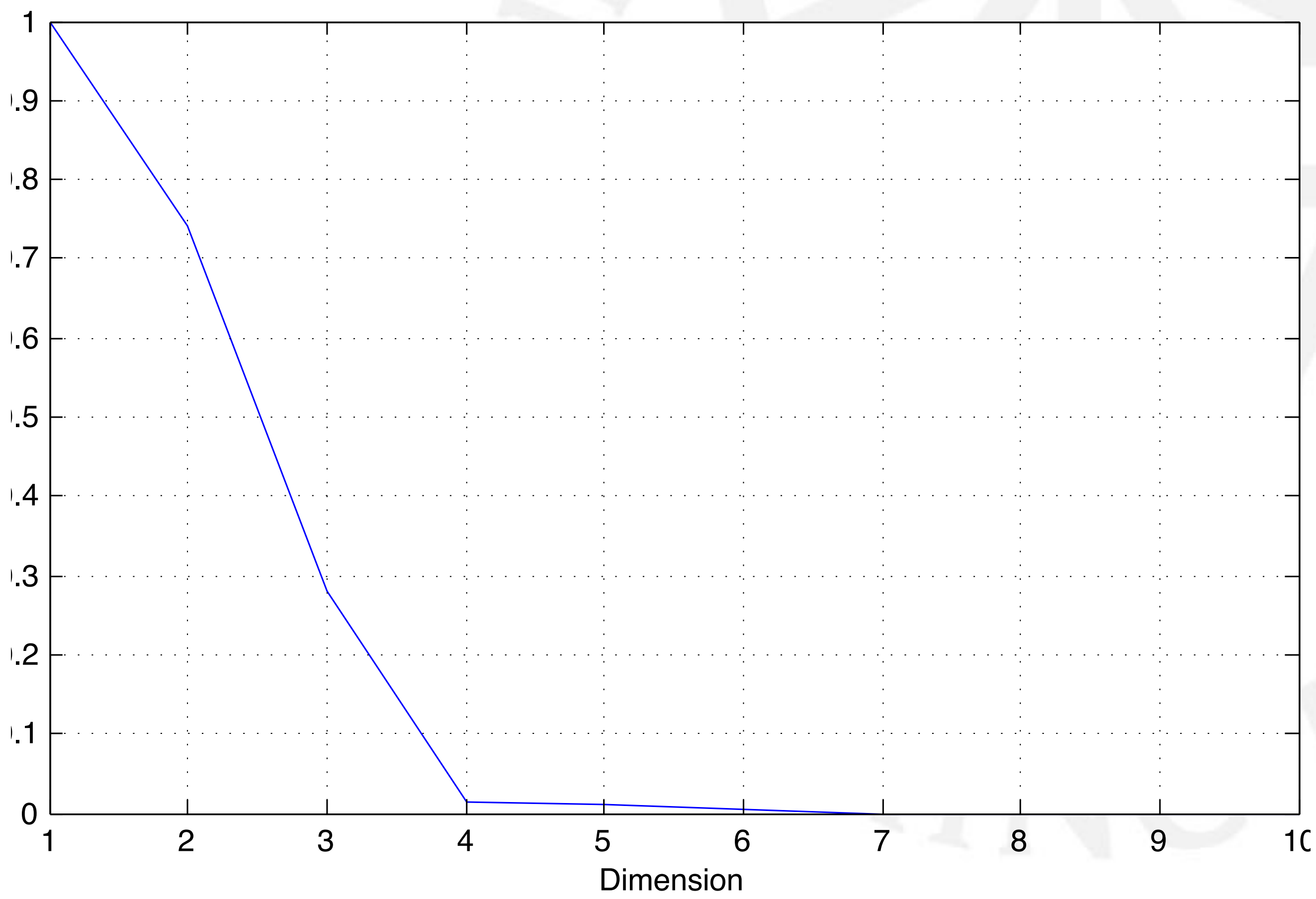
- Determine the embedding lag:
- Lag = 1?



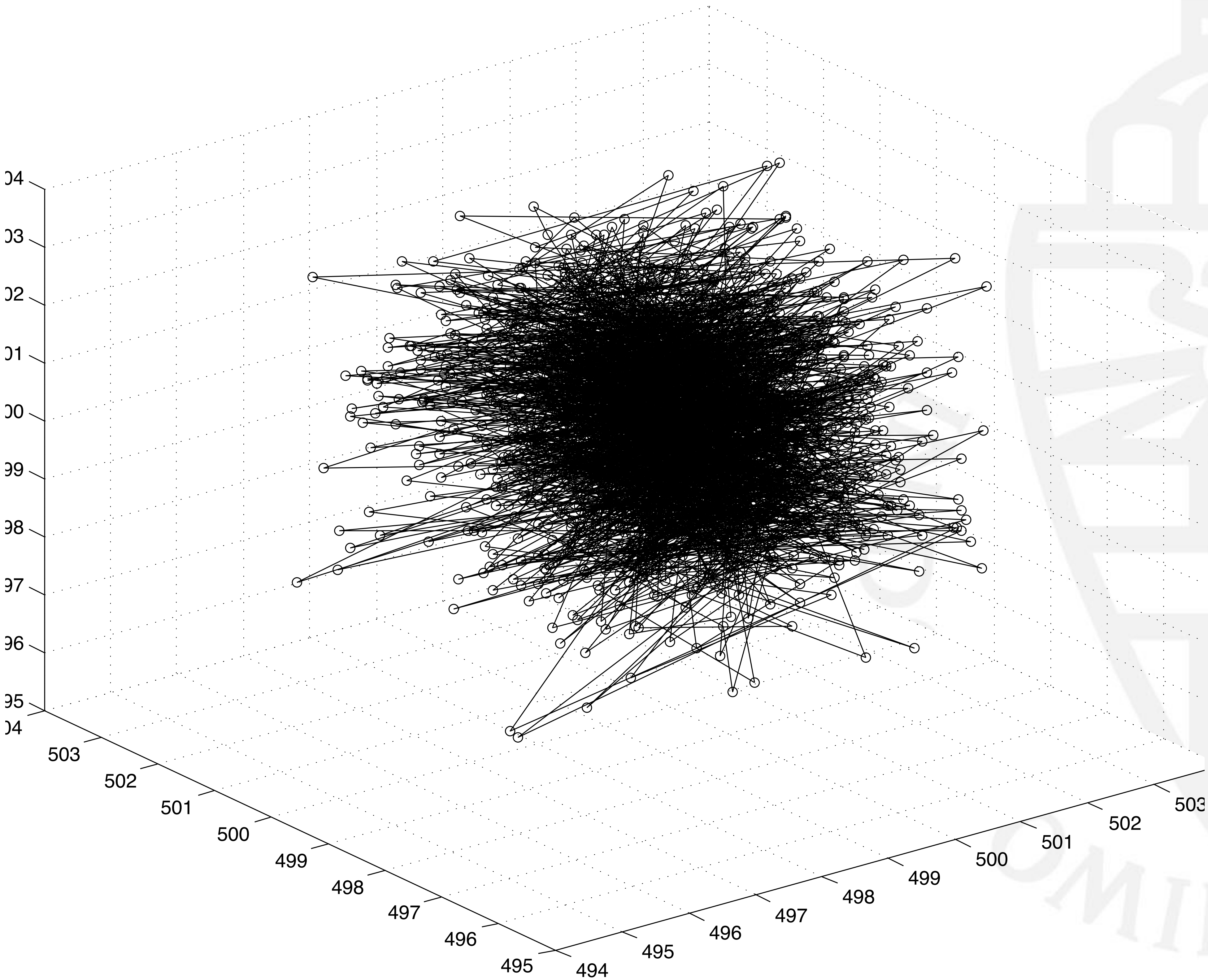
Suppose we have measured a true IID variable



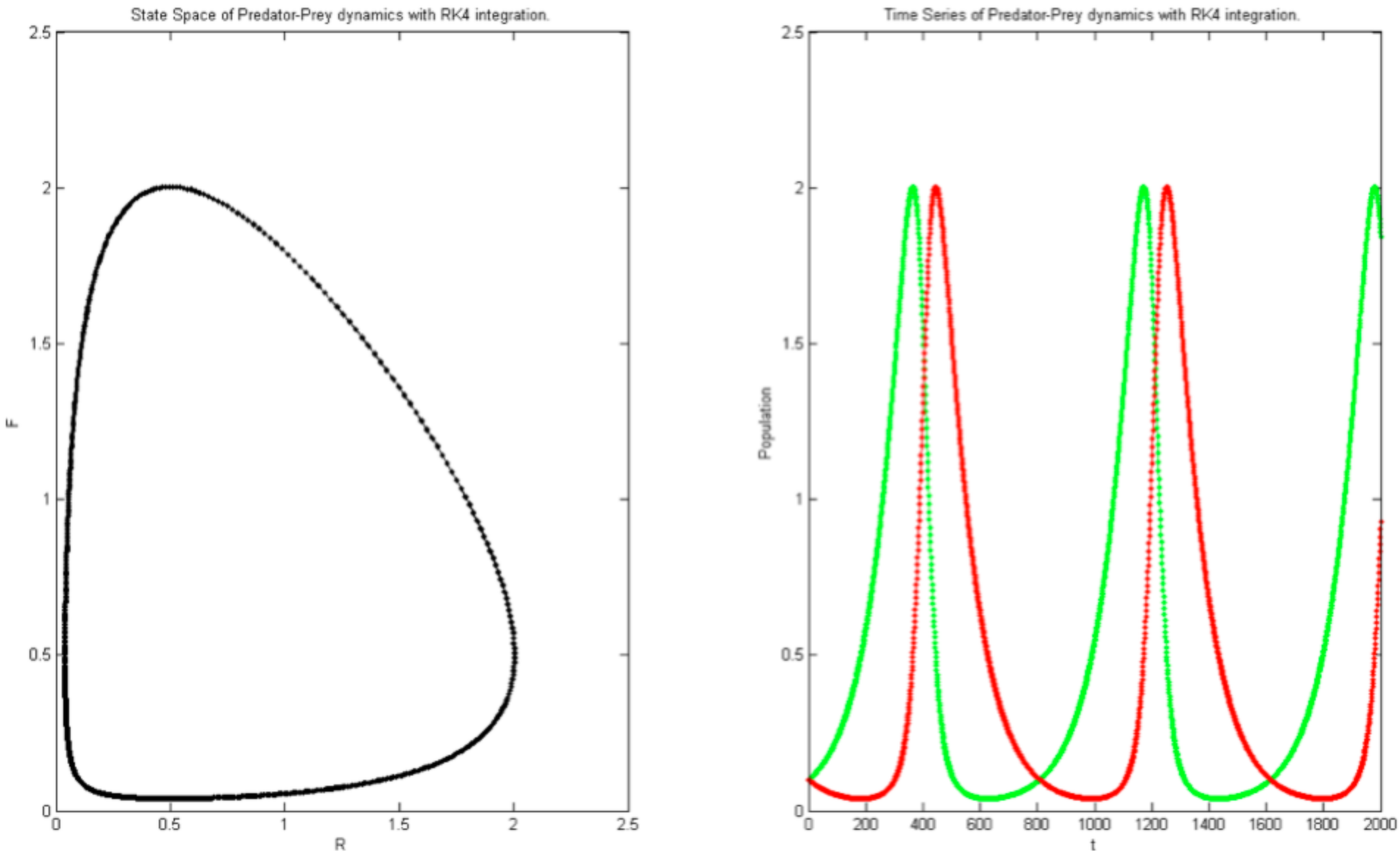
- Determine the embedding dimension:
- Dimension = 4,5,6,7?



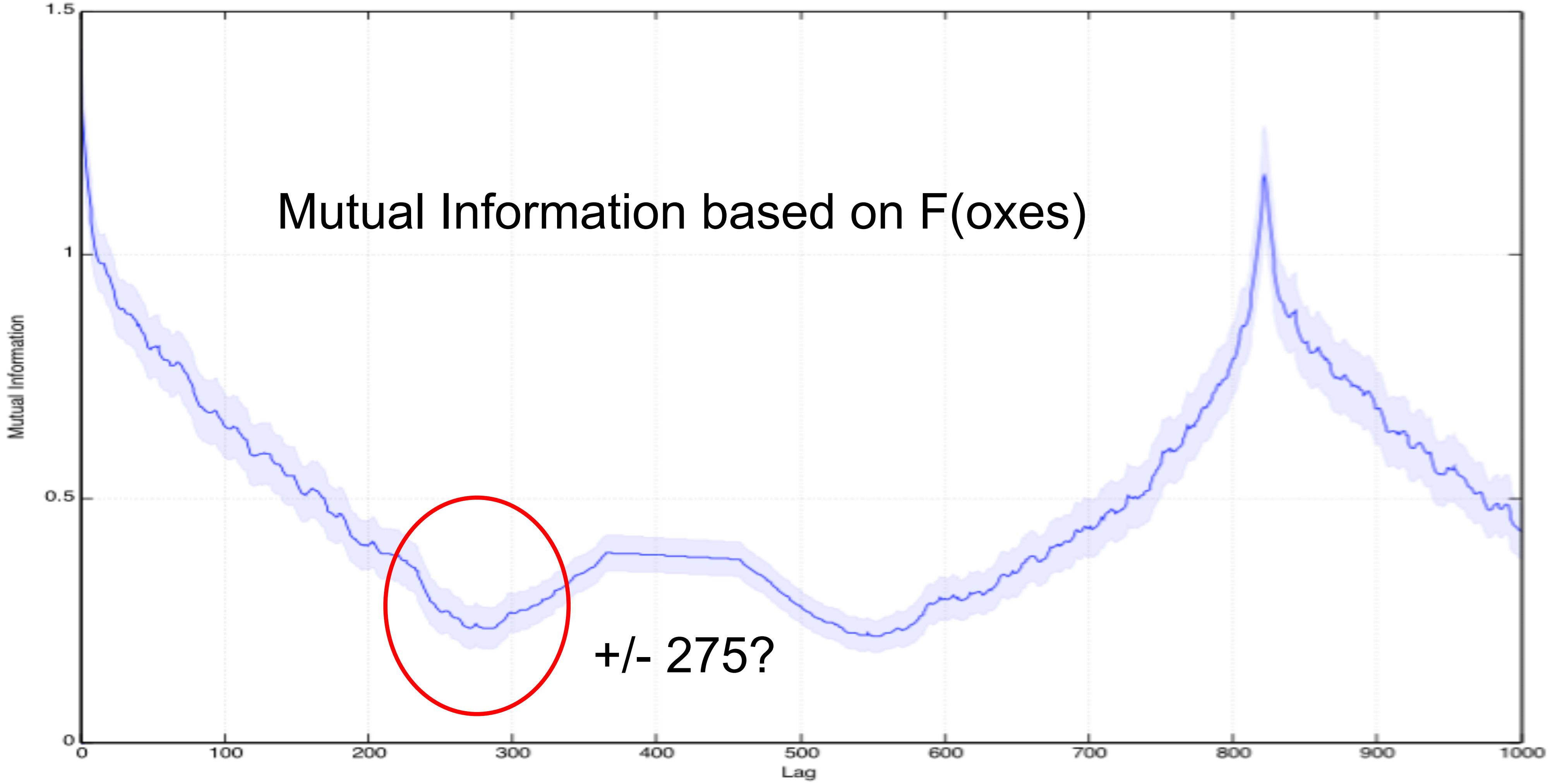
Suppose we have measured a true IID variable

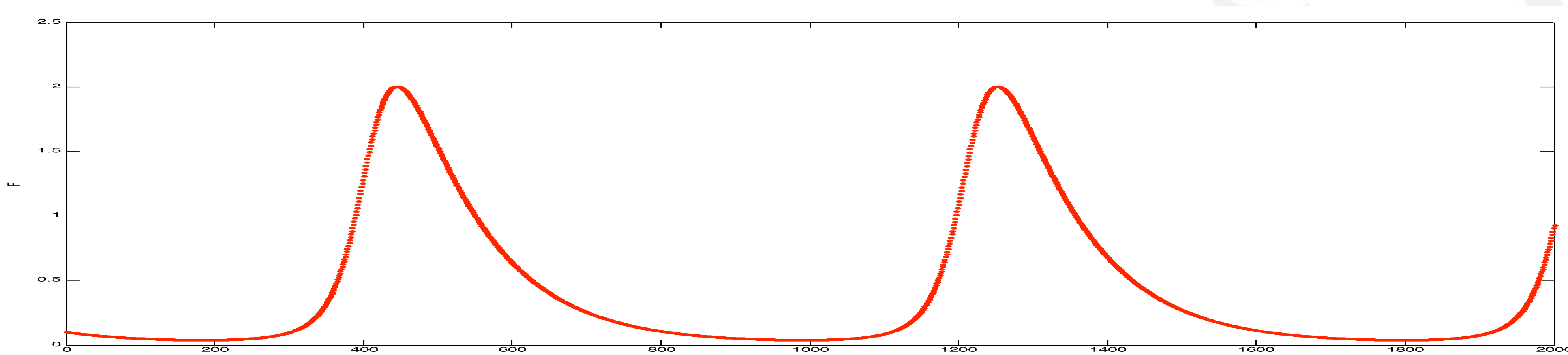


Another familiar example: Predator-Prey dynamics

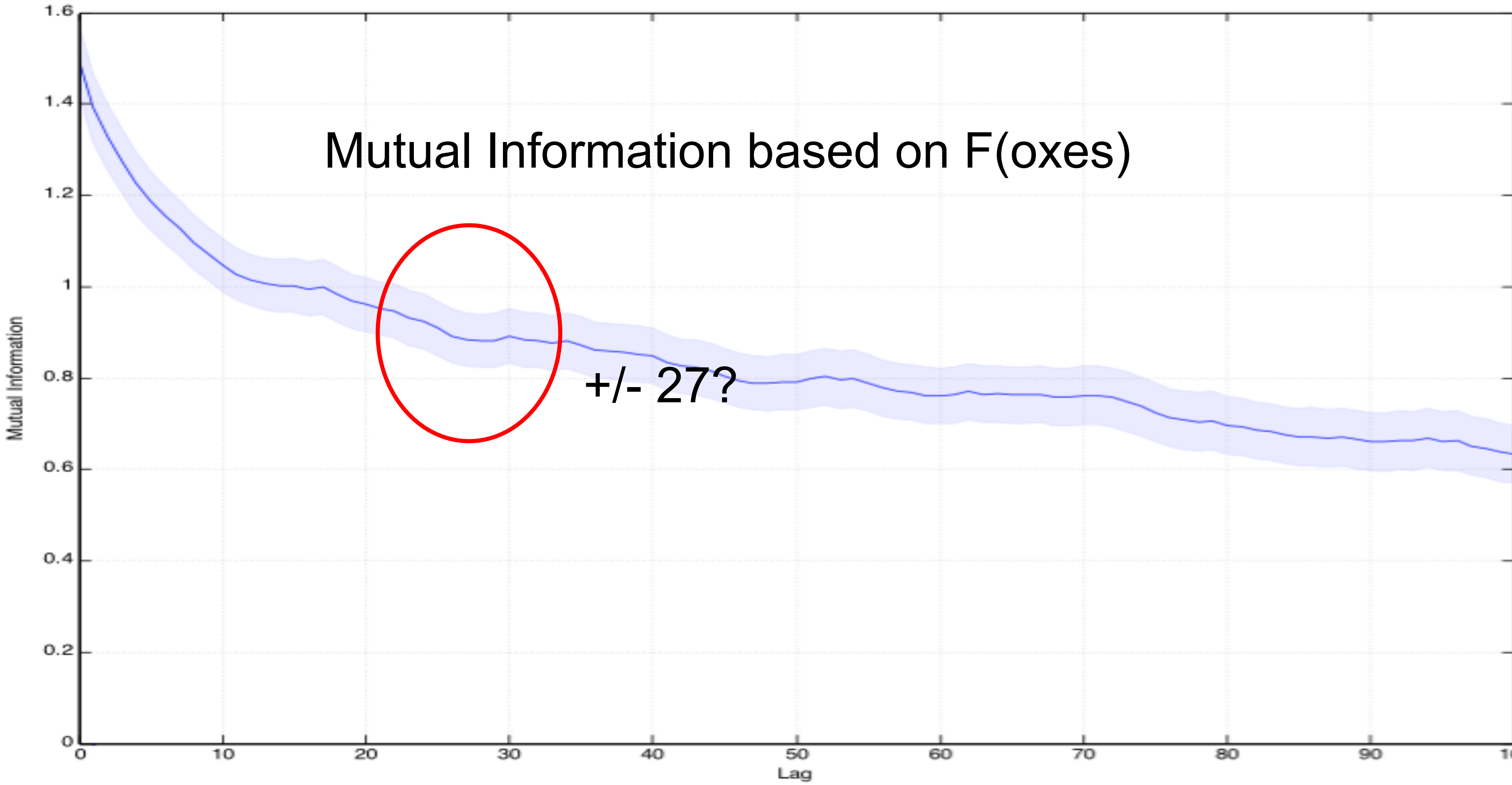


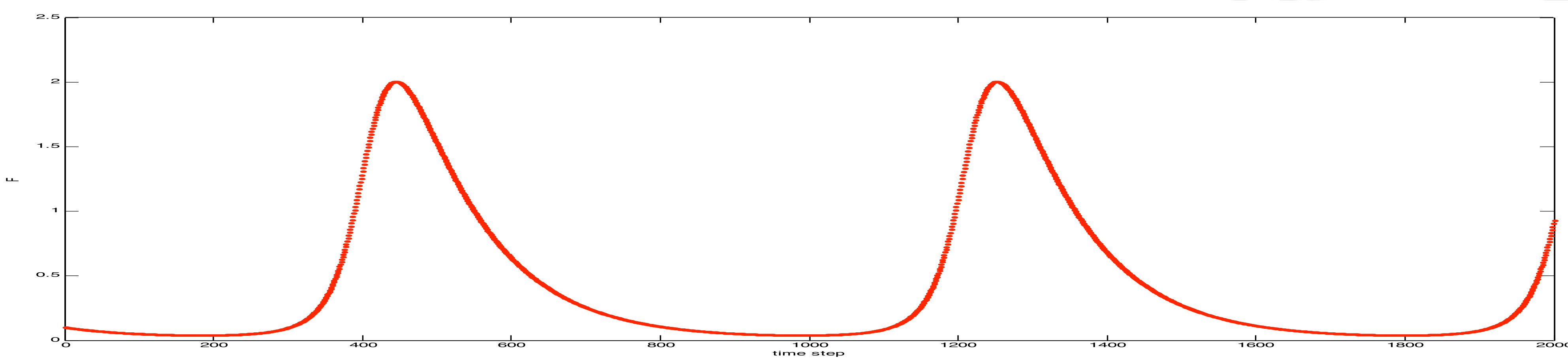
Another familiar example: Predator-Prey dynamics



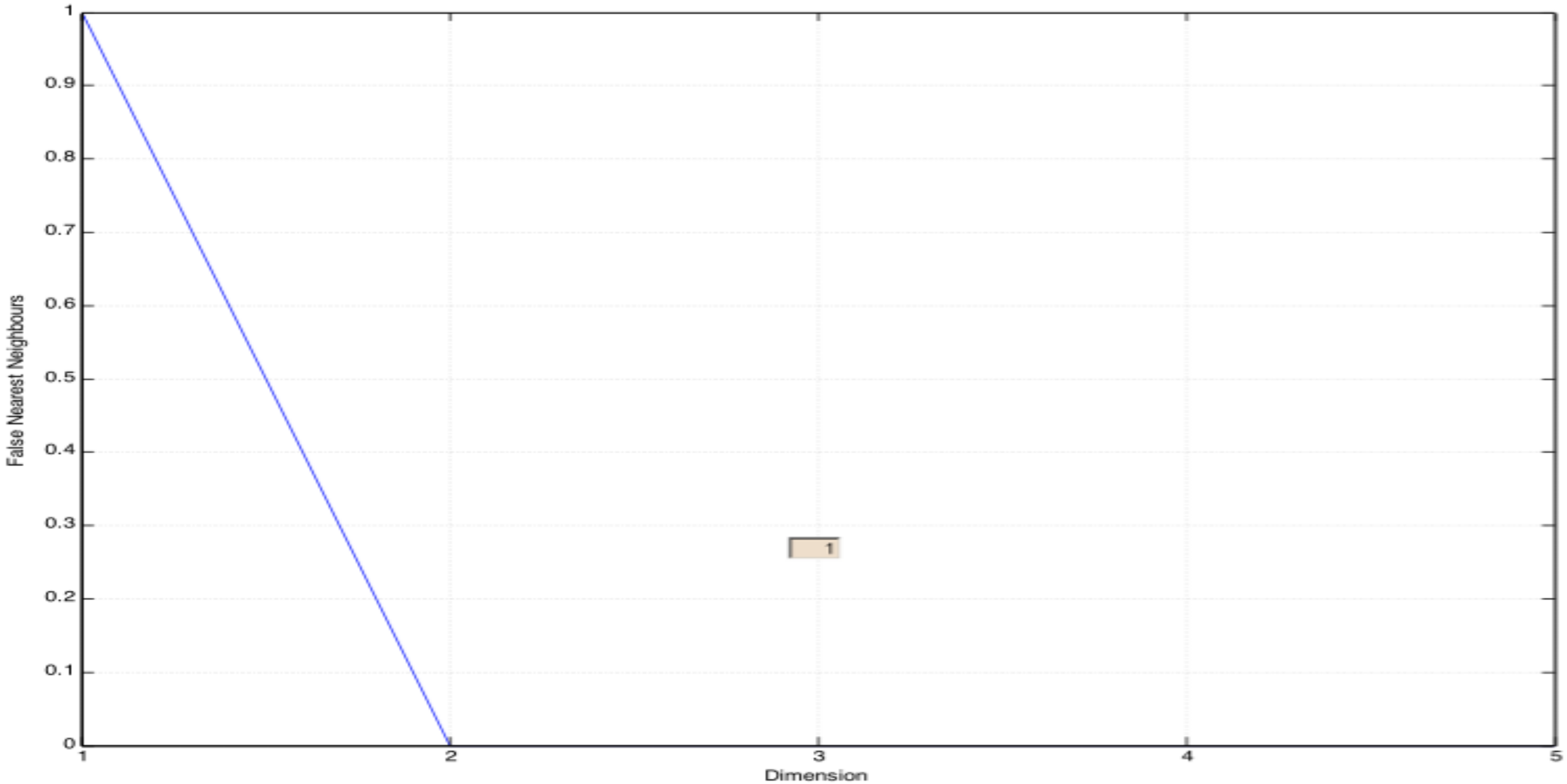


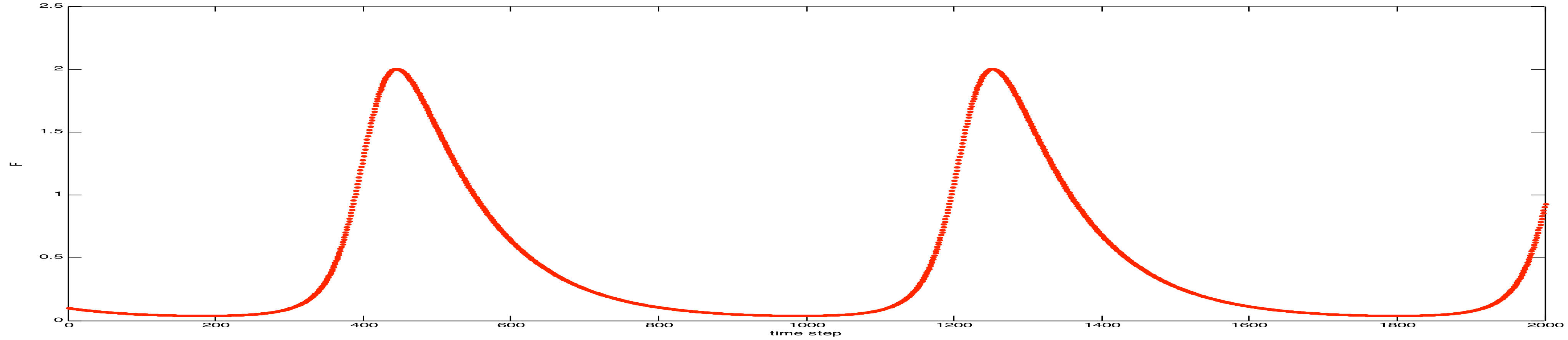
Another familiar example: Predator-Prey dynamics





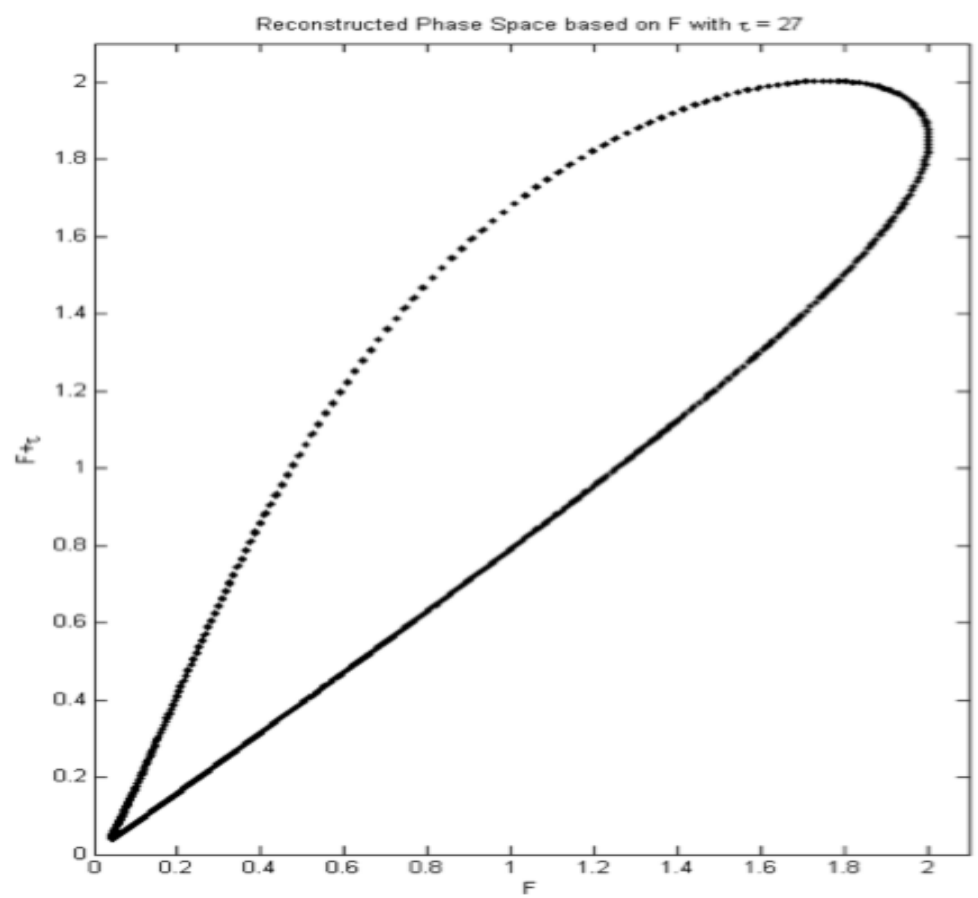
Another familiar example: Predator-Prey dynamics

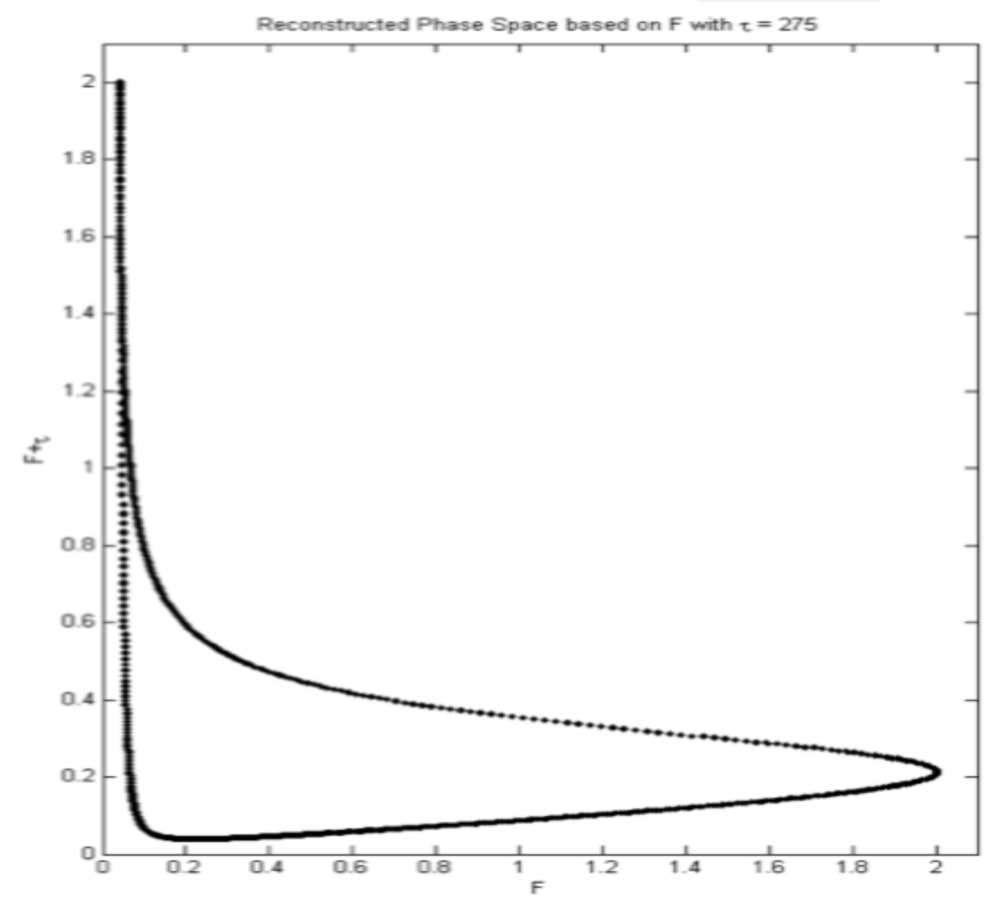


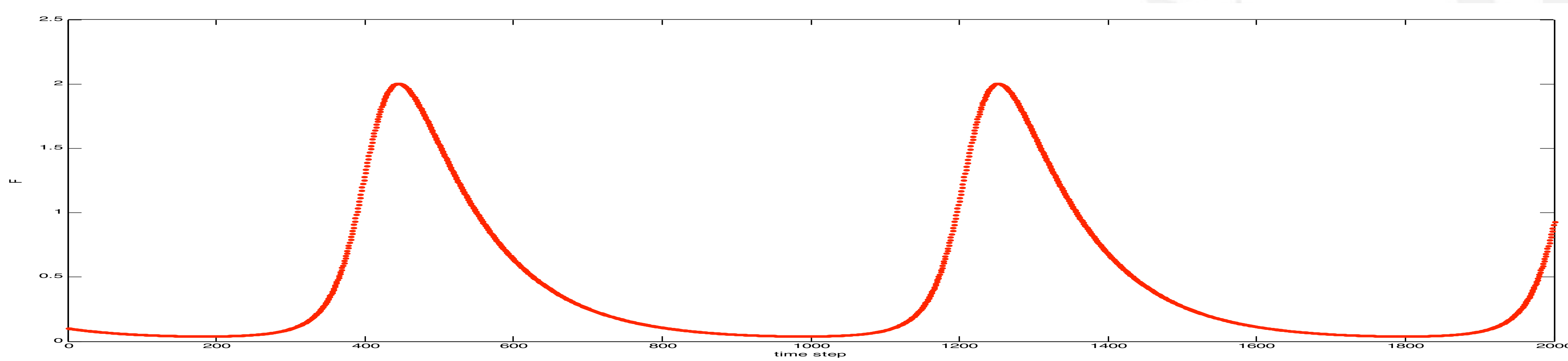


Another familiar example: Predator-Prey dynamics

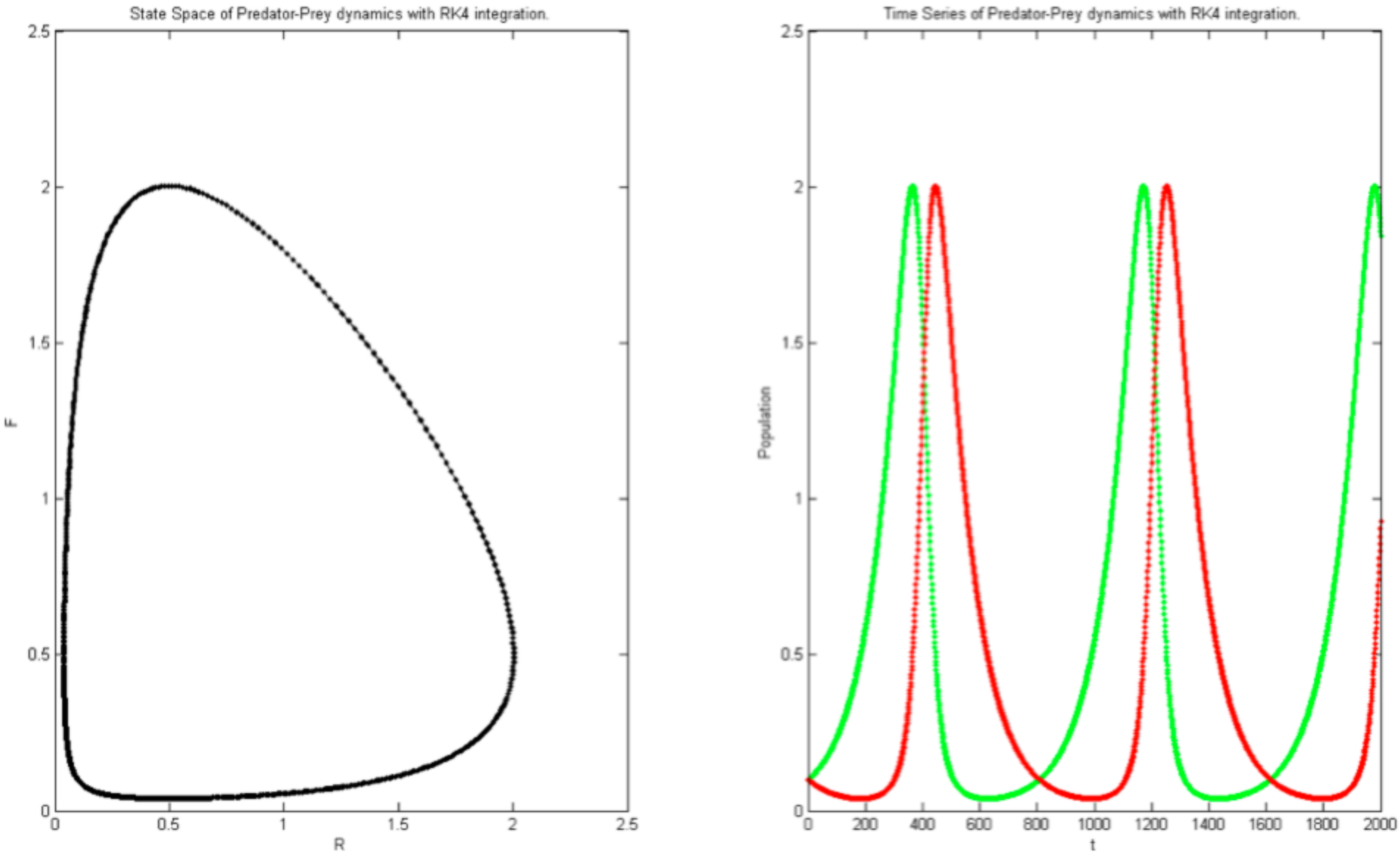
Embedding lag = 27 / 275 Embedding dimensions = 2







Another familiar example: Predator-Prey dynamics



Not so amazing?

- The reconstructed attractor is *'Topologically equivalent'* not exactly the same!!! (compare to random cloud of points)
  The exact lag is not that important, it is just a way to optimize the reconstruction

- If you are working with 'real' data from psychological experiments you will find that the dimensionality needed to describe the system is usually 10 dimensions or higher… No visual inspection anymore!

- Solution: Quantify the dynamic behaviour of the system in state space in terms of periodicity, randomness, etc. This remains similar to the original dynamics even if the attractor is not reconstructed exactly the same way (the reconstructed attractor is still much more constrained than all the states theoretically possible).

- (Cross) Recurrence Quantification Analysis!

Recurrence Quantification Analysis

- A technique to quantify the dynamical behaviour of the system in its’ reconstructed phase space
- Already many applications in physics, biology and psychology
- Many flavours of RQA, today: Auto-Recurrence.
- Next week: *Cross Recurrence Quantification* (analyse if two signals share the same phase space, are synchronised), *Categorical data*, *Order Patterns Recurrence* and *Lagged Recurrence*

Recurrence Quantification Analysis

Consider this poem by Jules Deelder
to be a time series of letters.

Blues on tuesday

Geen geld.
Geen vuur.
Geen speed.

Geen krant.
Geen wonder.
Geen weed.

Geen brood.
Geen tijd.
Geen weet.

Geen klote.
Geen donder.
Geen reet.

Recurrence Quantification Analysis

Blues on tuesday

Geen geld.
Geen vuur.
Geen speed.

Geen krant.
Geen wonder.
Geen weed.

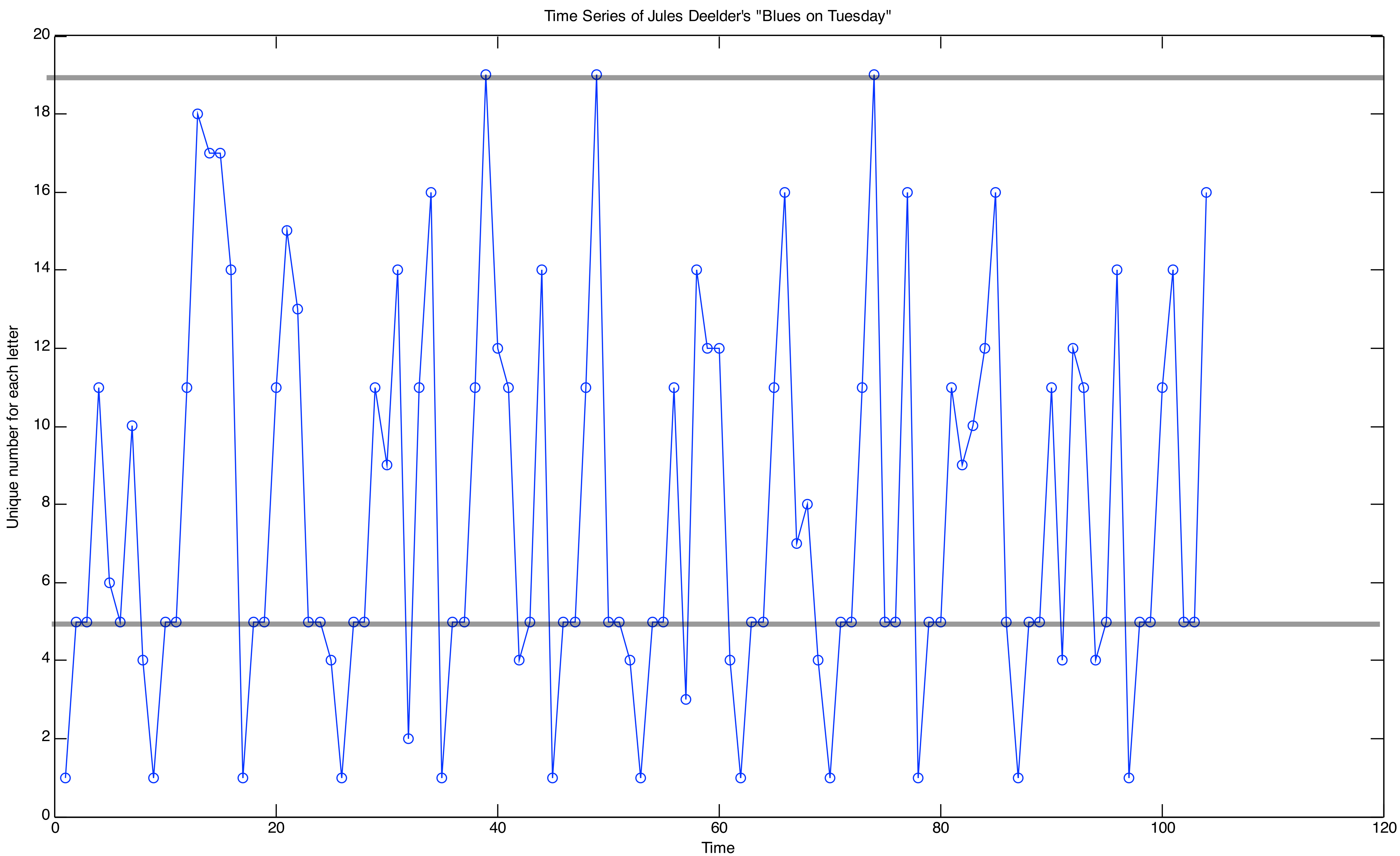
Geen brood.
Geen tijd.
Geen weet.

Geen klote.
Geen donder.
Geen reet.

g = 1
e = 2
e = 2
n = 3
g = 1
e = 2
l = 4
d = 5
g = 1
e = 2
e = 2
n = 3
v = 6
u = 7
u = 7
r = 8
…

Recurrence Quantification Analysis

Blues on tuesday



Plot a point every time a letter (value) recurs

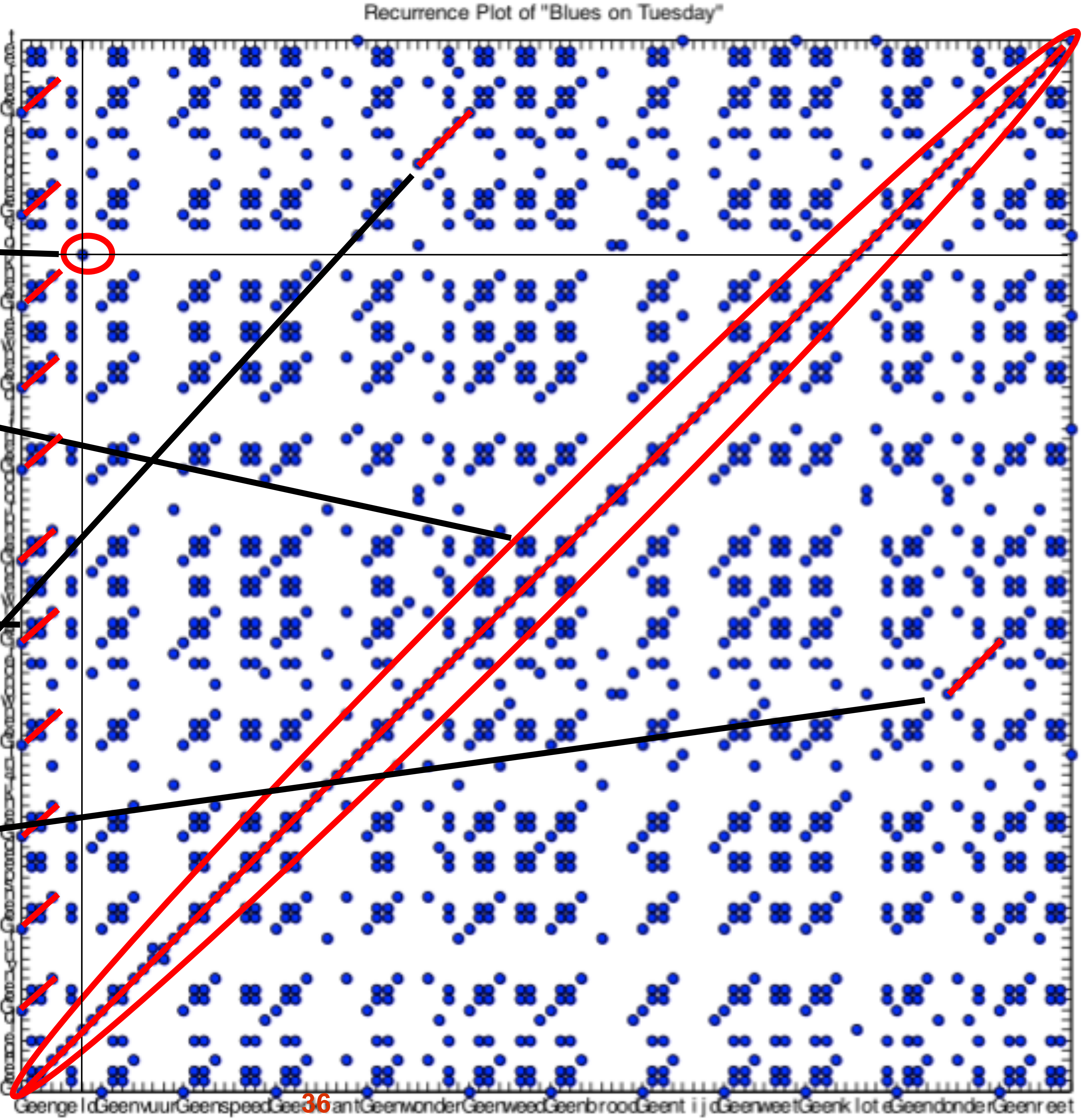
Letter L recurs 1 time

Line of Identity

Plot is symmetrical: Auto-recurrence

Larger recurring patterns form diagonal lines: GEEN

What's this? Recurrence of: ONDER



A Recurrence Plot is a way to quantify the dynamics of a system in its reconstructed phase space

- Look in the reconstructed phase space when a value is recurring
- For continuous data: The value does not have to be exactly the same. Decide on a radius in which it is acceptable to call it recurring
- Many measures can be computed from recurrence plots!

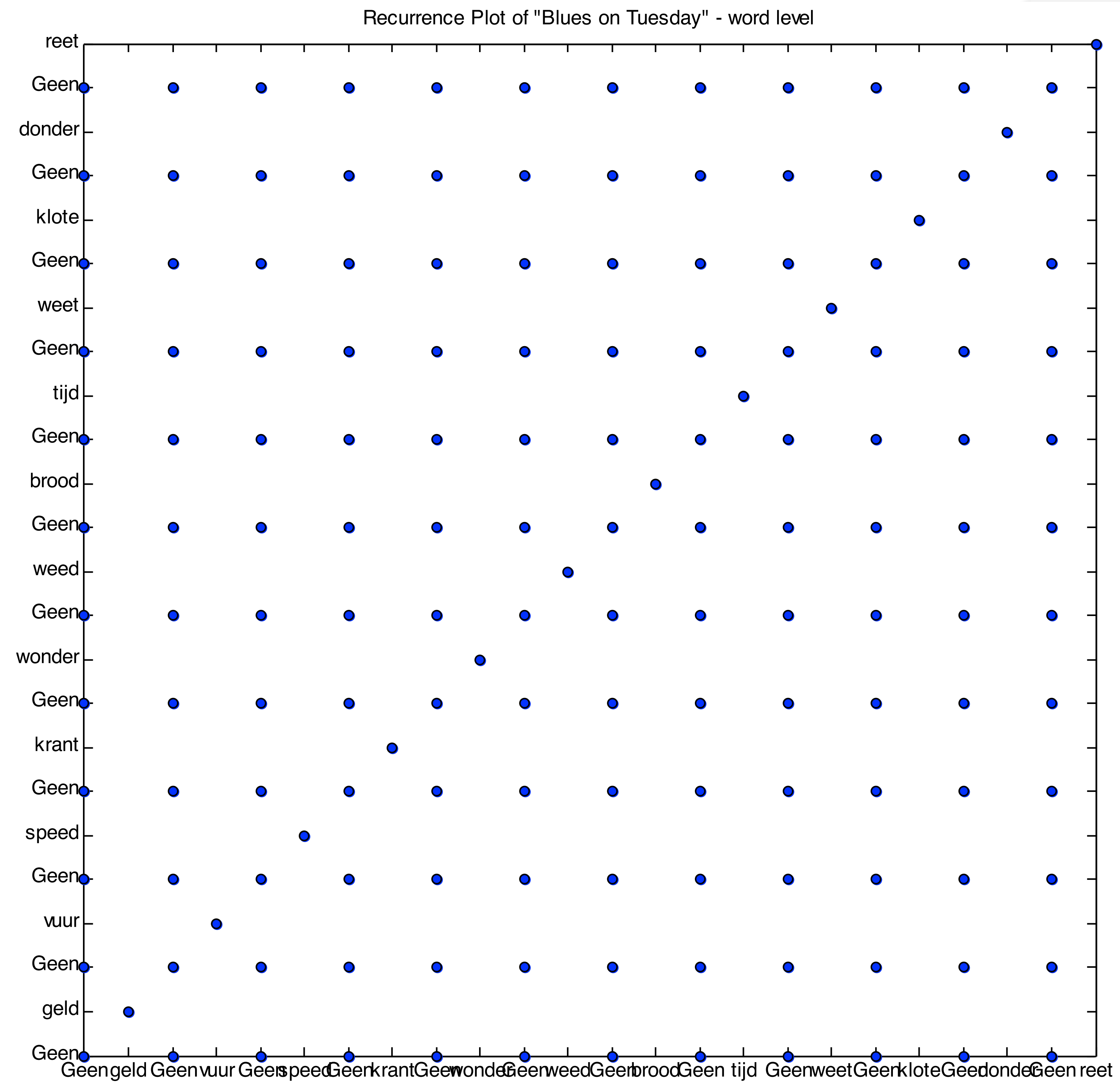
What would a plot of the poem at the word level look like?

Geen geld.
Geen vuur.
Geen speed.

Geen krant.
Geen wonder.
Geen weed.

Geen brood.
Geen tijd.
Geen weet.

Geen klote.
Geen donder.
Geen reet.



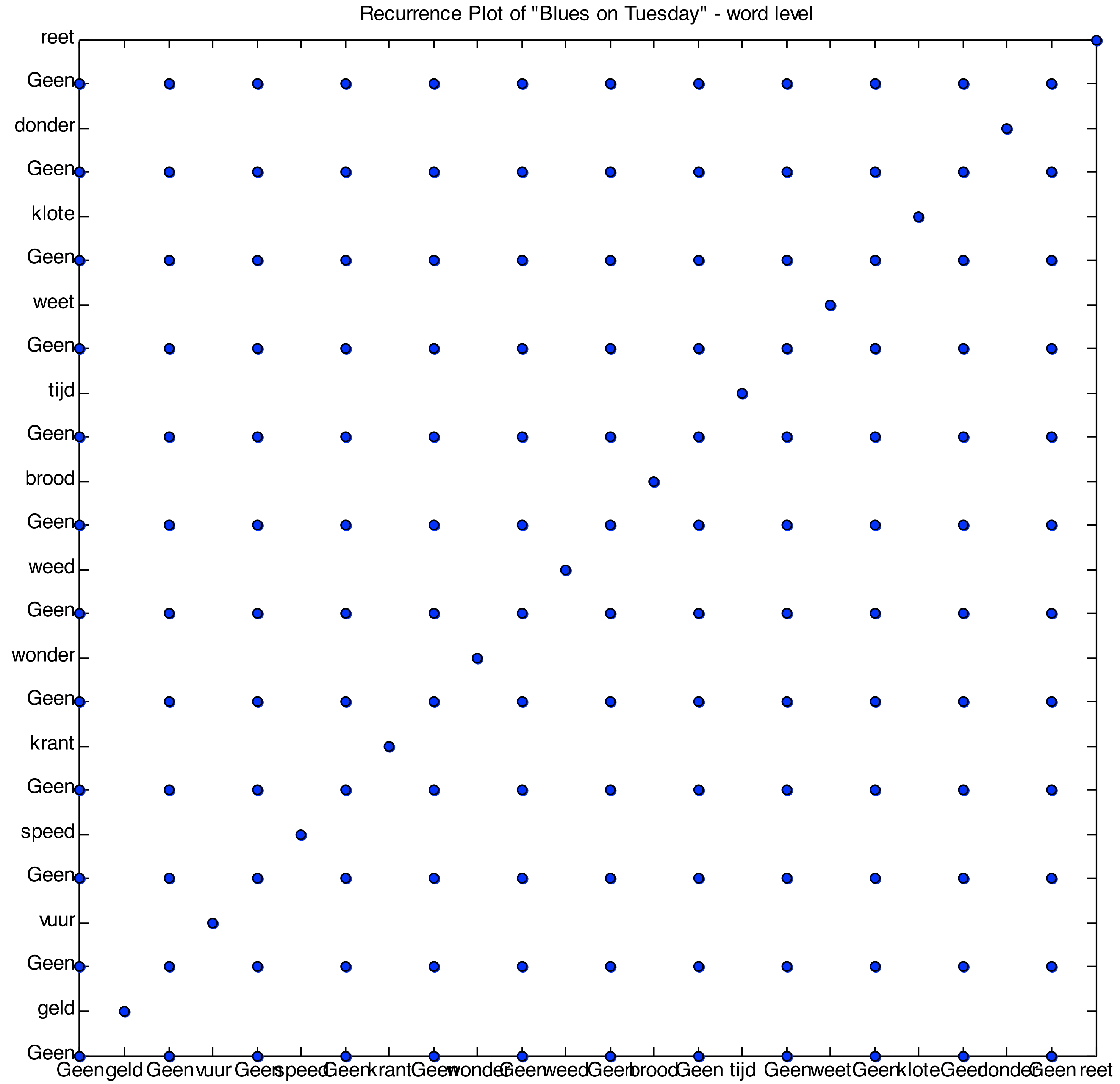
What would a plot of the poem at the word level look like?

Just the word: GEEN

It recurs very regularly

How to quantify this?

Back to Lorenz

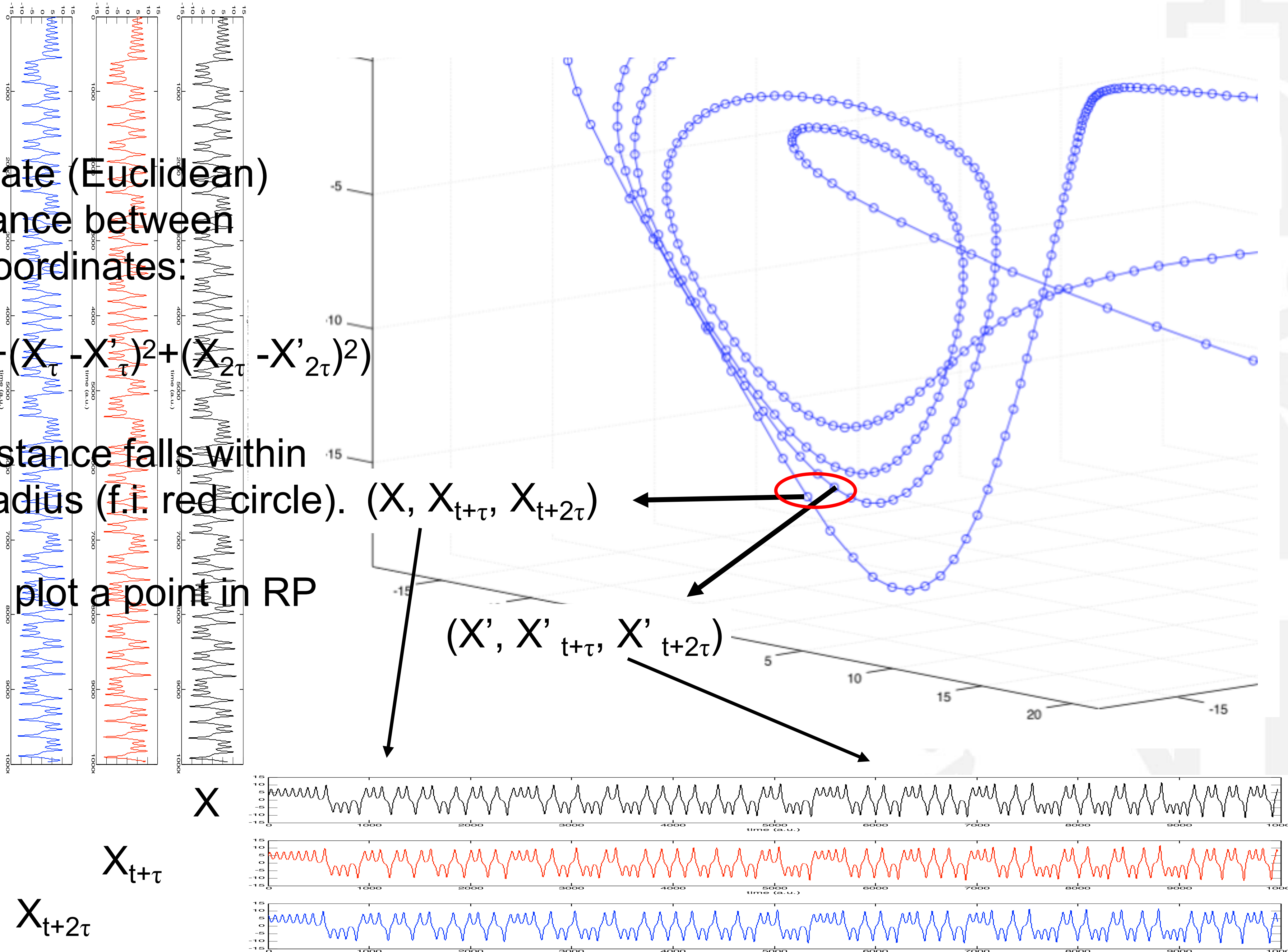


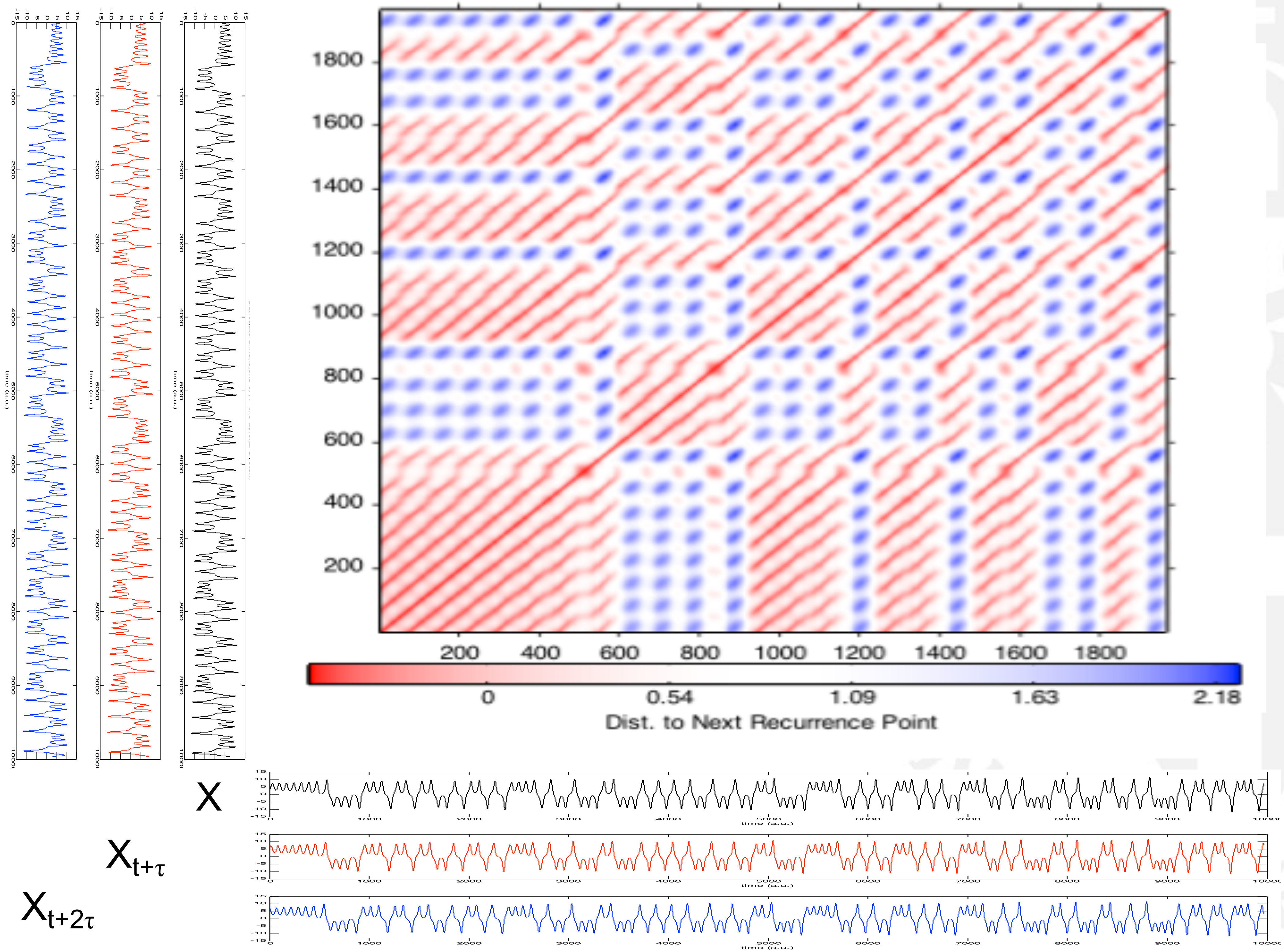
Calculate (Euclidean) distance between coordinates:

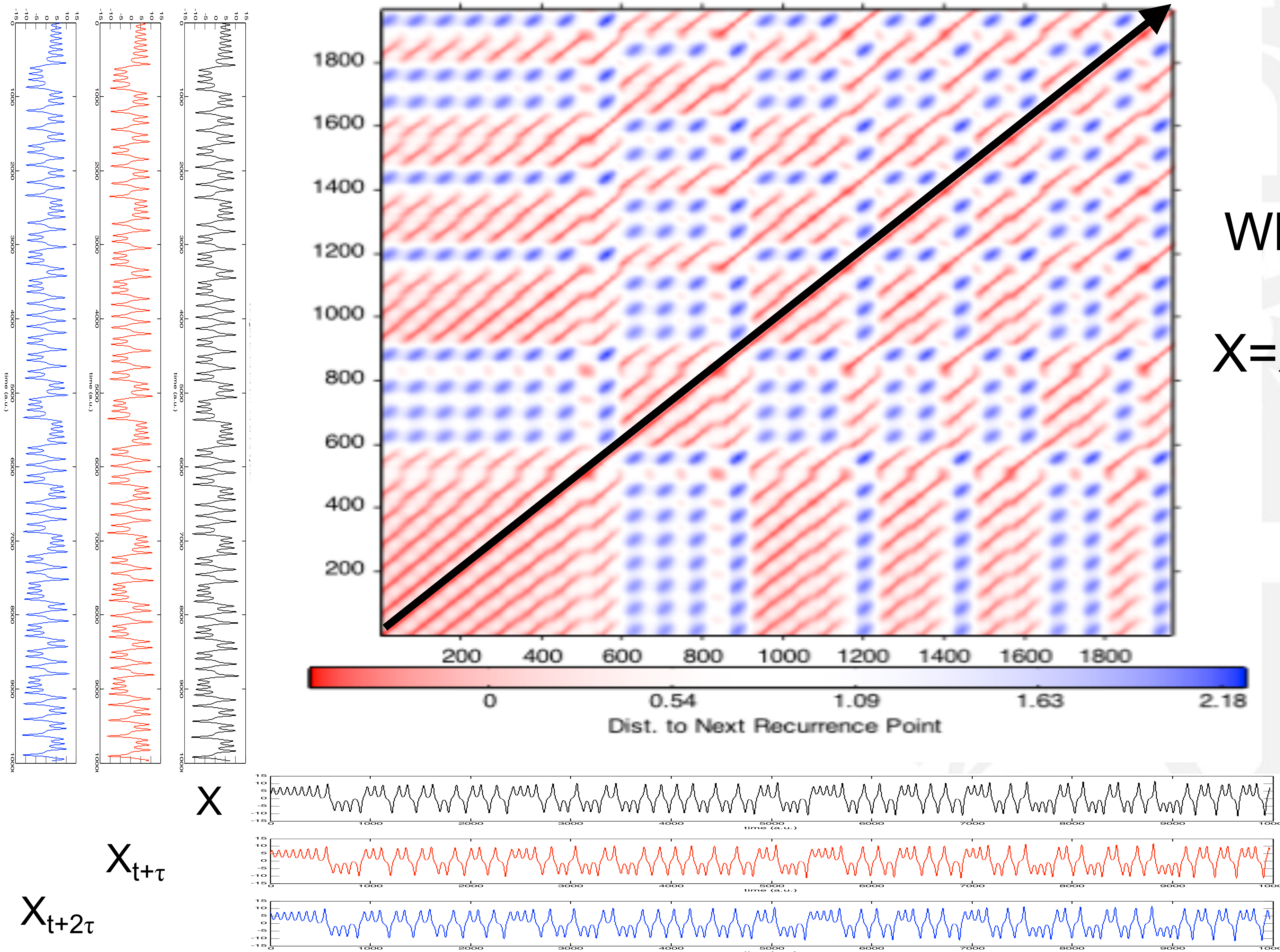
$\mathrm{sqrt}((X-X')^2+(X_\tau - X'_\tau)^2+(X_{2\tau} - X'_{2\tau})^2)$

See if distance falls within a certain radius (f.i. red circle).

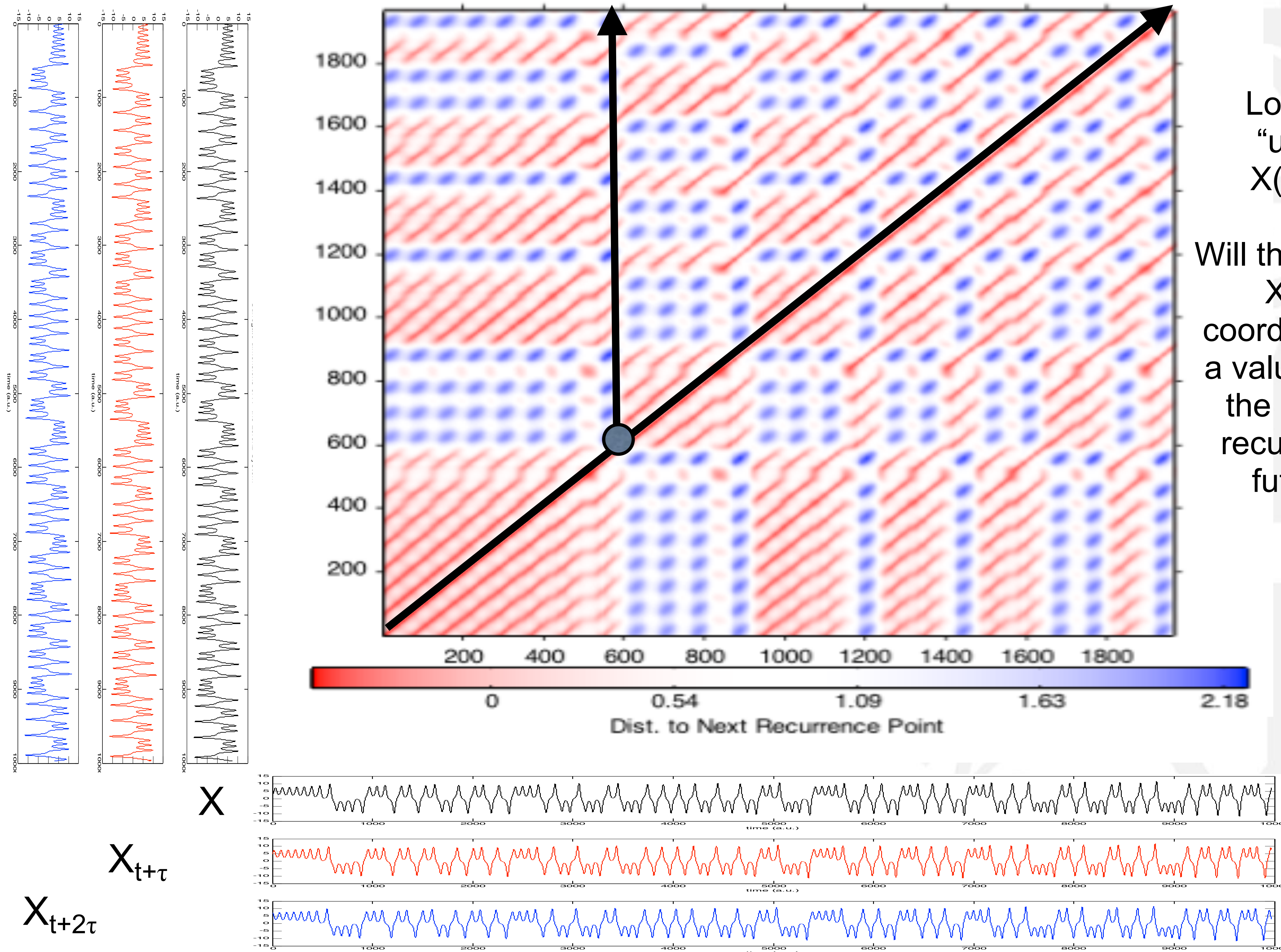
If it does, plot a point in RP




1800
1600
1400
1200
1000
800
600
400
200
200
400
600
800
1000
1200
1400
1600
1800
0
0.54
1.09
1.63
2.18
Dist. to Next Recurrence Point
X
$X_{t+\tau}$
$X_{t+2\tau}$


1800
1600
1400
1200
1000
800
600
400
200
200 400 600 800 1000 1200 1400 1600 1800
0 0.54 1.09 1.63 2.18
Dist. to Next Recurrence Point
X
$X_{t+\tau}$
$X_{t+2\tau}$
time (a.u.)

Where is X=X(t)?



Looking "up" at X(600):

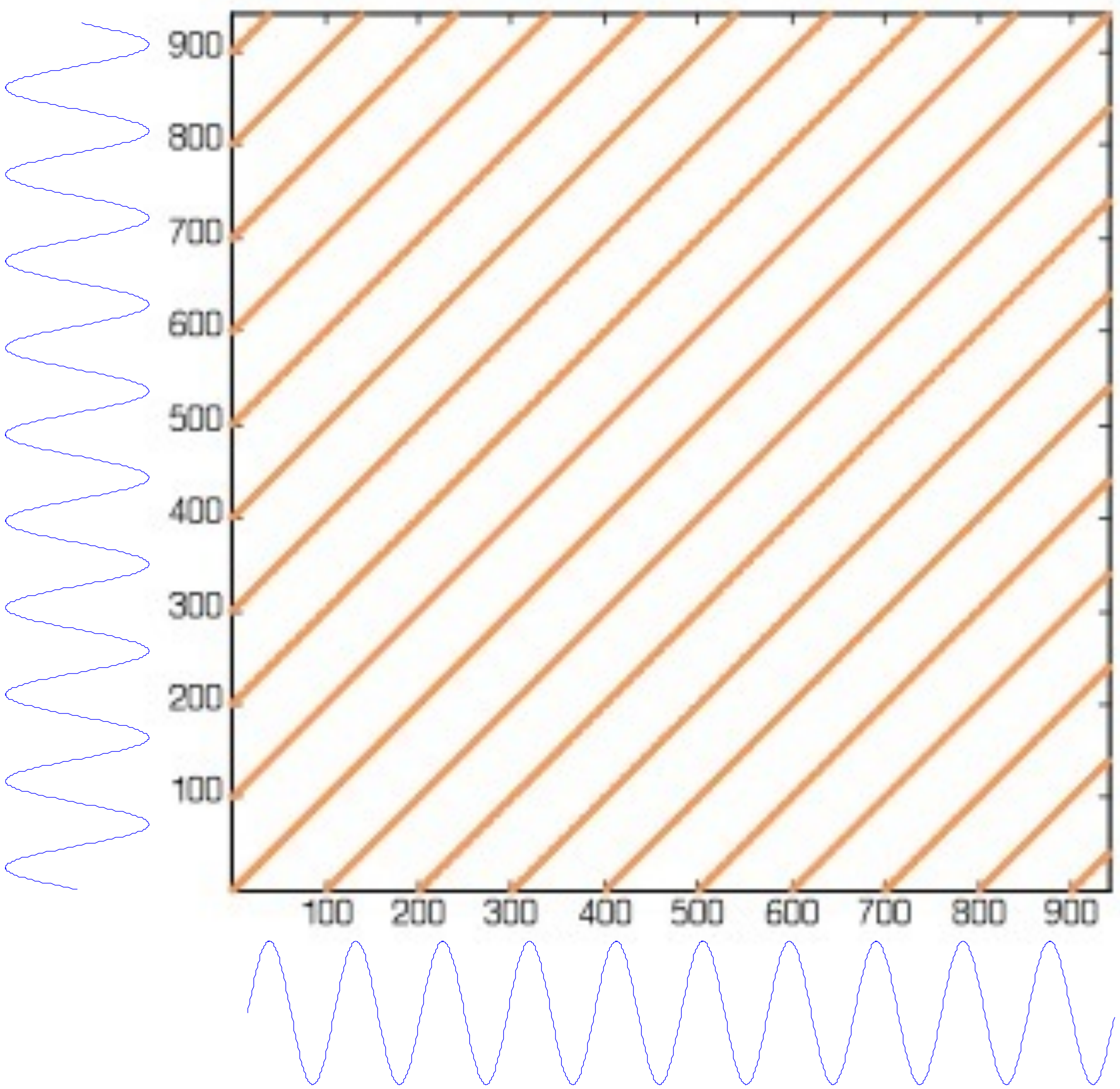
Will the current X,Y,Z coordinate (or a value within the radius) recur in the future?

Quantifying Recurrence

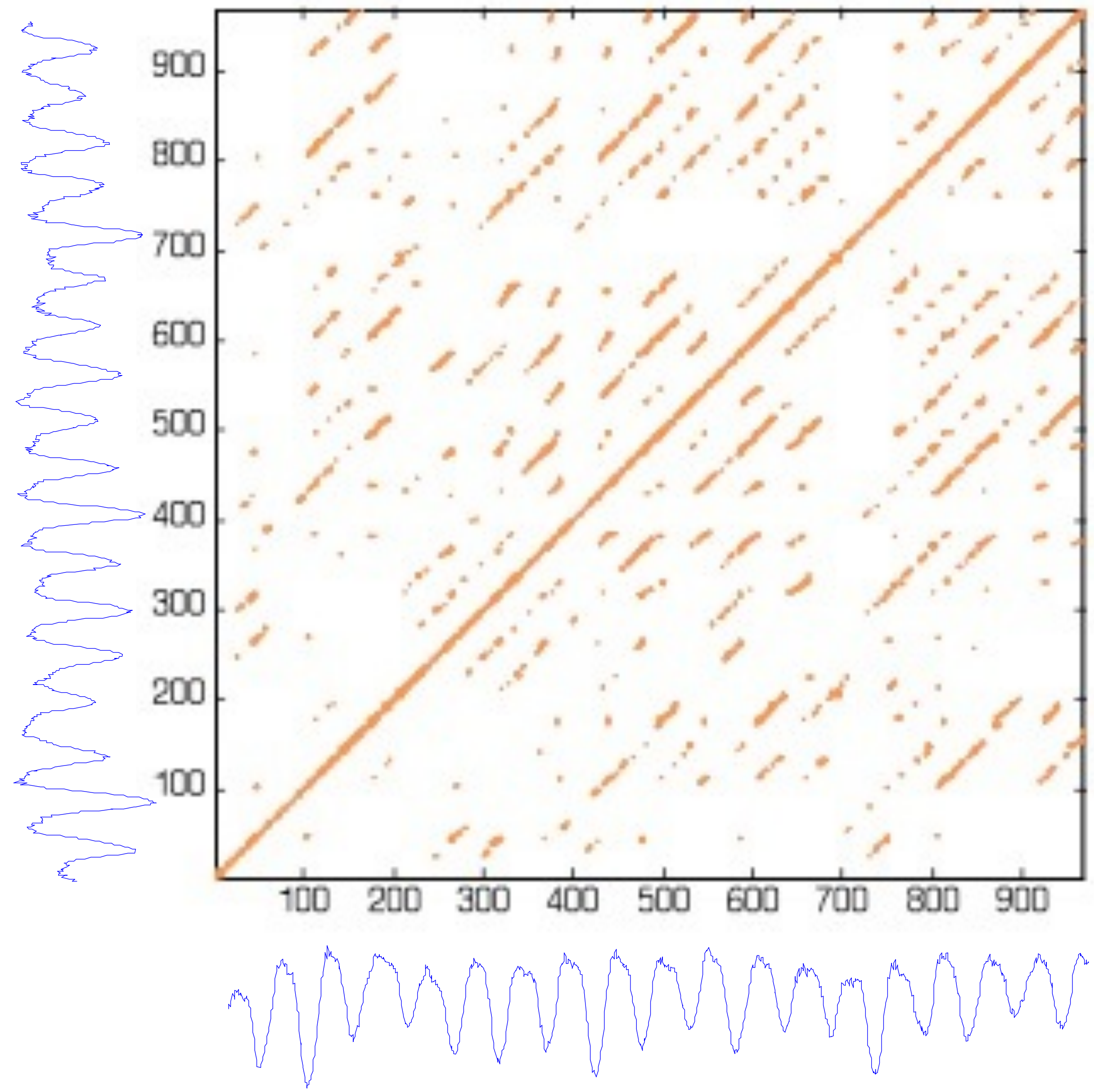
Shockley 2007

$$\%REC = \frac{\text{Number of recurrent points}}{\text{Total number of locations}} \times 100$$

Sine
%REC = 2.9



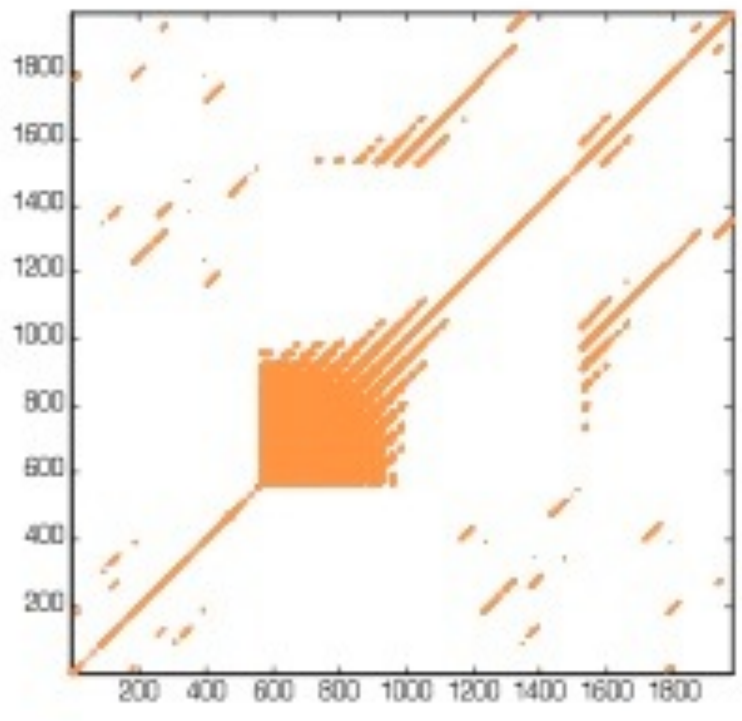
Limb oscillation to a metronome
%REC = .72



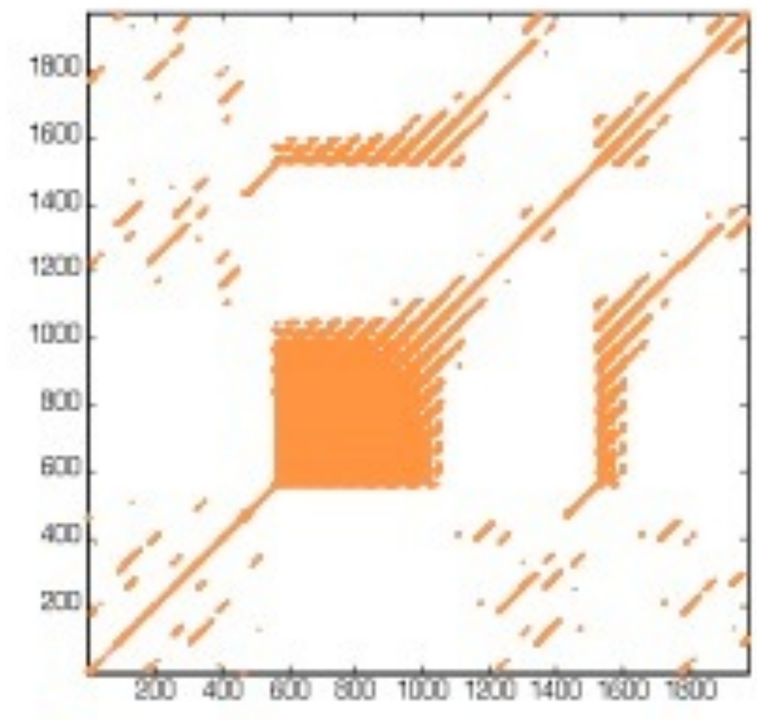
Note that %REC is the number of points in phase space that recur, relative to all possible points that could recur. It is influenced by the radius you choose!
When comparing groups or subjects: keep %REC constant.

Note how the recurrence plot changes with changes in radius

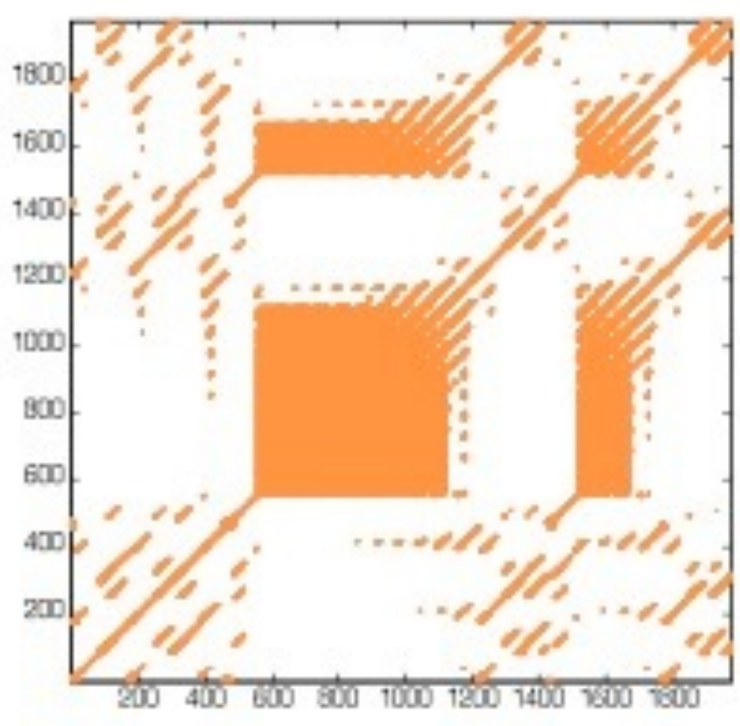
Radius = 3



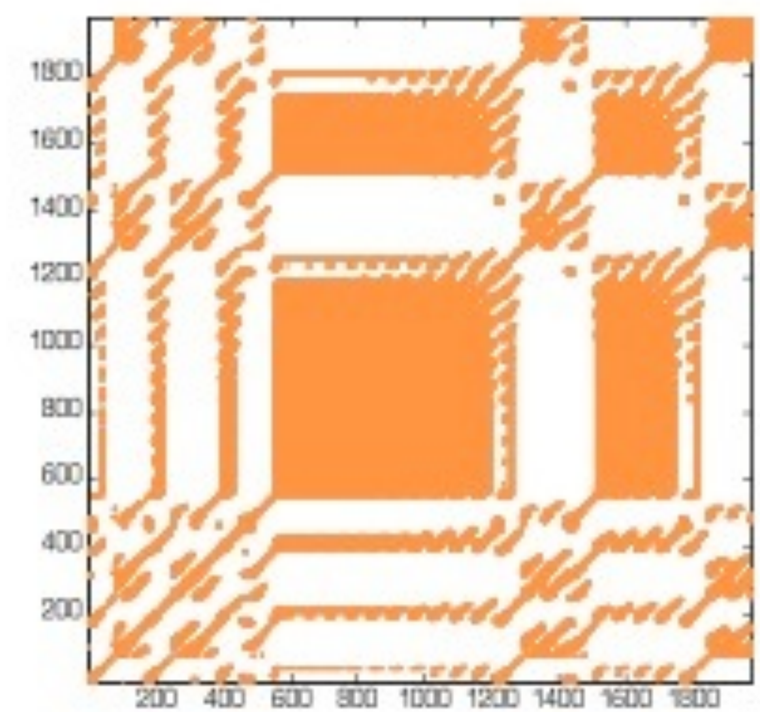
Radius = 5



Radius=10



Radius=20



Shockley 2007

Is there a prescription for picking your radius?

Adapted from Shockley 2007

%DETERMINISM

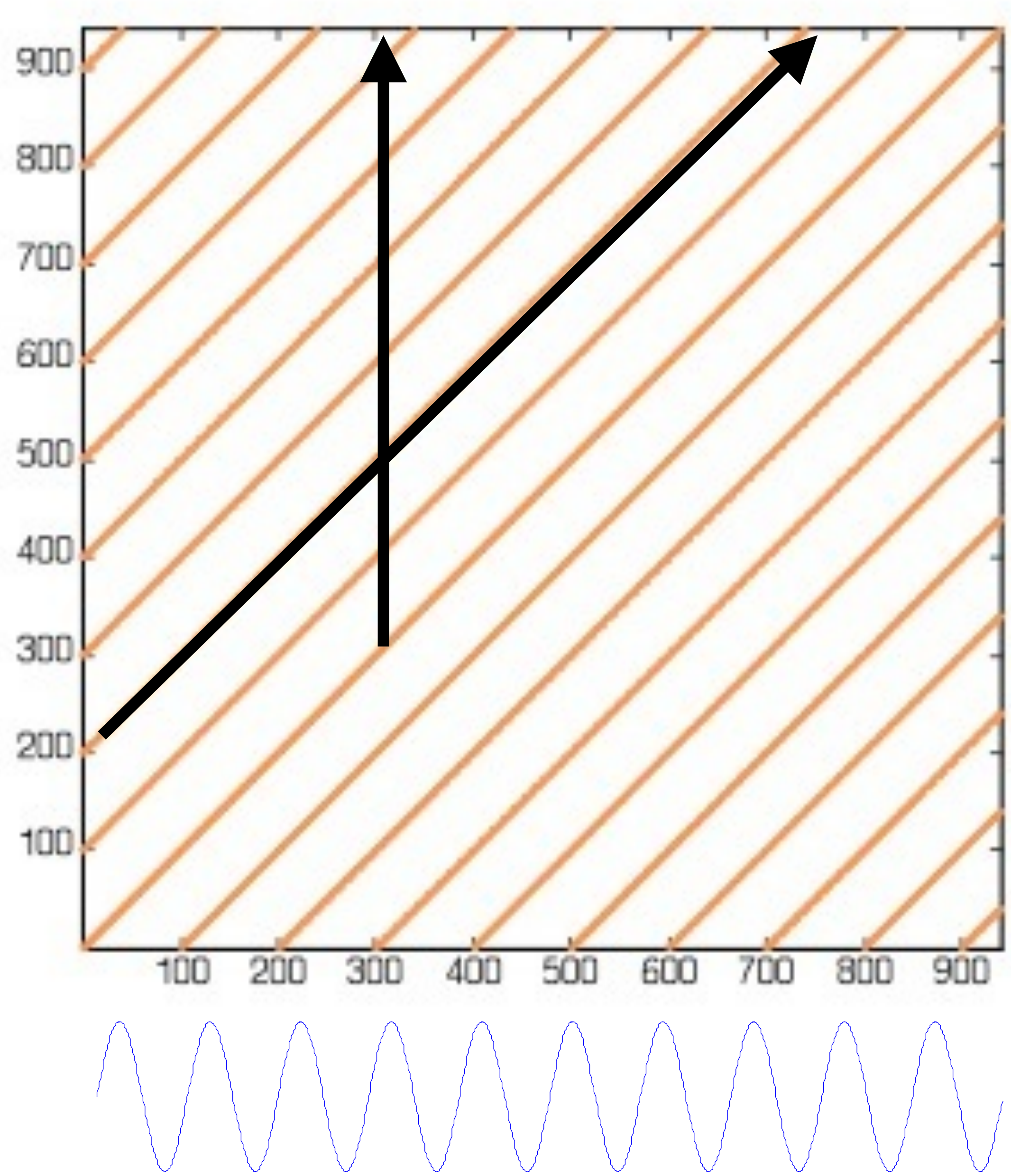
Indexes how “patterned” the data are.

Does the system return to the same region of phase space for a longer period of time?

$$\%DET = \frac{\text{Number of recurrent points forming diagonal line}}{\text{Total recurrent points}} \times 100$$

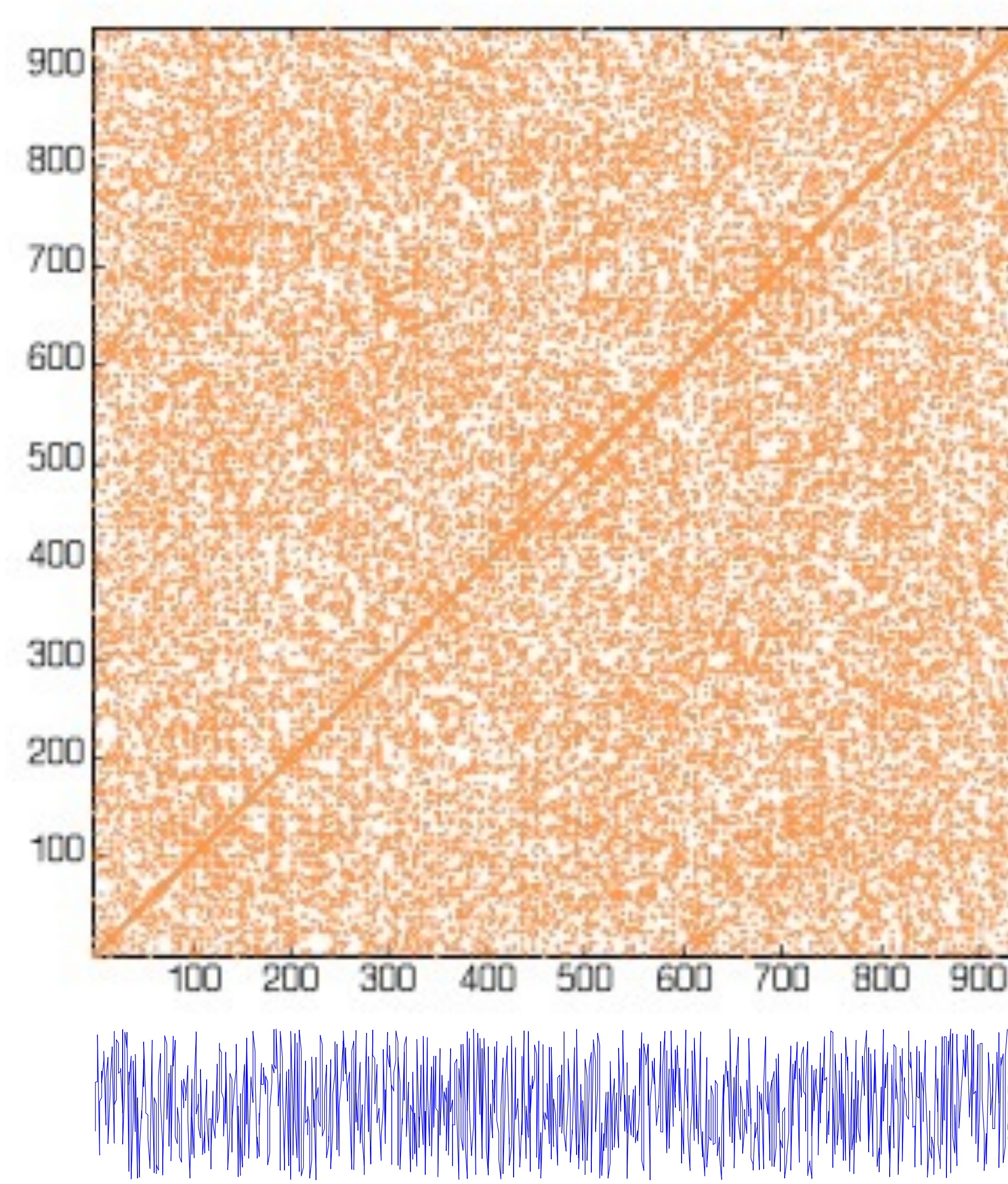
Sine

%REC = 2.9
%DET = 99.8



White Noise

%REC = 2.9
%DET = 5.4



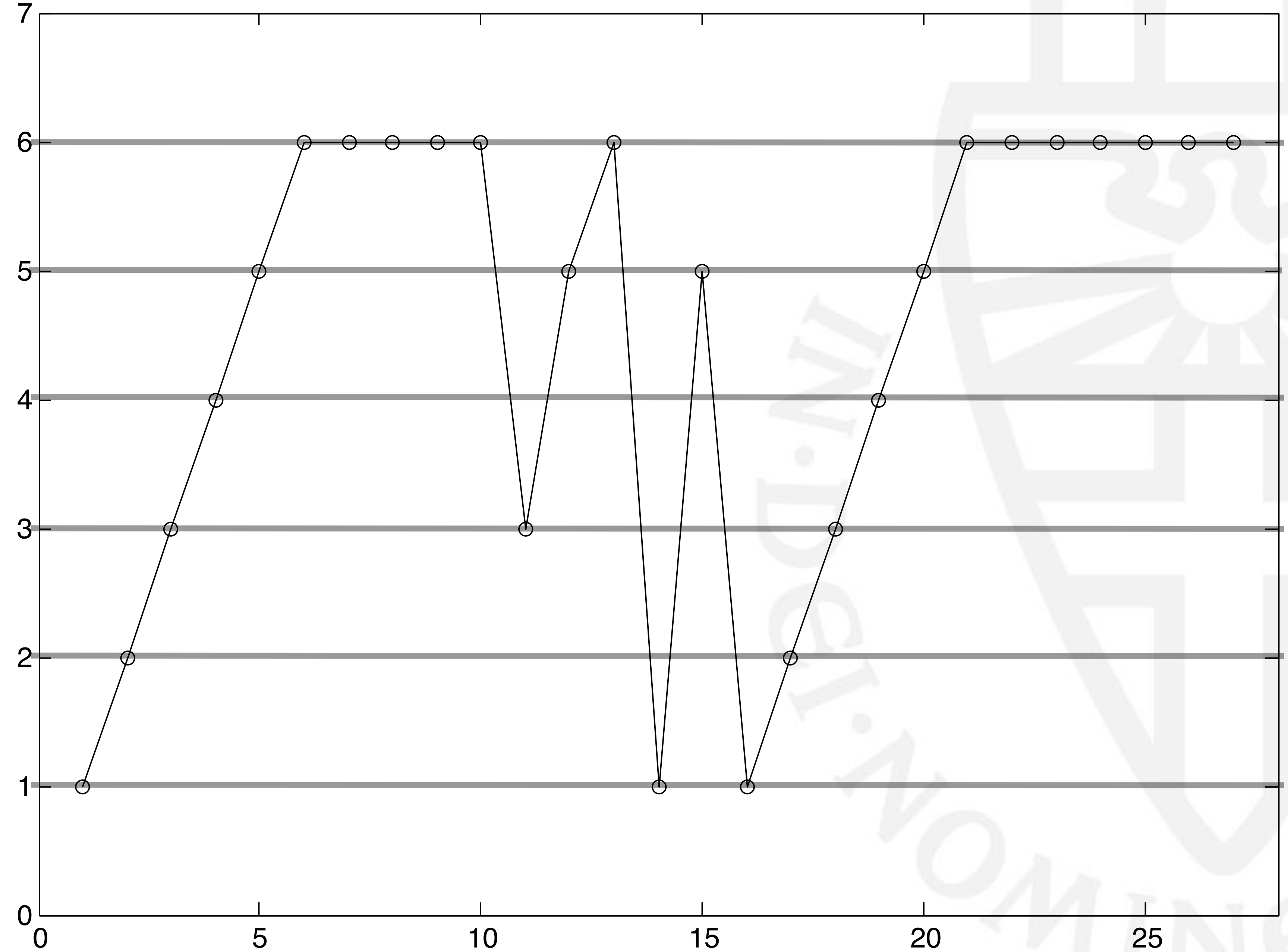
Diagonal lines signify recurring patterns...

not necessarily recurring values!

Let's take radius / threshold = 0

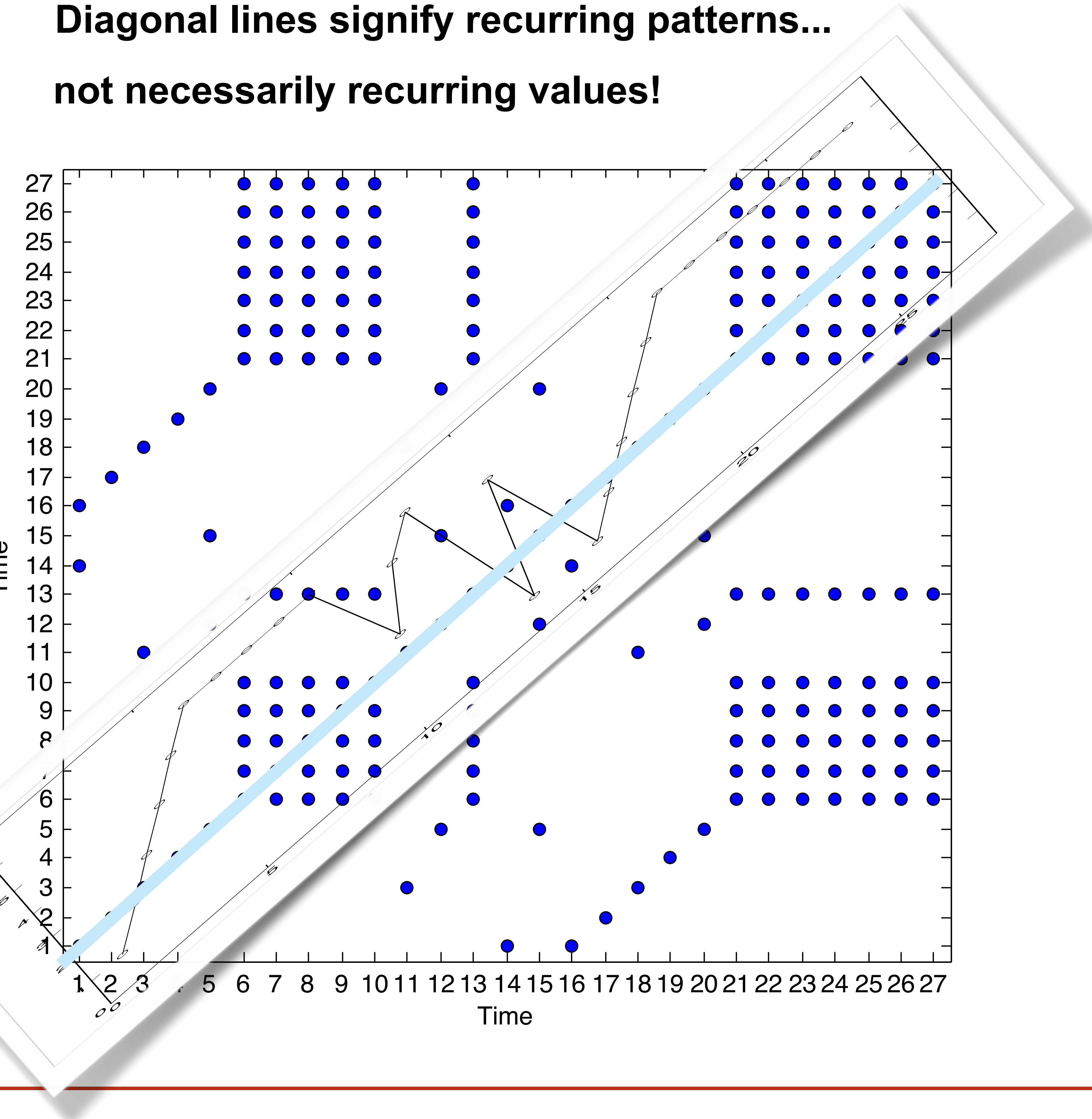
Looking at exact recurrence

1 dimensional state space (no embedding, just "raw" recurrence)



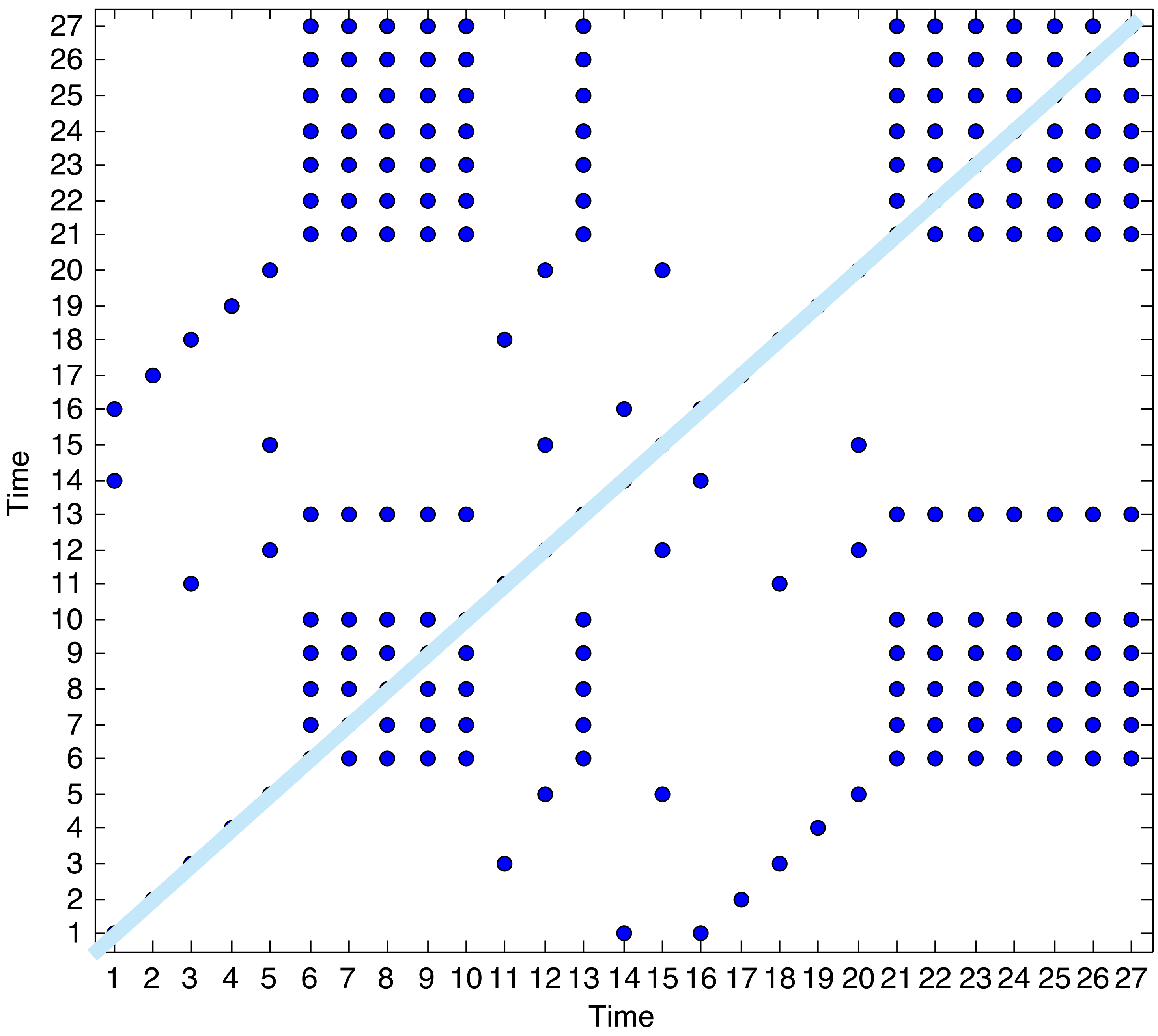
Diagonal lines signify recurring patterns...

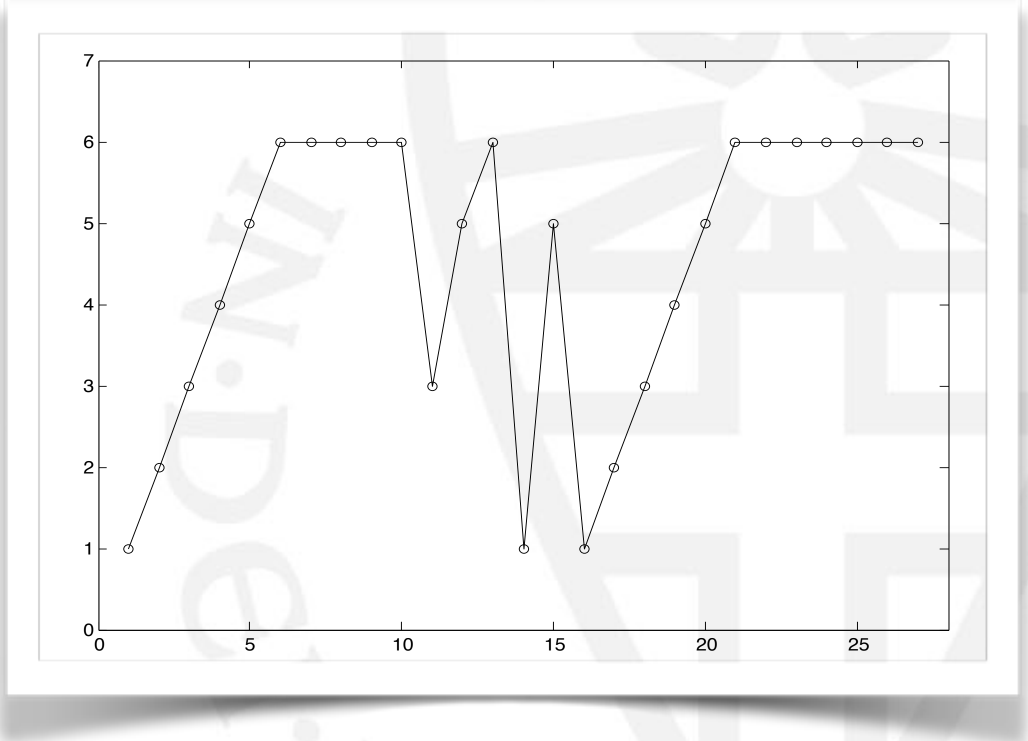
not necessarily recurring values!



Diagonal lines signify recurring patterns...

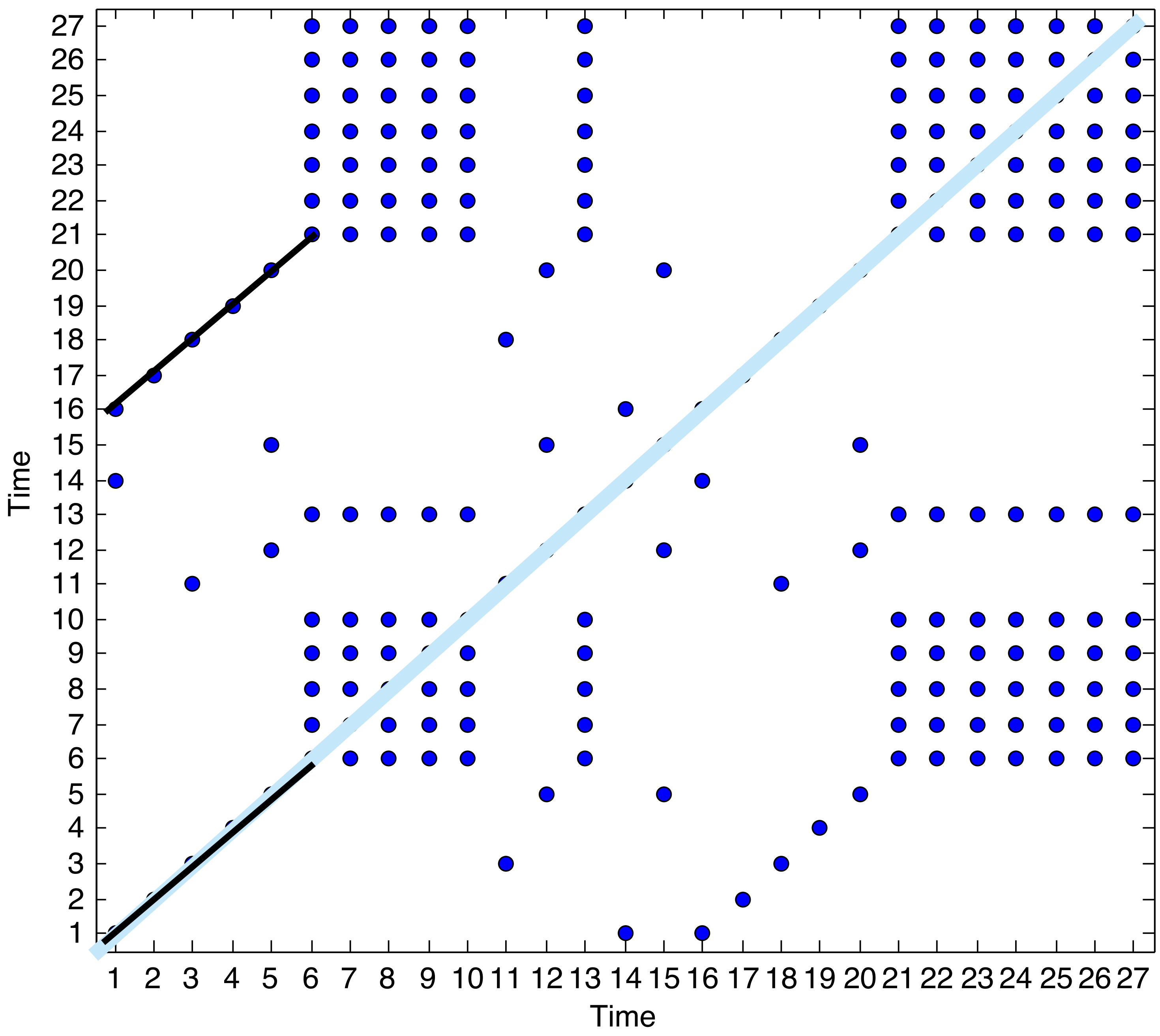
not necessarily recurring values!



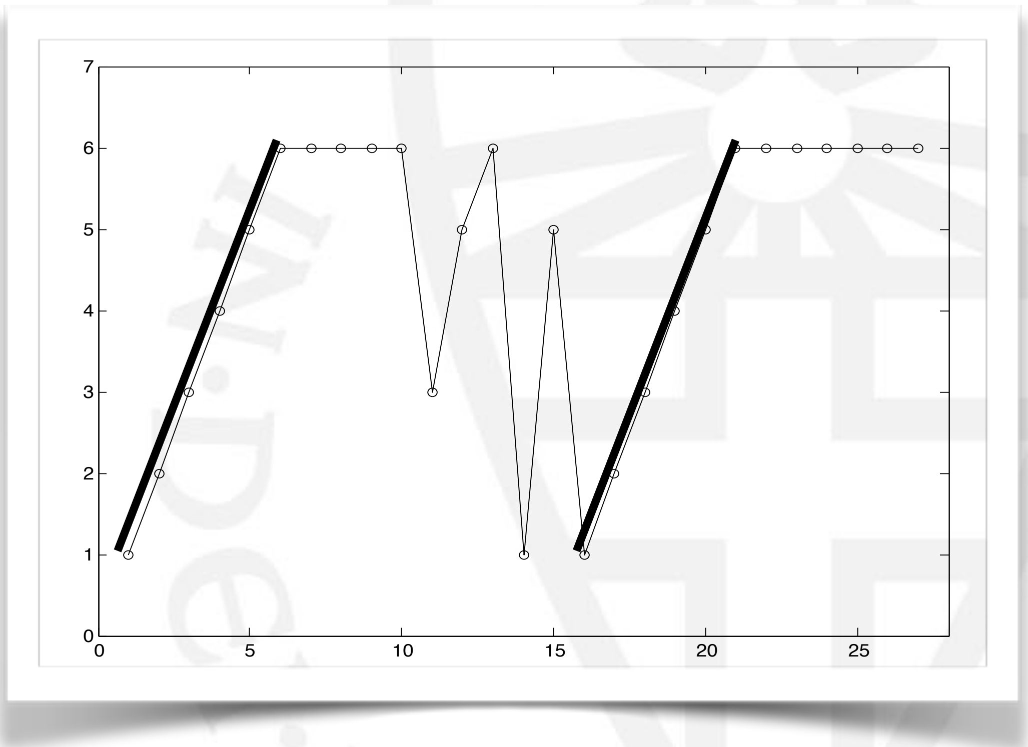


Diagonal lines signify recurring patterns...

not necessarily recurring values!

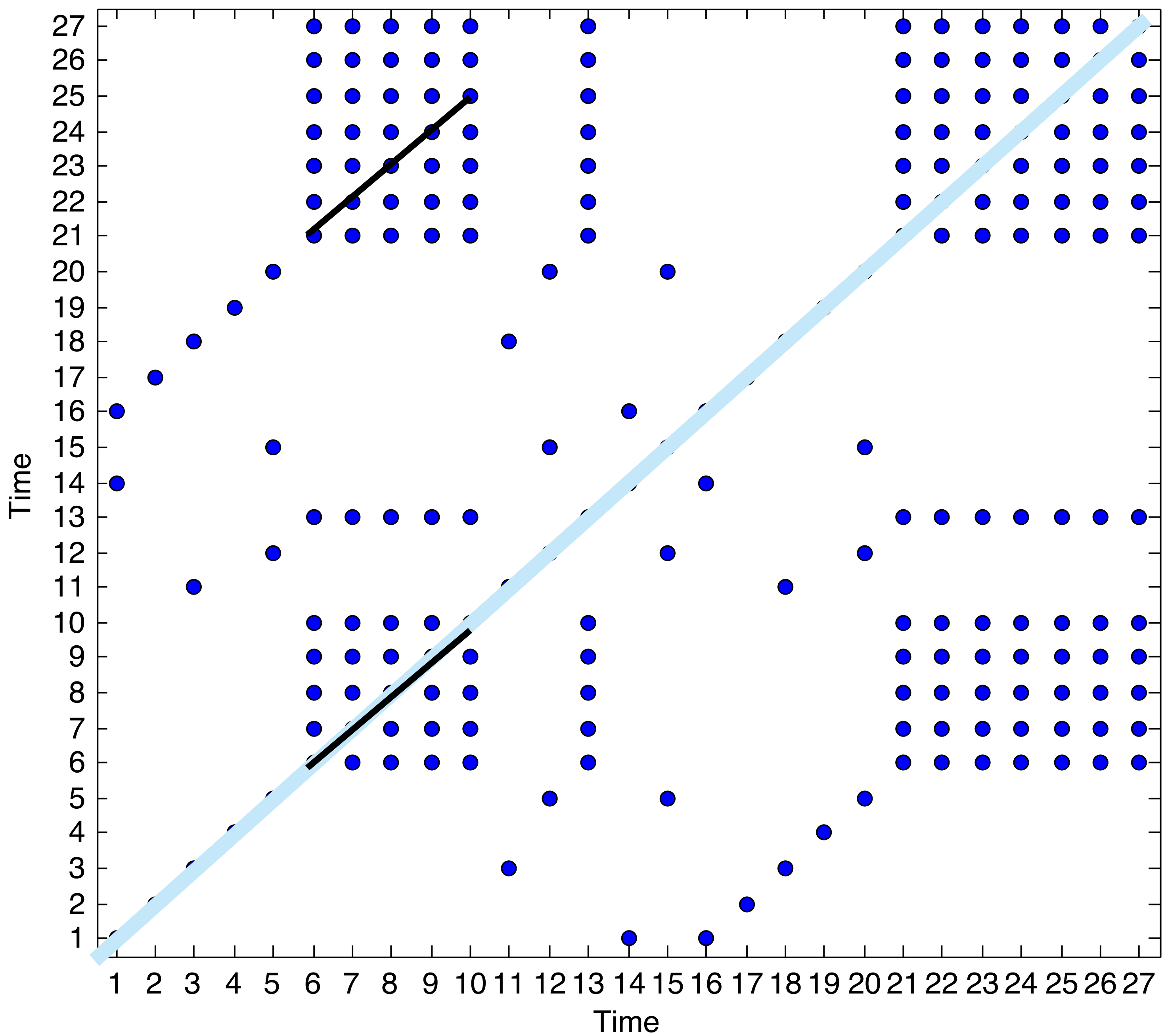


1-2-3-4-5-6 at t=1 recurs at t=16



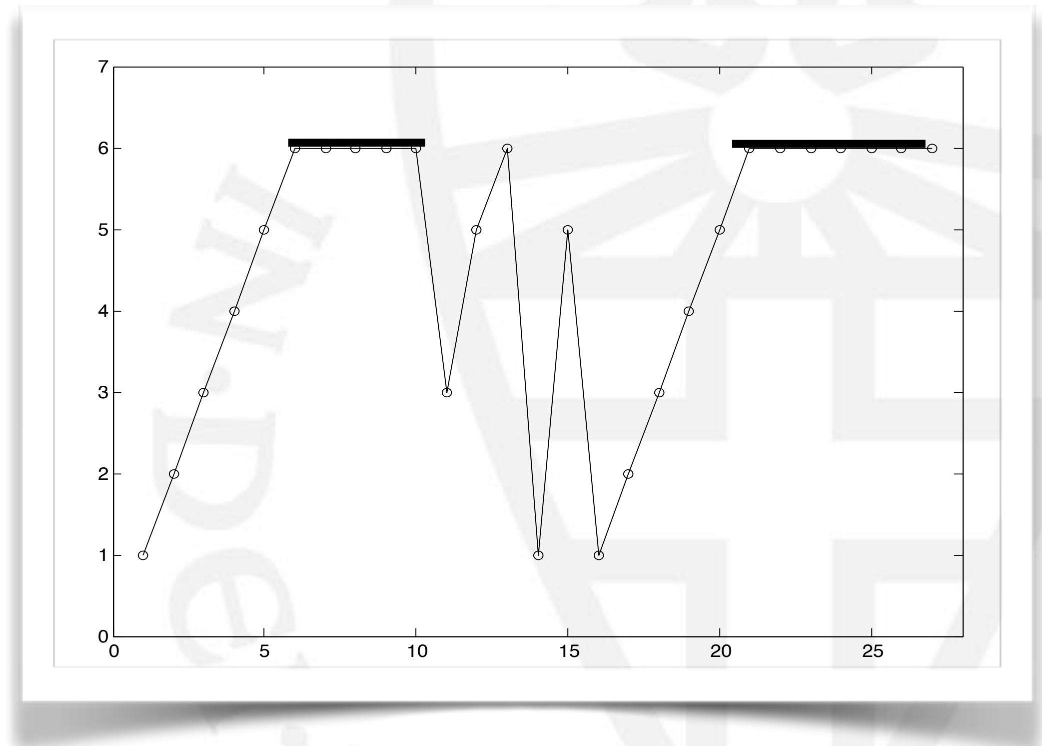
Diagonal lines signify recurring patterns...

not necessarily recurring values!



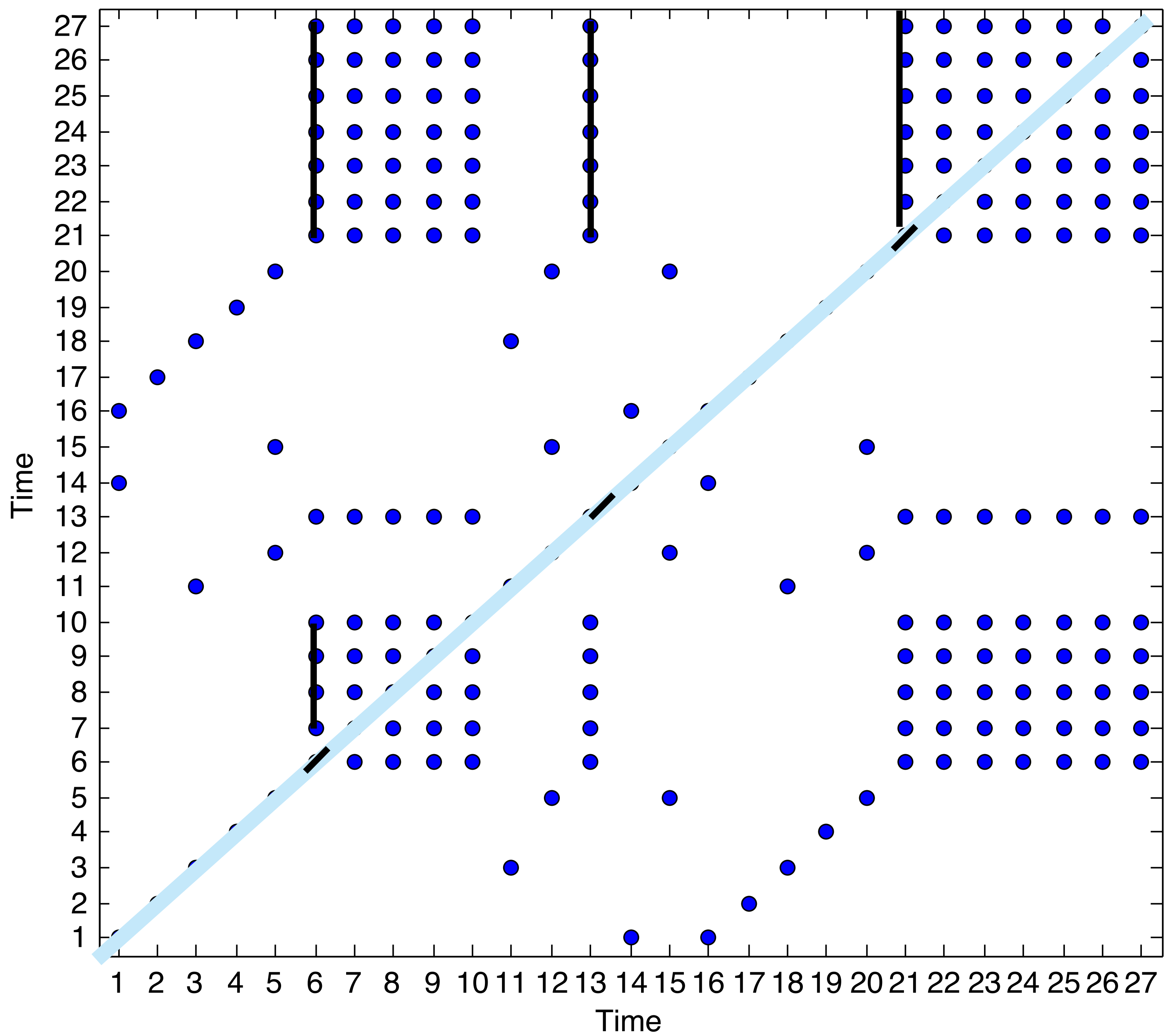
1-2-3-4-5-6 at t=1 *recurs* at t=16

6-6-6-6-6 at t=6 *recurs* at t=21



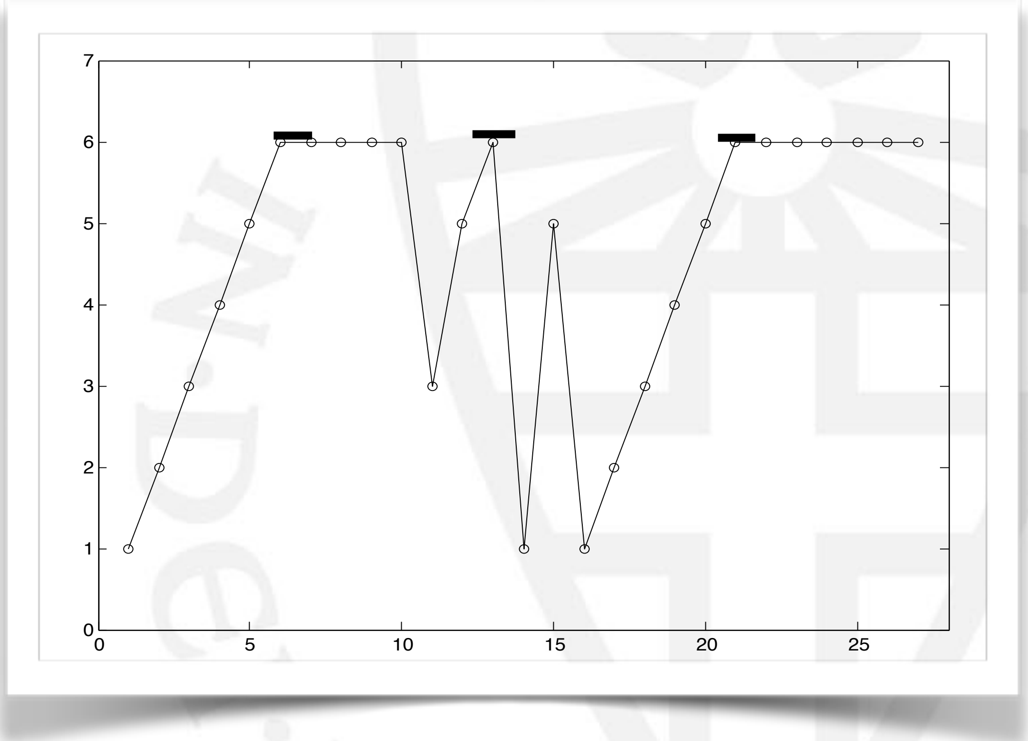
Diagonal lines signify recurring patterns...

not necessarily recurring values!



1-2-3-4-5-6 at t=1 *recurs* at t=16

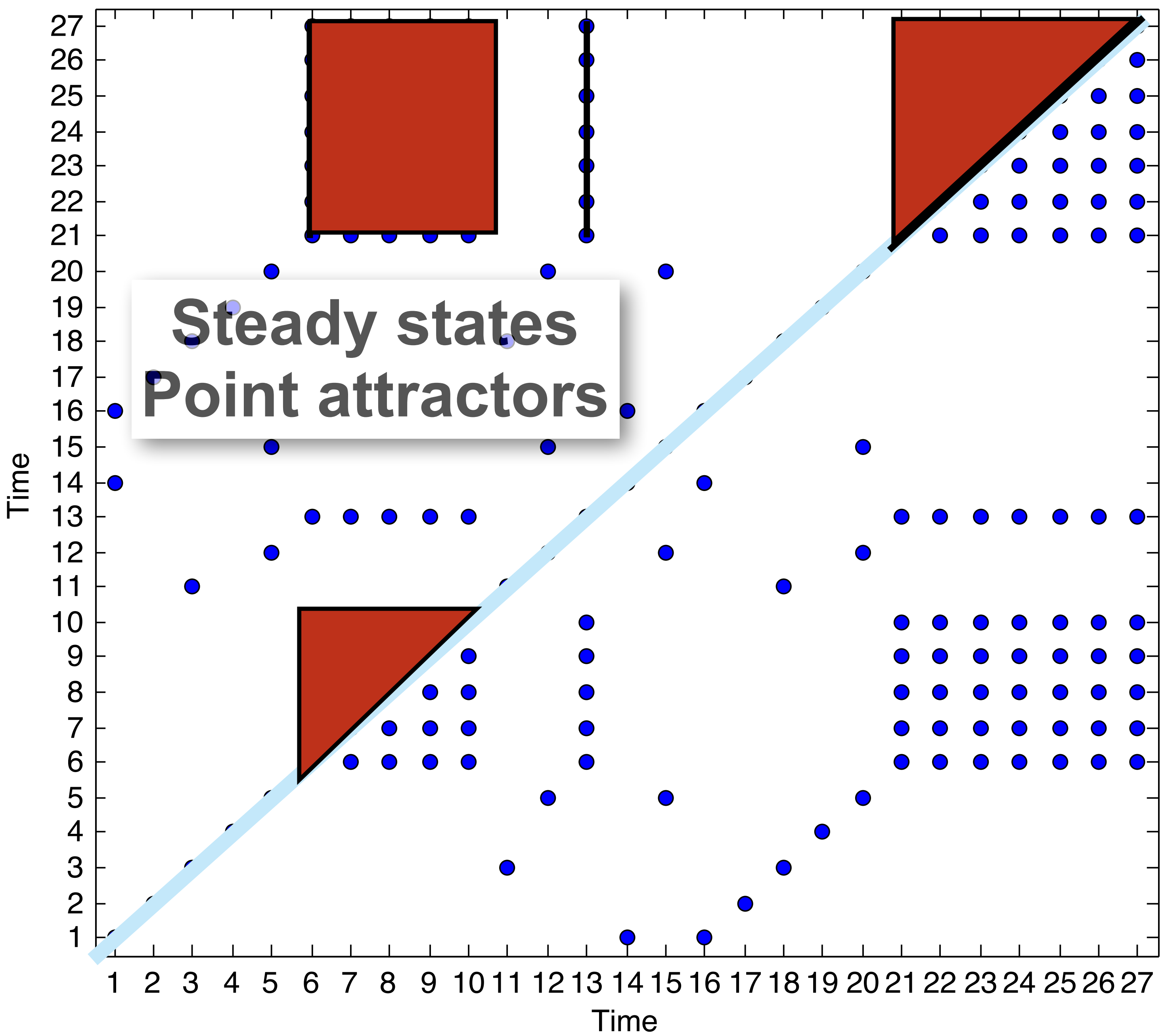
6-6-6-6-6 at t=6 *recurs* at t=21



Once the value 6 has occurred
The system will get "trapped"
in displaying 6 again in the future
(*Laminarity* = %points on vertical line, *Trapping Time* = aver. vertical line length)

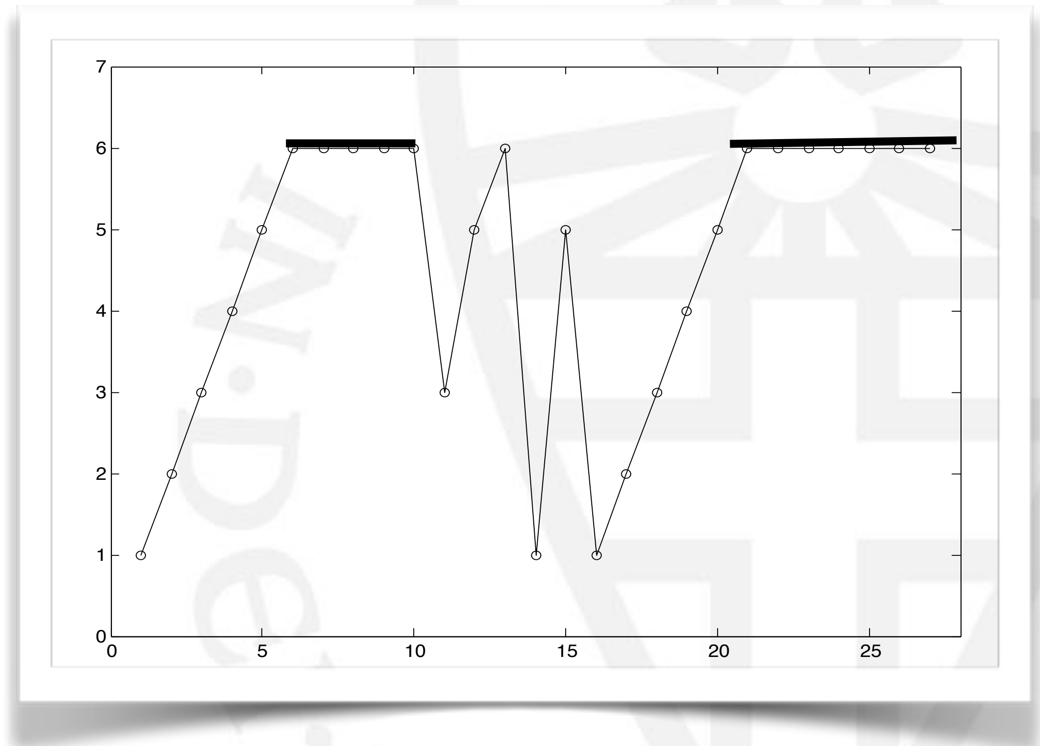
Diagonal lines signify recurring patterns...

not necessarily recurring values!



1-2-3-4-5-6 at t=1 *recurs* at t=16

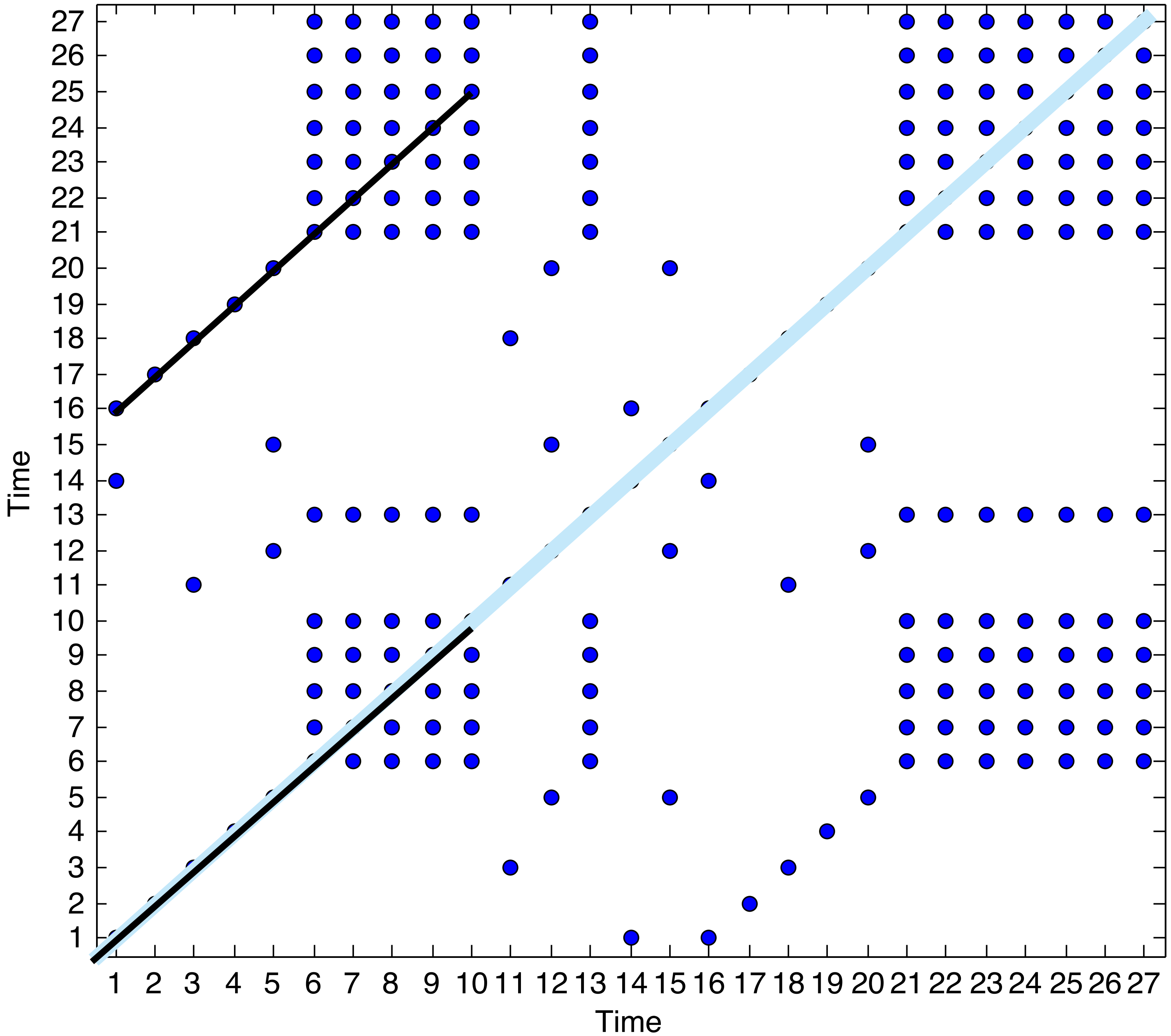
6-6-6-6-6 at t=6 *recurs* at t=21



Once the value 6 has occurred
The system will get "trapped"
in displaying 6 again in the future
(*Laminarity* = %points on vertical line, *Trapping Time* = aver. vertical line length)

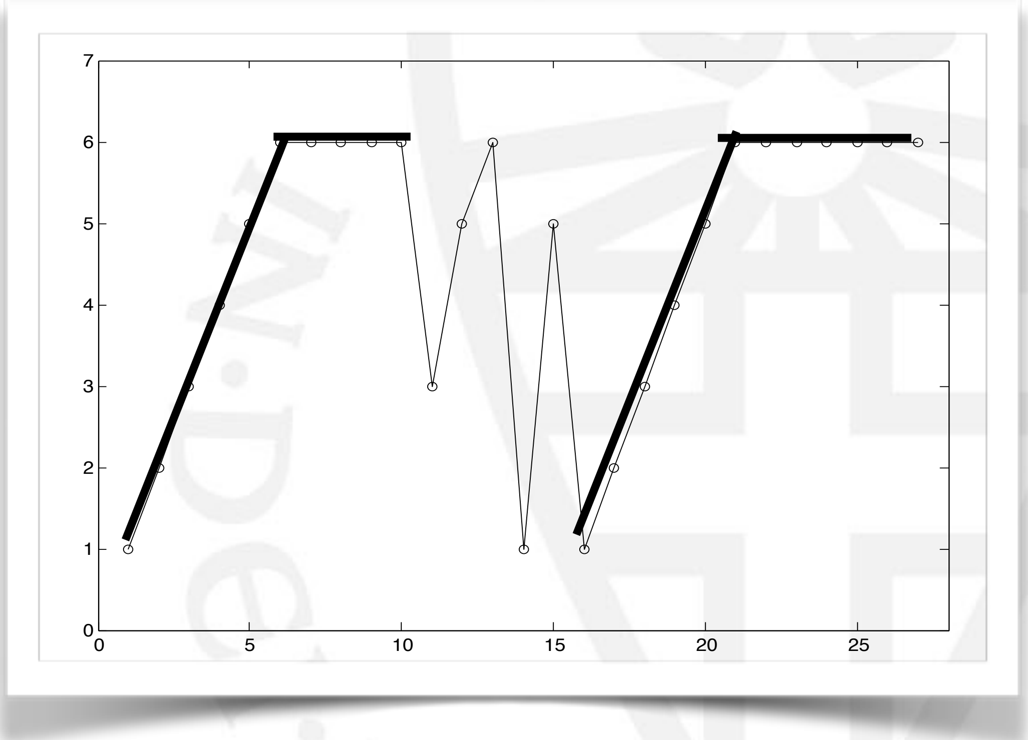
Diagonal lines signify recurring temporal patterns...

not necessarily recurring values!



1-2-3-4-5-6 at t=1 *recurs* at t=16

6-6-6-6-6 at t=6 *recurs* at t=21



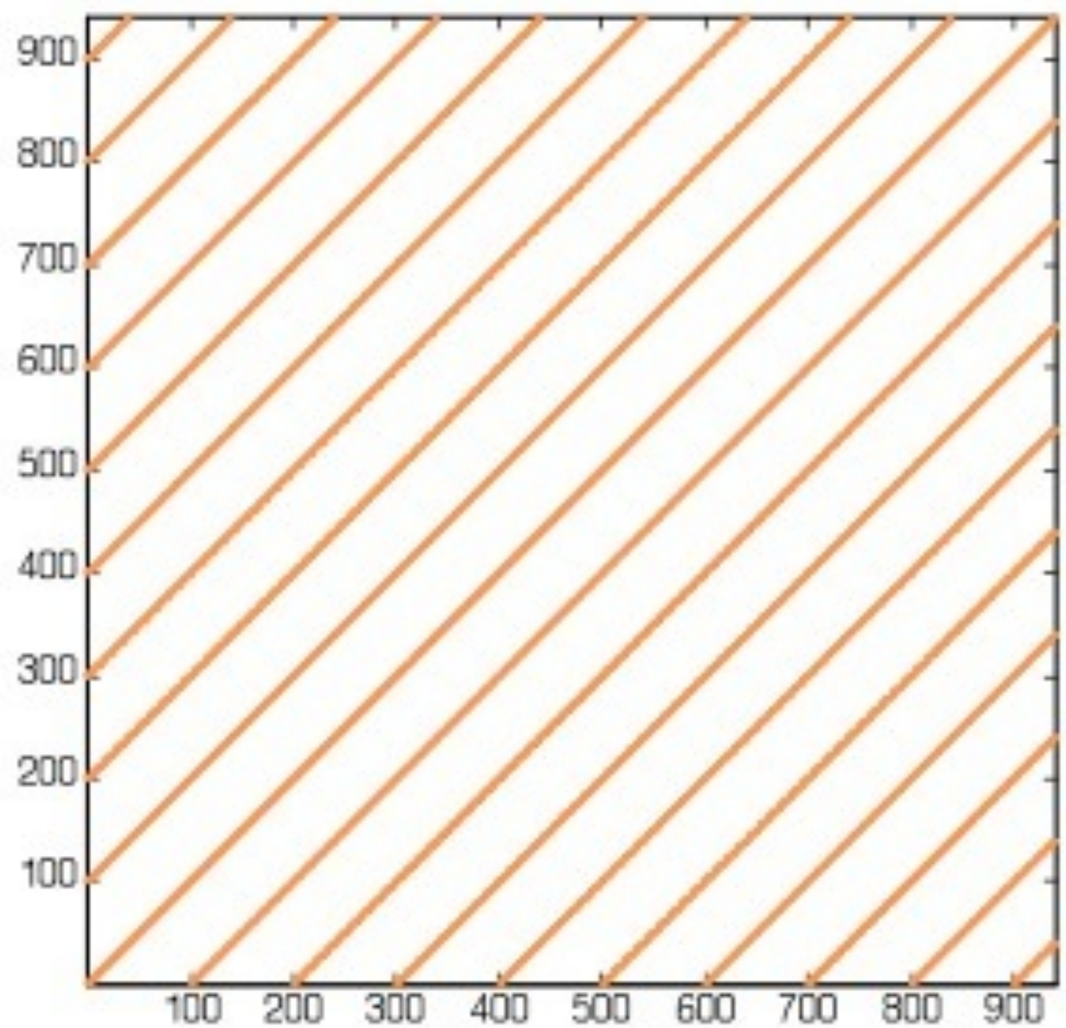
1-2-3-4-5-6-6-6-6-6 at t=1 *recurs* at t=16

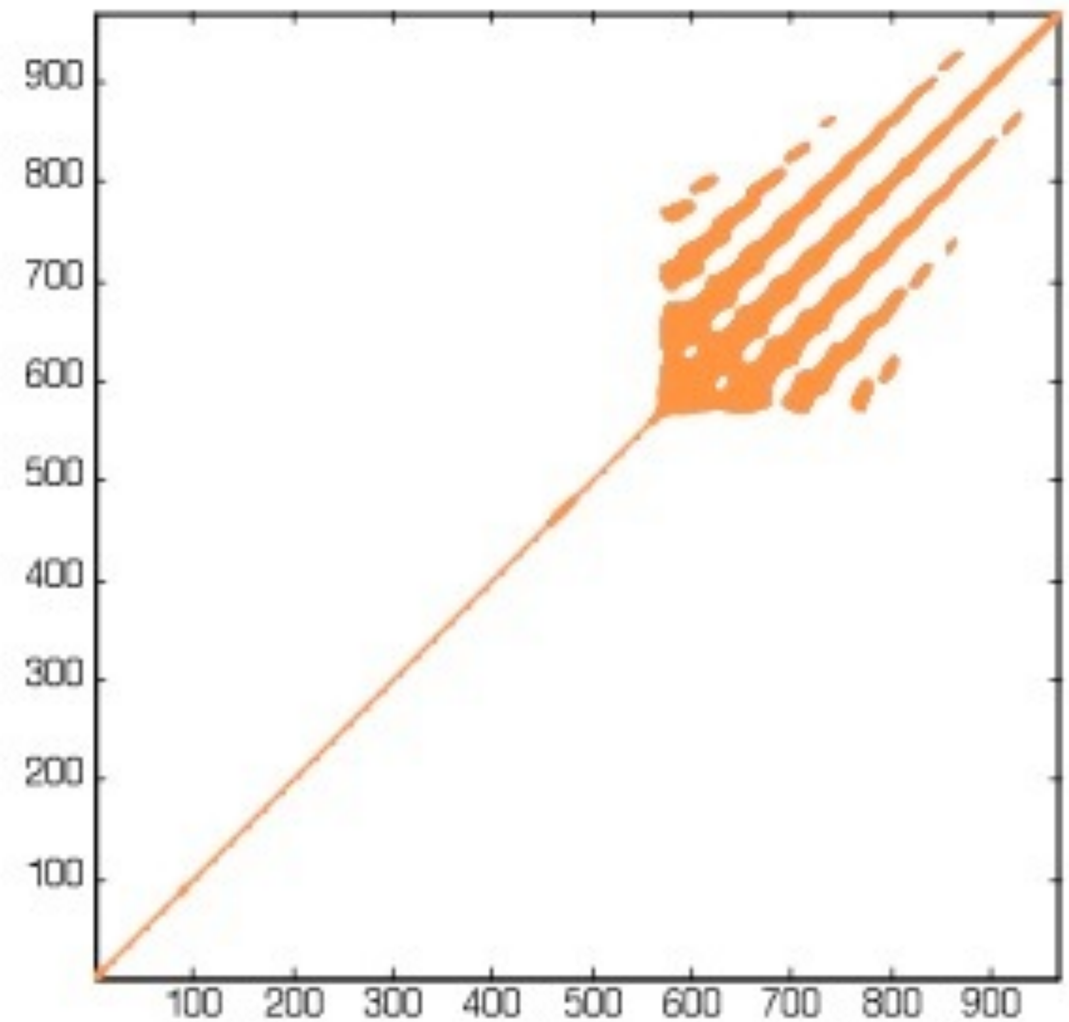
MAXLINE

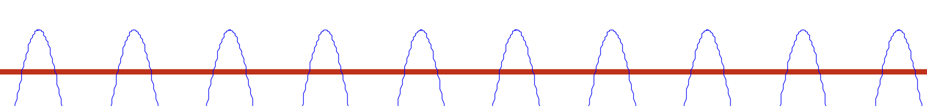
How long the system can maintain a recurring pattern ~ “Stability”

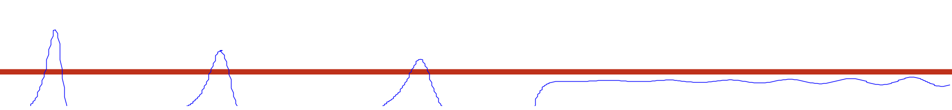
MAXLINE = The longest sequence of recurring points

Sine	Lorenz
%REC = 2.9	%REC = 2.9
MAXLINE = 938	MAXLINE = 410









1/maxline = Divergence (Thought to be an estimate of largest Lyapunov exponent)

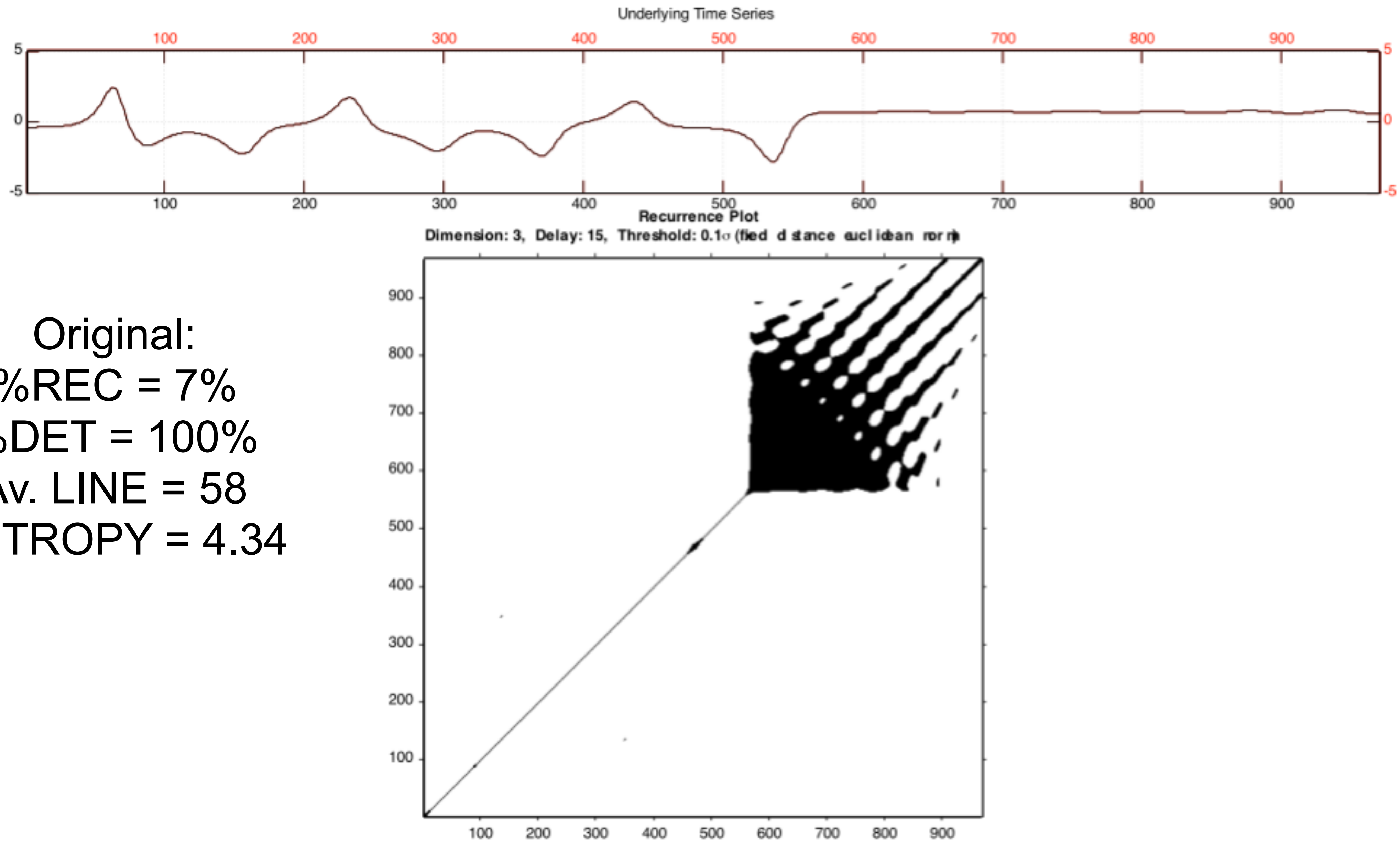
Shockley 2007

RQA measures

- %REC or RR (recurrence rate)
- %DET (is the data from a deterministic process or random?)
- MAXLINE (maximal diagonal line length)
- DIV (divergence, 1/maxline, suggested estimate of largest Lyapunov exponent)
- Average LINE (average diagonal line length)
- ENTROPY (complexity of deterministic structure)
- TREND (is the data stationary?)
- %LAM (laminarity, points on vertical lines, connected to Laminar phases)
- TT (Trapping Time, average length of vertical lines: How long the system stays in a specific state)

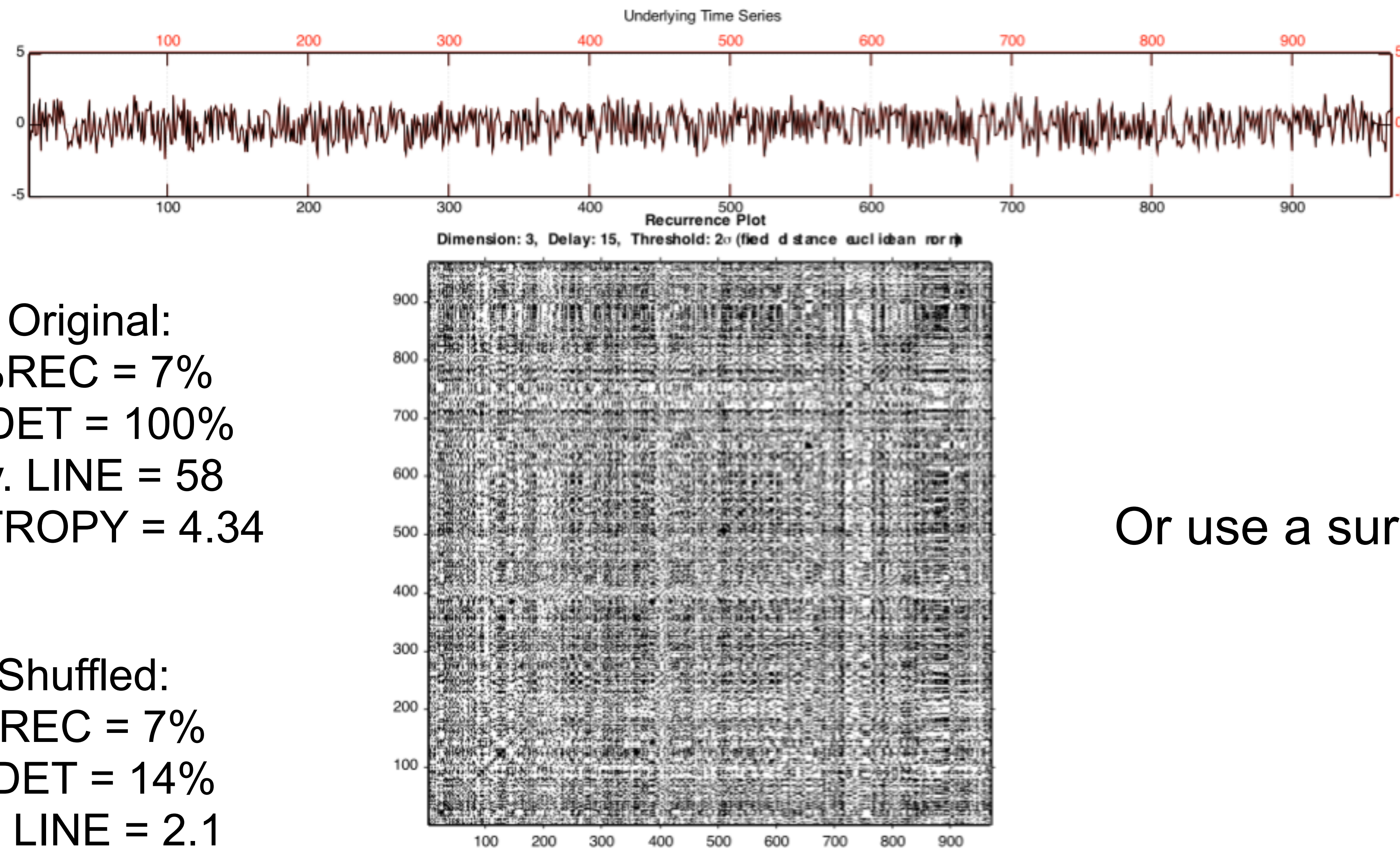
- Create your own…

How to decide these values have meaning?



Original:
%REC = 7%
%DET = 100%
Av. LINE = 58
ENTROPY = 4.34

How to decide these values have meaning?

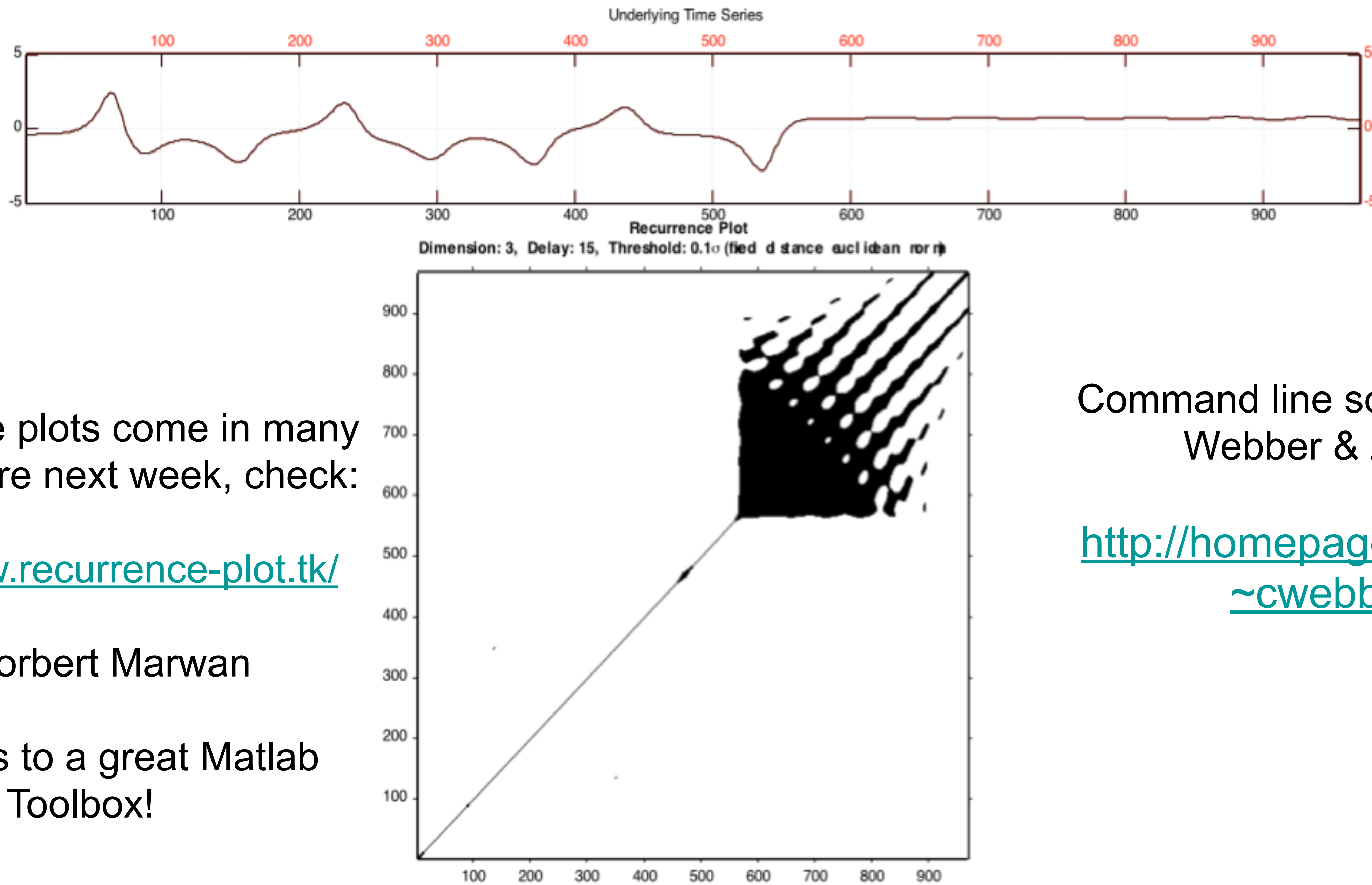


Original:
%REC = 7%
%DET = 100%
Av. LINE = 58
ENTROPY = 4.34

Shuffled:
%REC = 7%
%DET = 14%
Av. LINE = 2.1
ENTROPY = 0.25

Or use a surrogate

Recurrence Plots - Software



Recurrence plots come in many flavors, more next week, check:

http://www.recurrence-plot.tk/

By Norbert Marwan

Also links to a great Matlab Toolbox!

Command line software from Webber & Zbilut:

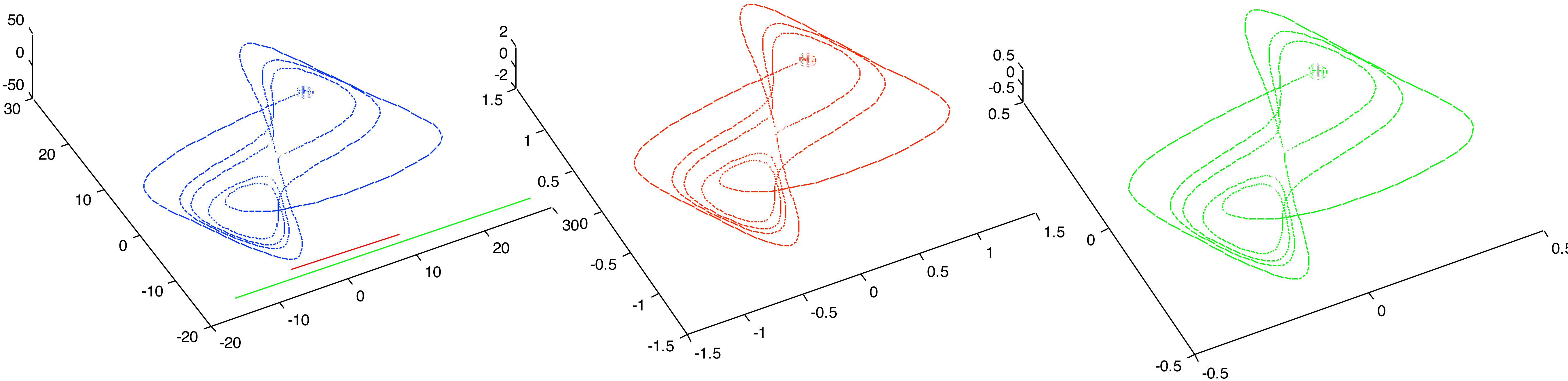
http://homepages.luc.edu/~cwebber/

Data Considerations

Generally it is a good idea to re-scale your data relative to either the mean or maximum distance separating points in reconstructed phase space.

This way data is scaled to itself which allows comparisons across data sets.

No Rescale | Mean Distance Rescale | Maximum Distance Rescale



Maximum distance re-scaling recommended

Webber, C.L., Jr., & Zbilut, J.P. (2005). Recurrence quantification analysis of nonlinear dynamical systems. In: *Tutorials in contemporary nonlinear methods for the behavioral sciences*, (Chapter 2, pp. 26-94), M.A. Riley, G. Van Orden, eds. Retrieved June 5, 2007 http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.pdf

General Recipe for Recurrence Quantification with toolbox:

- **Decide which lag to use:**

  Calculate the Average Mutual Information for a range of lags (*crqa_parameters*).
  Take the lag where AMI reaches its first minimum. This is the lag at which least is known about X(t+τ) given X(t), so we can create surrogate dimensions which give most new information about the system.

- Decide which **embedding dimension** to use:

  Calculate how many False Nearest Neighbours you loose by adding a dimension (*crqa_parameters*). Take the embedding dimension with the lowest % of nearest neighbours (or start with the dimension which gives the greatest decrease of neighbours).

- Decide which type of **rescaling** you want to use:

  Plot your timeseries: Lots of outliers? Use Mean Distance. Otherwise: Max Distance.
  Calculate the max distance in reconstructed phasespace, after lag and embedding are known using *max(recmat(y,emDim,emLag),* divide by this value.

- Decide which **radius / threshold** to use:
- Use *recmat_plot* to show unthresholded (without radius) plots use *crqa_radius* to find a radius

- Run **RQA** (*crqa_cl*) with these parameters!

- Compare to shuffled data (*shuffle, surrogates*)

Note that:

Recurrence values **will** change with changes in the parameters

The safest bet for behavioral data:

- Do recurrence calculations with one set of parameters for all of your data sets.

- Then, do this again with another set of parameters and make sure the overall results pattern the same way.

- Then, you can be sure that your results are not artefacts of your parameter selection