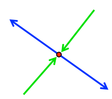
Lyapunov exponents

• nonlinear analogs of eigenvalues: one λ for each dimension



Lyapunov exponents: summary

• negative λ_i compress state space; positive λ_i stretch it

• nonlinear analogs of eigenvalues: *n* λ in an *n*-dimensional system

• they parametrize growth/shrinkage along the unstable and stable manifolds W^u and W^s

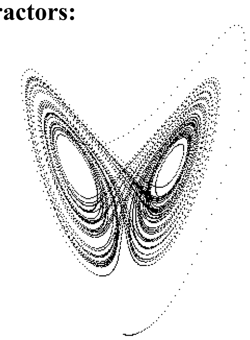
• $\Sigma\lambda_i < 0$ for dissipative systems

• λ_i are same for all ICs in one basin

• long-term average in definition; biggest one (λ_1) dominates as t → ∞

• *positive λ_1 is a signature of chaos*

"Strange" or chaotic attractors:

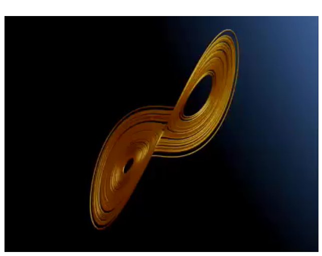


• exponential divergence of neighboring trajectories

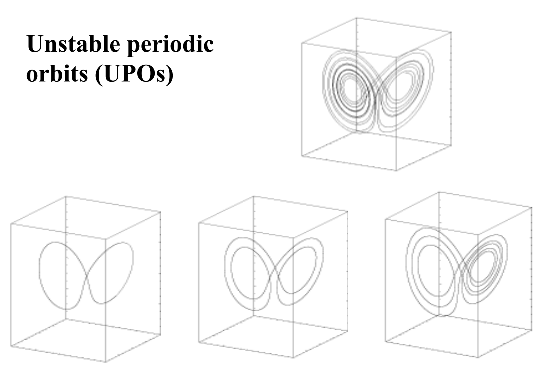
• *often* fractal

• covered densely by trajectories

• contain an infinite number of "unstable periodic orbits"…

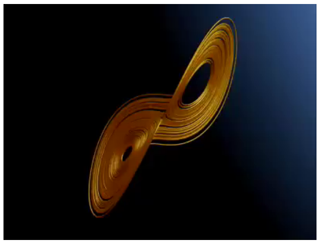


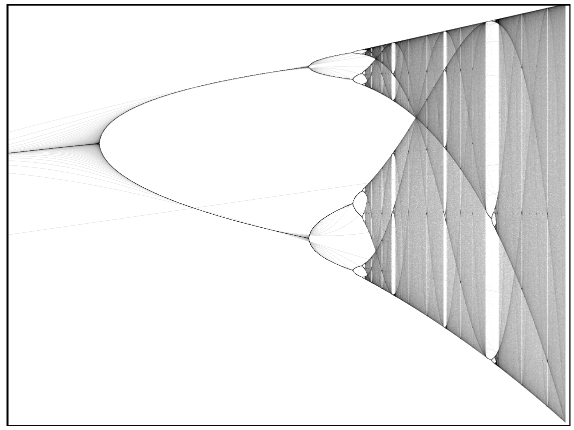
Unstable periodic orbits (UPOs)



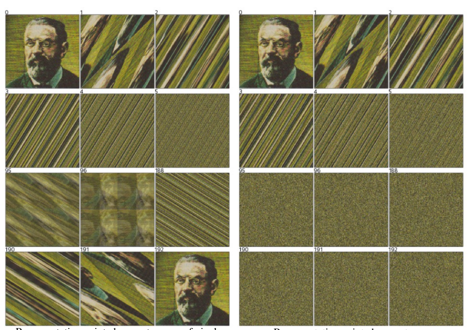
Bradley/Mantilla, *Chaos* **12**:596

“Attractor bones”…





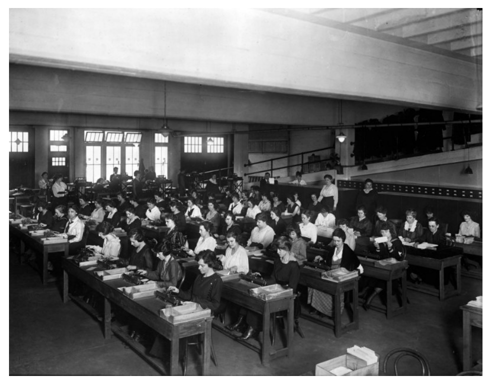
Poincare recurrence



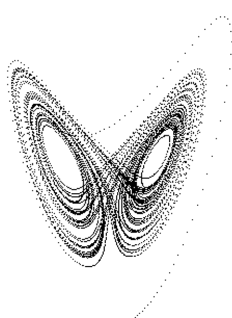
Representative point chosen at corner of pixel

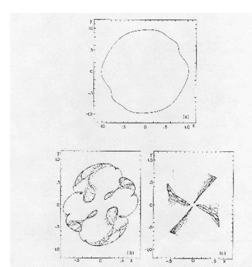
Representative point chosen at a random place in pixel

Crutchfield *et al*. *Chaos* **255**:46 and
`http://www.mpipks-dresden.mpg.de/mpi-doc/kantzgruppe/wiki/projects/Recurrence.html`



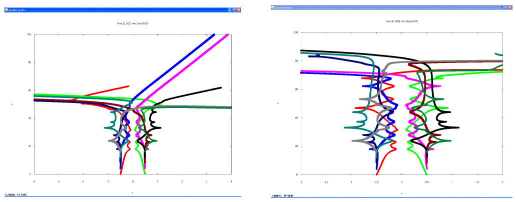
www.computerhistory





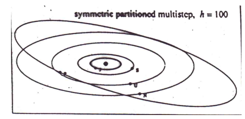
Different timestep

Lorenz, *Physica D* **35**:229

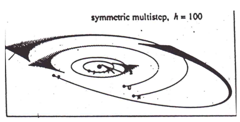


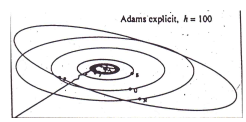
Different arithmetic

N. Ross Ph.D. thesis, Ucolorado, 2008



Different solver algorithm…





Moral: numerical methods can run amok in "interesting" ways…

- can cause distortions, bifurcations, etc.
- and these look a lot like *real, physical* dynamics…
- source: algorithms, arithmetic system, timestep, etc.
- Q: what could you do to diagnose whether your results included spurious numerical dynamics?

Moral: numerical methods can run amok in "interesting" ways…

- can cause distortions, bifurcations, etc.
- and these look a lot like *real, physical* dynamics…
- source: algorithms, arithmetic system, timestep, etc.
- Q: what could you do to diagnose whether your results included spurious numerical dynamics?
  - *change the timestep*
  - *change the method*
  - *change the arithmetic*

But beware machine ε…

Another important issue

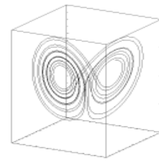
Many solvers, such as Matlab's `ode45`, are *adaptive*: they change the timestep and/or the method itself, on the fly, in order to correctly simulate the dynamics.

(The algorithms for this are interesting; we can talk about them offline.)

That means that the points that are output by tools like `ode45` are *not evenly spaced in time*. That can matter, depending on how you're using that solution…

So ODE solvers make mistakes.

…and chaotic systems are sensitively dependent on initial conditions….



…??!?

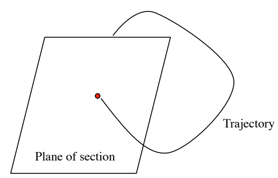
Shadowing lemma

Every* noise-added trajectory on a chaotic attractor is *shadowed* by a true trajectory.

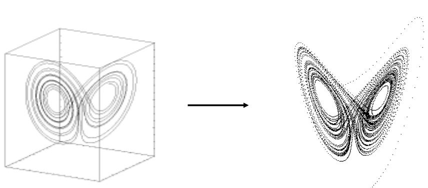
Important: this is for *state* noise, not *parameter* noise.

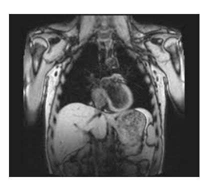
(*) Caveat: not if the noise bumps the trajectory out of the basin

Section



Not the same thing as a projection!

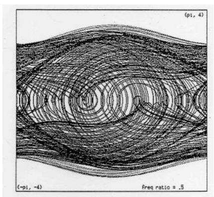


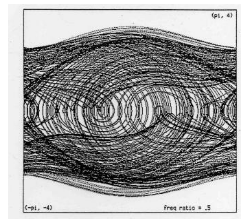


mri.radiology.uiowa.edu

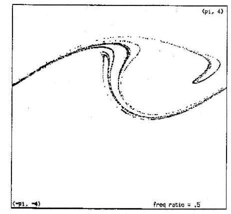
prixray.com

The driven damped pendulum

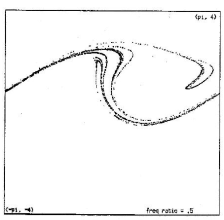




trajectory



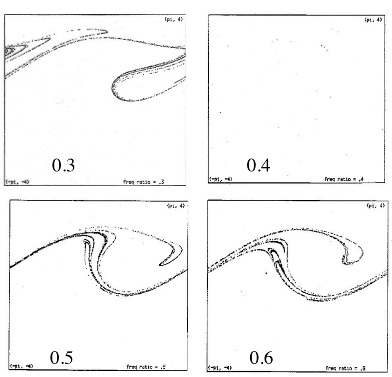
Poincare section



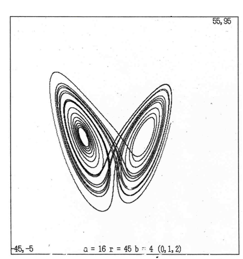
Time-slice sections of periodic orbits: some thought experiments

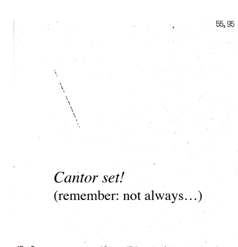
- pendulum rotating @ 1 Hz and strobe @ 1 Hz?
- pendulum rotating @ 1 Hz and strobe @ 2 Hz?
- pendulum rotating @ 1 Hz and strobe @ 3 Hz?
- pendulum rotating @ 1 Hz and strobe @ 1/2 Hz?
- pendulum rotating @ 1 Hz and strobe @ π Hz? (or some other irrational)

What bifurcations look like on a Poincare section

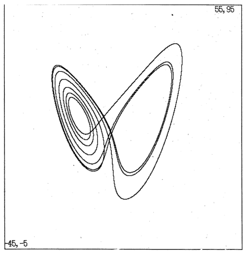


Spatial sections

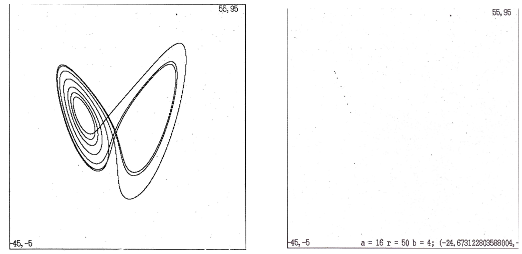




What about a section of a UPO?

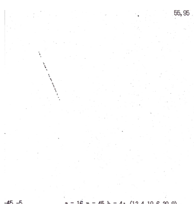


?



NB: the number of dots will depend on the plane of section.
(Issue: how to define "period"?)
Also, the place of section may totally miss the attractor…

Aside: finding UPOs

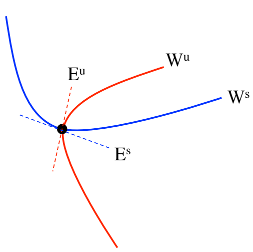


- Section
- Look for close returns
- Cluster
- Average
- See Gunaratne, So papers

How to compute sections?

- If you're slicing in state space: use the "inside-outside" function
- If you're slicing in *time*: use modulo on the timestamp
- See Parker & Chua for more details

λ_i and the un/stable manifolds (W^u and W^s)

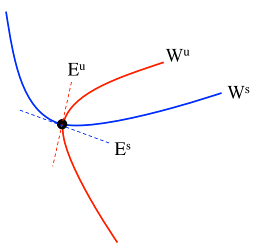


Aside: finding those un/stable manifolds

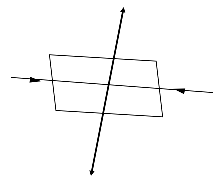
- Linearize the system
- Find the eigenvectors E^s and E^u
- Take a step along E^s; run time forwards
- Take a step along E^u; run time backwards
- See Osinga & Krauskopf paper for more details

Note: saddles are not the only possible landscape geometry around fixed points (they're just the most interesting ones!)

These λ_i & manifolds play a critical role in the control of chaos…



Local-linear control* of a hyperbolic point



* e.g., via pole placement

Lyapunov exponents, revisited:

- *n*-dim system has *n* λ_i; $\Sigma\lambda_i < 0$ for dissipative systems
- λ_i are same for all ICs in one basin
- negative λ_i compress state space along *stable manifolds*
- positive λ_i stretch it along *unstable manifolds*
- biggest one (λ_1) dominates as $t \rightarrow \infty$
- positive λ_1 is a signature of chaos
- calculating them:
  - From equations: eigenvalues of the variational matrix (see variational system notes on CSCI5446 course webpage, which you can access from Liz's homepage.)
  - From data: various creative algorithms…

Calculating λ (& other invariants) from data

- The bible: H. Kantz & T. Schreiber, *Nonlinear Time Series Analysis*
- Associated software: TISEAN
  www.mpipks-dresden.mpg.de/~tisean
- A recent review article: EB & H. Kantz, "Nonlinear Time Series Analysis Revisited," *CHAOS* **25:**097610 (2015)

TISEAN
Nonlinear Time Series Analysis

Rainer Hegger
Holger Kantz
Thomas Schreiber

Go to Version 3.0.1 (released March 2007)

Go to Version 2.1 (released December 2000)

TISEAN 3.0.1: Table of Contents

TISEAN 3.0.1

TISEAN home

Table of Contents

General Manual

Surrogates Manual

Tutorial

Usage Notes

Installation

Problems

All programs in alphabetical order

Sections

- Generating time series
- Linear tools
- Utilities
- Stationarity
- Embedding and Poincaré sections
- Prediction
- Noise reduction
- Dimension and entropy estimation
- Lyapunov exponents
- Surrogate data
- Spike trains
- XTisean
- Unsupported

Generating time series

A few routines are provided to generate test data from simple equations. Since there are powerfull packages (for ex
Helena Nusse and Jim Yorke) that can generate chaotic data, we have only included a minimal selection here.

Lyapunov exponents are an important means of quantification for unstable systems. They are however difficult to estimate from a time series. Unless low dimensional, high quality data is at hand, one should not attempt to calculate the full spectrum. Try to compute the maximal exponent first. The two implementations differ slightly. While lyap_k implements the formula by Kantz, lyap_r uses that by Rosenstein et al. which differs only in the definition of the neighbourhoods. We recommend to use the former version, lyap_k.

The estimation of Lyapunov exponents is also discussed in the introduction paper. A recent addition is a programm to compute finite time exponents which are not invariant but contain additional information.

Maximal exponent	lyap_k, lyap_r
Lyapunov spectrum	lyap_spec

Description of the program: lyap_k

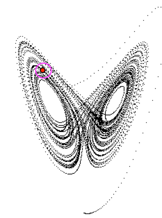
The program estimates the largest Lyapunov exponent of a given scalar data set using the algorithm of Kantz.

Usage:

lyap_k [Options]

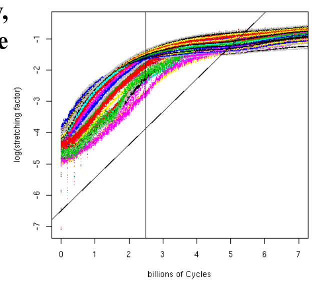
Everything not being a valid option will be interpreted as a potential datafile name. Given no datafile at all, means read stdin. Also - means stdin

Kantz's algorithm:



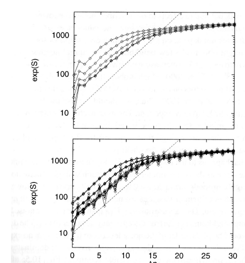
1. Choose point K •
2. Look at the points around it (ε neighborhood)
3. Measure how far they are from K
4. Average those distances
5. Watch how that average grows with time (Δn)
6. Take the log, normalize over time ➔ S(Δn)
7. Repeat for lots of points K and average the S(Δn)

If you're lucky, things look like this.



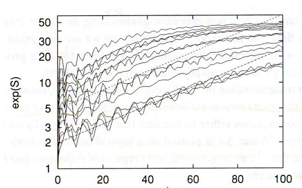
• The slope of the scaling region—iff one exists—is the λ

Or this:



This is fig 5.3 in Kantz & Schreiber

If you're not lucky:

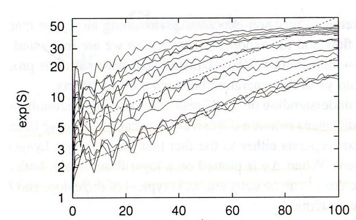


• The slope of the scaling region—iff one exists—is the λ

⬆

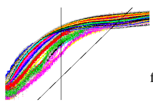
This is fig 5.4 in Kantz & Schreiber

What do you think those oscillations might be?



Calculating λ (& other invariants) from data

Be careful! All algorithms for computing these things have lots of knobs and their results are incredibly sensitive to their values!



Different colors on that plot from before = different settings for one of those knobs

Option	Description	Default
-l#	number of data to be used	whole file
-x#	number of lines to be ignored	0
-c#	column to be read	1
-M#	maximal embedding dimension to use	2
-m#	minimal embedding dimension to use	2
-d#	delay to use	1
-r#	minimal length scale to search neighbors	(data interval)/1000
-R#	maximal length scale to search neighbors	(data interval)/100
-##	number of length scales to use	5
-n#	number of reference points to use	all
-s#	number of iterations in time	50
-t#	'theiler window'	0
-o#	output file name	without file name: 'datafile'.lyap (or stdin.lyap if the data were read from stdin)
-V#	verbosity level 0: only panic messages 1: add input/output messages 2: add statistics for each iteration	3
-h	show these options	none

Description of the Output:

For each embedding dimension and each length scale the file contains a block of data consisting of 3 columns

1. The number of the iteration
2. The logarithm of the stretching factor (the slope is the Lyapunov exponent if it is a straight line)
3. The number of points for which a neighborhood with enough points was found

Calculating λ (& other invariants) from data

• *Be careful! All algorithms for computing these things have lots of knobs and their results are incredibly sensitive to their values!*

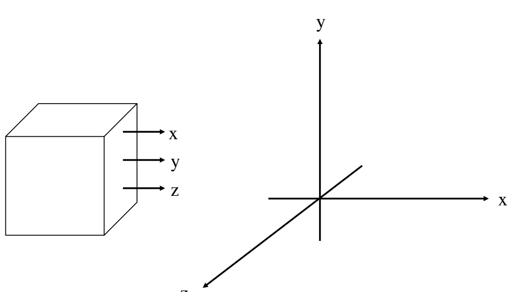
• Use your dynamics knowledge to understand & use those knobs intelligently

• *Look* at the results plots. For example, do not blindly fit a regression line to something that has no scaling region (this is a good idea in general, of course)

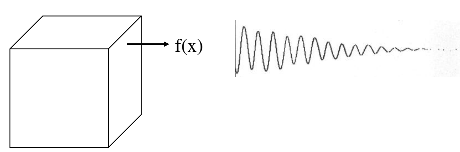
Fractal dimension:

- Capacity
- Box counting
- Correlation (`d2` in TISEAN)
- Lots of others:
  - Kth nearest neighbor
  - Similarity
  - Information
  - Lyapunov
  - …
- See Chapter 6 and §11.3 of Kantz & Schreiber

We've been assuming that we can measure all the state variables…

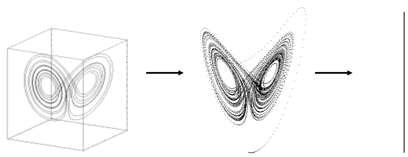


But often you can't.



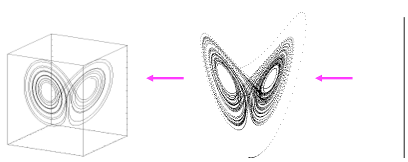
- Rarely do you even *know* what they are.
- Even if you did, you might not be able to *measure* all of them.
- And even if you could, doing so might *change the dynamics*…

How to undo a projection?



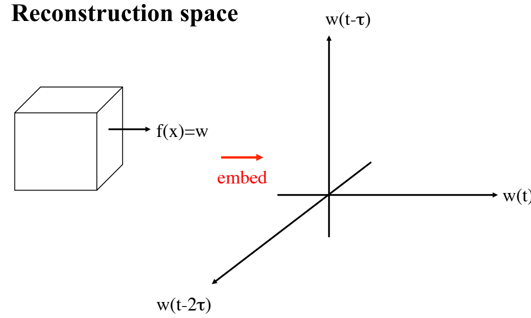
Delay-coordinate embedding

"reinflate" that squashed data to get a *topologically identical* copy of the original thing.



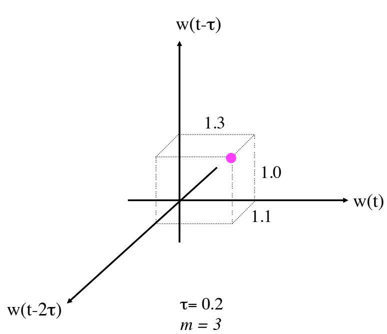
(almost)

Reconstruction space

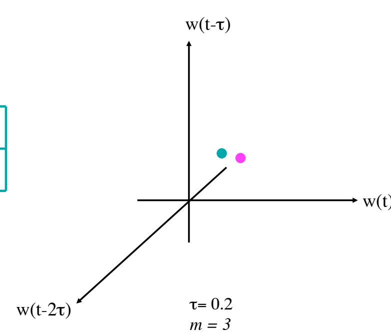


Mechanics

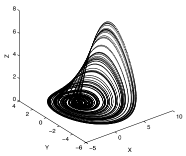
w	t
1.3	0.1
1.2	0.2
1.0	0.3
0.8	0.4
1.1	0.5
1.4	0.6
1.6	0.7



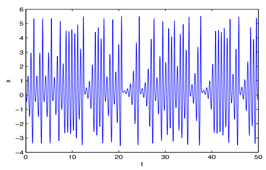
w	t
1.3	0.1
1.2	0.2
1.0	0.3
0.8	0.4
1.1	0.5
1.4	0.6
1.6	0.7



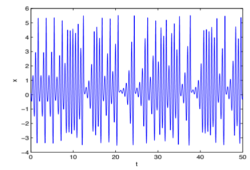
TISEAN's `delay` command does this

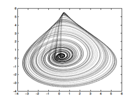


Measure just the *x* coordinate…



…and then embed:

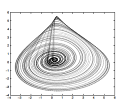




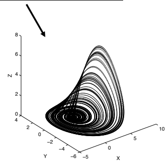
τ = 150

Takens* theorem

For the right τ and enough dimensions, the embedded dynamics are diffeomorphic to (and thus have same topology as) the original state-space dynamics.



τ = 150



* Whitney, Mane, …

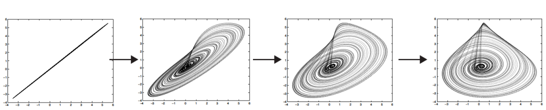
Note: the measured quantity must be a smooth, generic function of at least one state variable, and must be uniformly sampled in time.

Diffeomorphic: mapping from the one to the other is differentiable and has a differentiable inverse.

What that means:

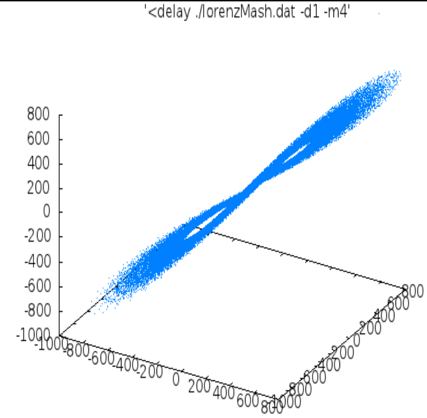
- *qualitatively* the same shape (topology)
- i.e., can deform one into the other
- have same dynamical invariants (e.g., λ)

Choosing τ:



τ = 1 τ = 50 τ = 100 τ = 150

NB: TISEAN contains tools that help you do this (e.g., `mutual`)



Choosing *m*

$m > 2d$: sufficient to ensure no crossings in reconstruction space (Takens et al.)…

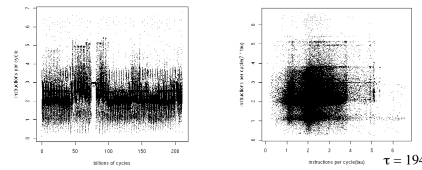
…but that may be overkill, and you rarely know d anyway.

"Embedology" paper: $m > 2\ d_{box}$
(box-counting dimension)

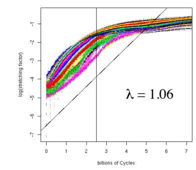
NB: TISEAN contains tools that help you do this (e.g., `false_nearest`)

NLTSA of computer performance dynamics

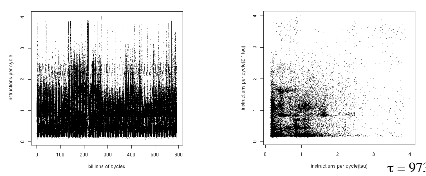
* nonlinear time-series analysis



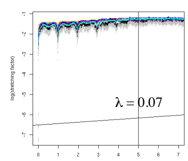
`bzip2` dynamics on an Intel Core2



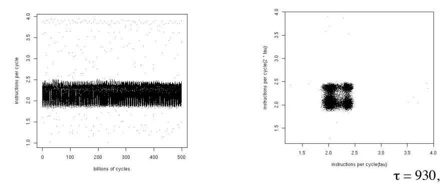
Mytkowicz *et al.*, *Chaos* **19**:033124



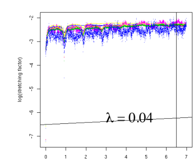
`bzip2` dynamics on an Intel Pentium 4



Mytkowicz *et al.*, *Chaos* **19**:033124



`povray` dynamics on an Intel Core2



Mytkowicz *et al.*, *Chaos* **19**:033124

Caveat: need enough data…

If Δt is not uniform

~~Theorem (Takens): for $\tau > 0$ and $m > 2d$, reconstructed trajectory is diffeomorphic to the true trajectory~~

~~Conditions: evenly sampled in time, smooth generic measurement function~~

Interspike interval embedding

<u>idea</u>: lots of systems generate spikes — hearts, nerves, etc.

if you assume that the spikes are the result of an integrate-and-fire system, then the Δt has a one-to-one correspondence to some state variable's *integrated* value…

<u>*in which case the embedding theorems still hold.*</u>

(with the Δ*t*s as state variables)

Sauer *Chaos* **5**:127