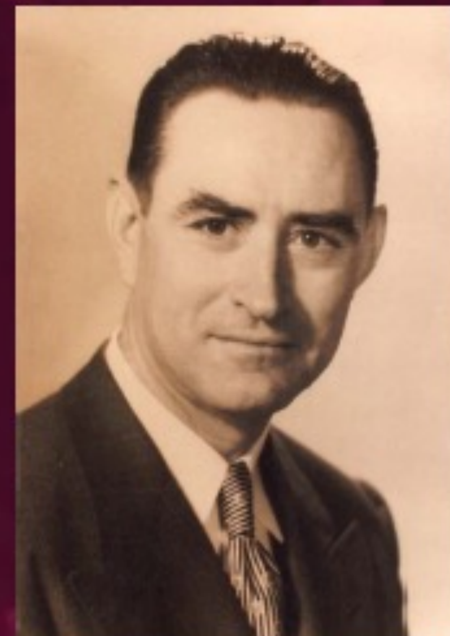
Villiam T. and Idalia Reid: His Mathematics anc
ler Mathematical Family



William T. Reid
1907 - 1977

Idalia Rei
1907 - 200

John A. Burns

VirginiaTec
Invent the Fu

ICAM

Interdisciplinary Center for Applied Mathematic

2010 SIAM NATIONAL MEETING

THAT VISION THING

IN 1973 BEFORE I LEFT OKLAHOMA TO GO TO BROWN, DR. REID CALLED ME INTO HIS OFFICE AND GAVE ME THE FOLLOWING (VERY SPECIFIC) CAREER ADVICE:

- THE WORLD IS CHANGING AND YOUR MATHEMATICS MUST CHANGE TOO
- YOU SHOULD LEARN TO COMPUTE – COMPUTERS WILL TRANSFORM THE WAY WE DO SCIENCE AND MATHEMATICS
- STATISTICS AND STOCHASTICS WILL BECOME MORE IMPORTANT BECAUSE THE REAL WORLD IS FULL OF UNCERTAINTIES
- PAY ATTENTION TO APPLICATIONS – NOT THE CLASSICAL APPLICATIONS, BUT LOOK AT NEW AREAS LIKE MATHEMATICAL BIOLOGY, COMMUNICATION SYSTEMS OR COMPUTER SCIENCE

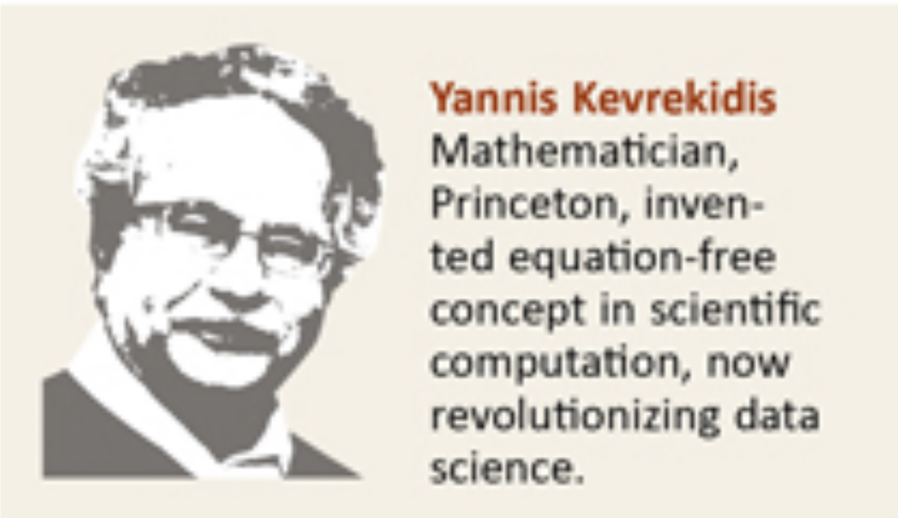
NO EQUATIONS, NO VARIABLES
NO PARAMETERS - even NO SPACE!

Data, and the computational modeling of Complex/Multiscale Systems

I.G. Kevrekidis, W. Gear, R. Coifman, G. Hummer
& good people: R.Talmon, E. Chiavazzo, R.Covino, A. Georgiou

Department of Chemical Engineering, PACM & Mathematics
Princeton University, Princeton, NJ 08544
This year: EinsteinVisitor, ZIB/FU & Fischer Fellow, IAS-TUMuenchen

Some well intended inaccuracies (and corrections)



Some well intended inaccuracies (and corrections)

Yannis Kevrekidis
Mathematician, Princeton, invented equation-free concept in scientific computation, now revolutionizing data science.

Applied (Mathematician)
also
Engineer
(Chemical, Computational)
Really
Modeler

Some well intended inaccuracies (and corrections)

Applied (Mathematician)
also
Engineer
(Chemical, Computational)
Really
Modeler

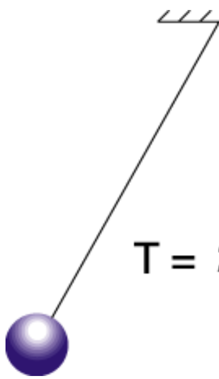
Working for a guy revolutionizing data science (R. R. Coifman, at Yale)

Some well intended inaccuracies (and corrections)

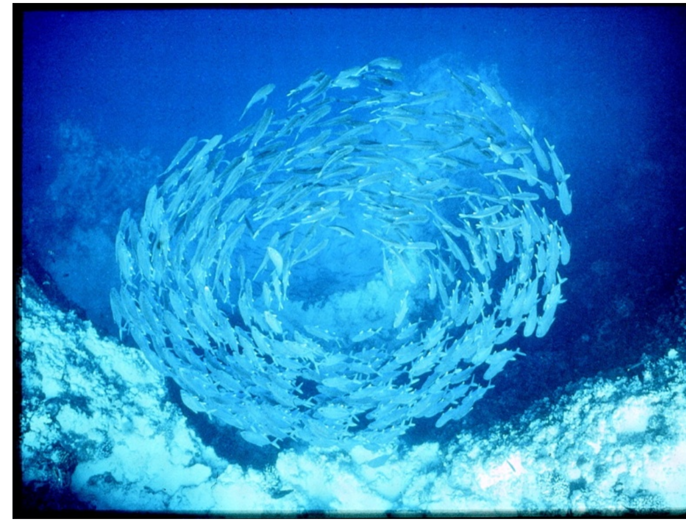
Applied (Mathematician)
also
Engineer
(Chemical, Computational)
Really
Modeler

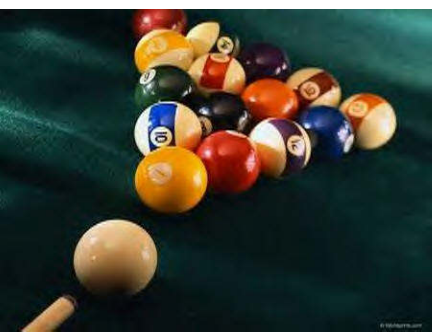
Working with a guy revolutionizing data science (R. R. Coifman, at Yale)

J. Ottino, NWU



$$T = 2\pi\sqrt{\frac{\ell}{g}}$$





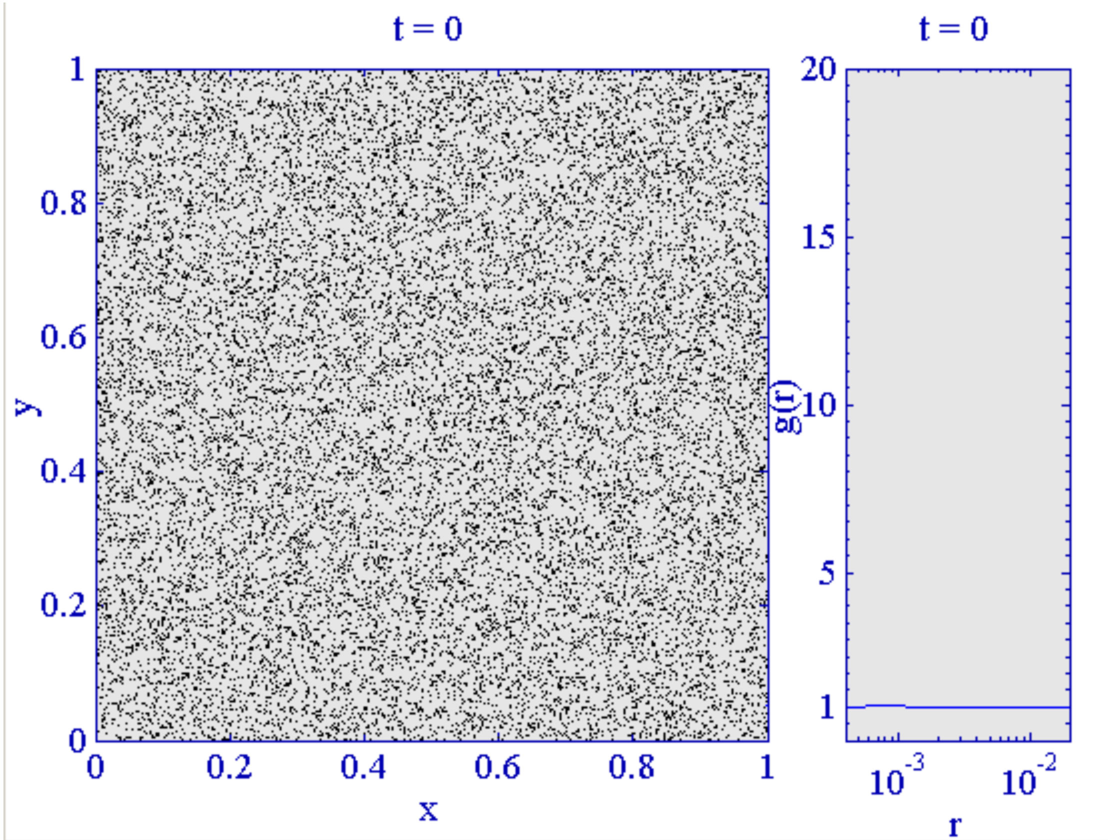
"simple" "complicated" "complex"

Clustering and stirring in a plankton model

An “equation-free” demo

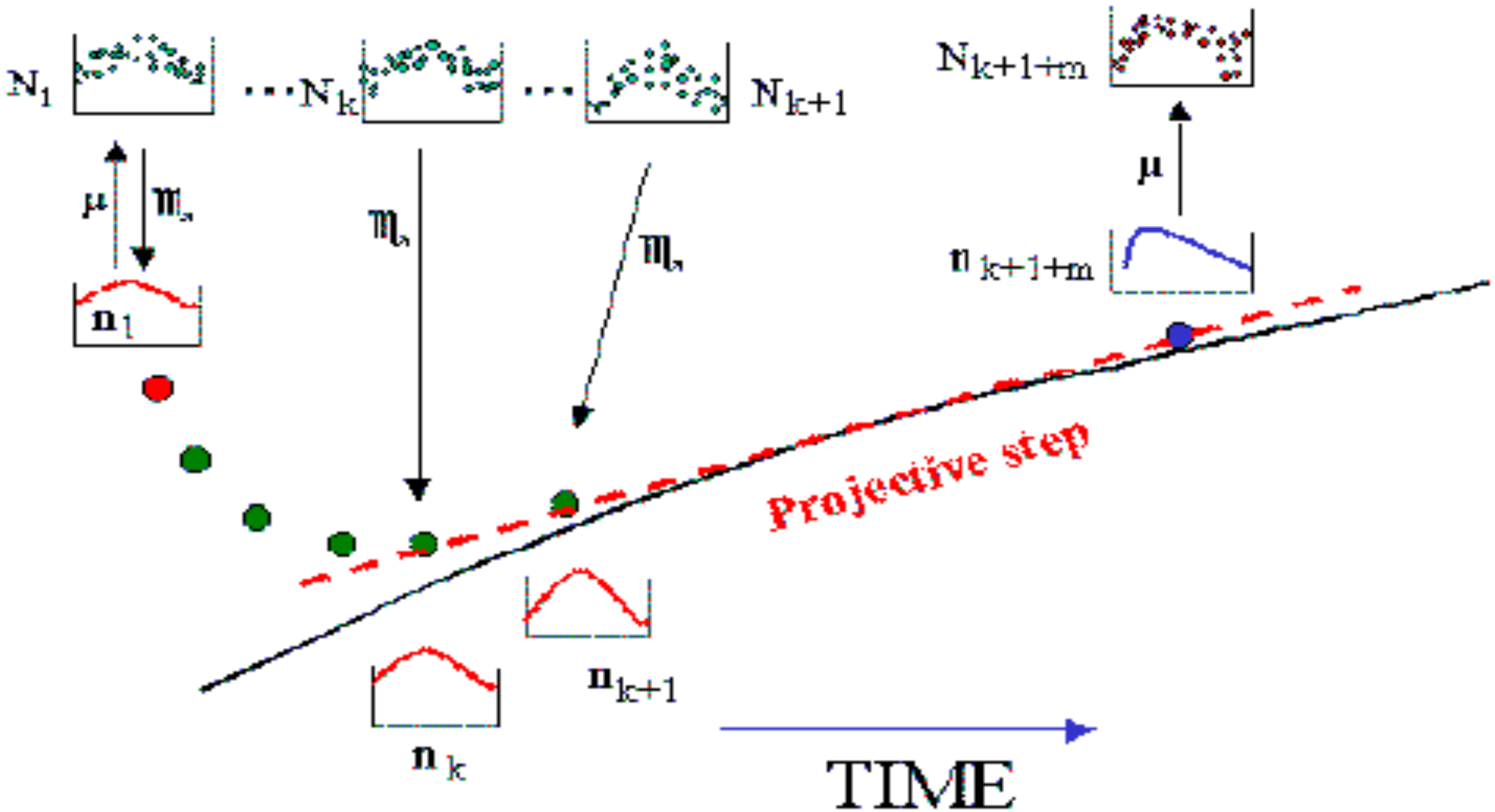
Young, Roberts and Stuhne, *Nature* 2001

Dynamics of System with convection

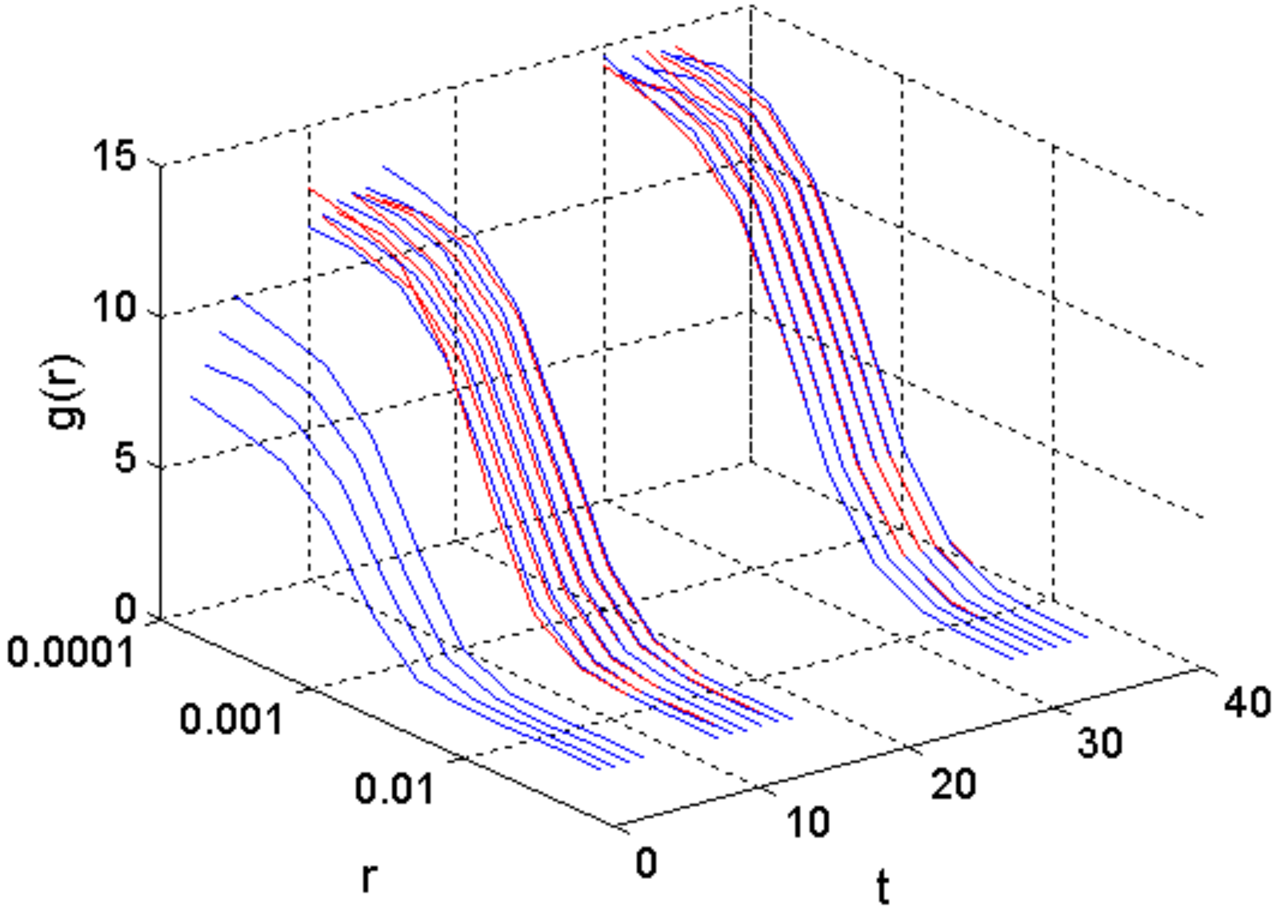


Coarse Projective Integration

(coarse projective Forward Euler)

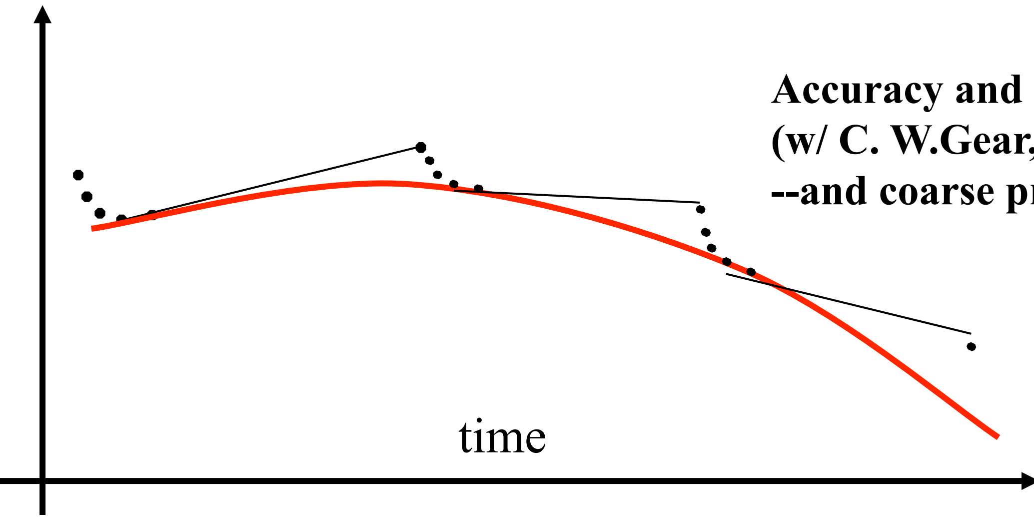


Projective Integration: From t=2,3,4,5 to 10



Projective Integration - a sequence of outer integration steps

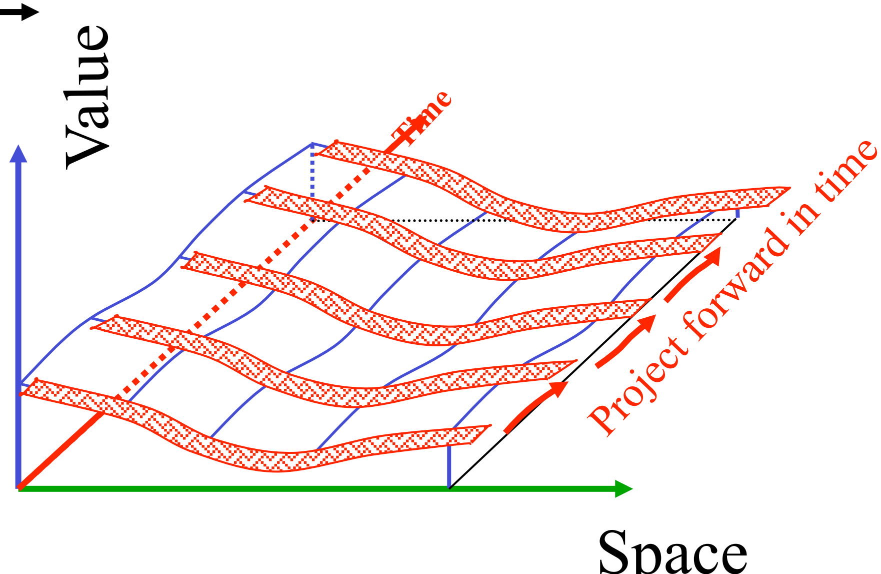
based on inner simulator + estimation (stochastic inference)



Accuracy and stability of these methods – NEC/TR 2001
(w/ C. W.Gear, *SIAM J.Sci.Comp.* 03, *J.Comp.Phys.* 03,
--and coarse projective integration (inner LB)
Comp.Chem.Eng. 2002

Projective methods in time:

-perform detailed simulation for short periods
or use existing/legacy codes
- and then *extrapolate* forward over large steps



Effective simplicity

- Construct *predictive models* (deterministic, Markovian)
- Get information from them: *CALCULUS, Taylor series*
  - Derivatives in time to jump in time
  - Derivatives in parameter space for sensitivity /optimization
  - Derivatives in phase space for contraction mappings
  - Derivatives in physical space for PDE discretizations

In complex systems --- no derivatives at the level we need them

sometimes no variables ---- no calculus

If we know what the right variables are, we can

PERFORM differential operations

on the right variables – ***A Calculus for Complex Systems***

So, the main point:

- You have a "detailed simulator" --- ***or an experiment***
- **Somebody tells you what are good coarse variable(s)**
- Then you can use the IDEA
  - that a coarse equation exists
  - to accelerate the simulation/ extraction of information.
  - Equation-Free

  - BUT
  - How do we know what the right coarse variables are

Data mining techniques – Diffusion Maps

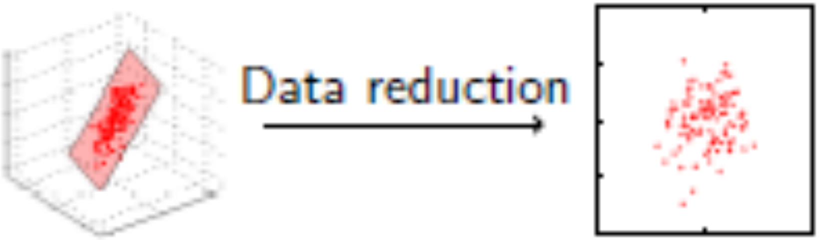
Where do good coarse variables come from?

- Systematic hierarchy (and mathematics)
  - Fourier modes, moments…..
- Experience/expertise/knowledge/brilliance
  - Human learning (“brain” data mining, observation, phase fields…)
- Machine Learning
  - Data mining, manifold learning (here: diffusion maps)

Our Approach: Data Reduction Techniques

Common *linear* technique: Principal Component Analysis (PCA)

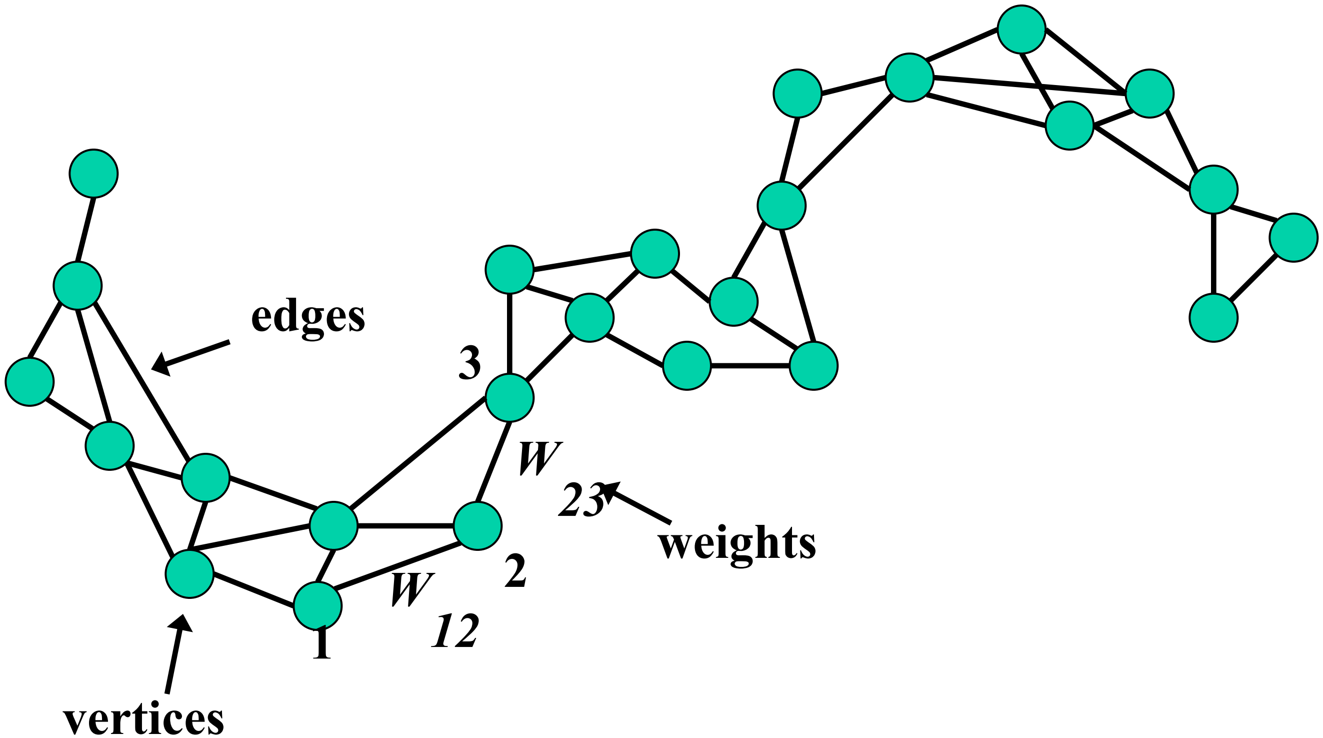
Project onto hyperplane which captures *maximum variance*



High dimensional data on low-dimensional structure

Reduced dimensionality

Dataset as Weighted Graph

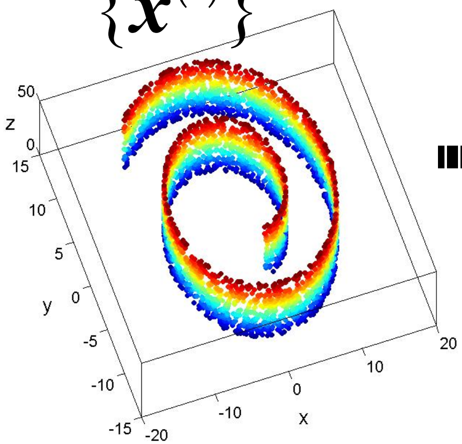


$$W_{ij} = \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{t}\right)$$

parameter $t \in R$

datapoints

$\{x^{(i)}\}$



Compute $N\times N$ "neighborhood" matrix K

$$K_{i,j} = \exp\left[-\left[\frac{\left\|x^{(i)} - x^{(j)}\right\|}{\sigma}\right]^2\right]$$

Parameter σ
Local neighborhood size

Compute diagonal normalization matrix D

$$D_{i,i} = \sum_{j=1}^{N} K_{i,j}$$

Compute Markov matrix M

$$M = D^{-1}K$$

Require: Eigenvalues λ and Eigenvectors Φ of M

Top → $M\Phi_1 = \lambda_1\Phi_1$

2nd → $M\Phi_2 = \lambda_2\Phi_2$

⋮

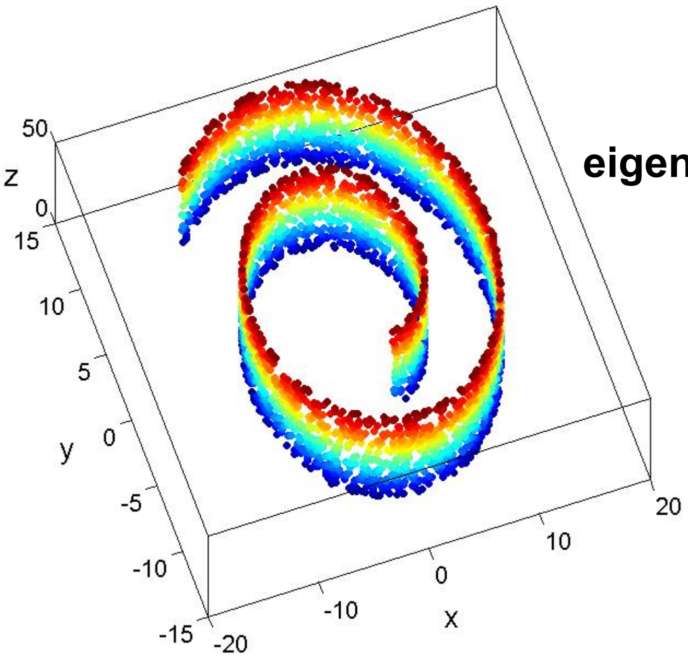
***N*th** → $M\Phi_N = \lambda_N\Phi_N$

$$\lambda_1 = 1 > \lambda_2 > \lambda_3 \cdots > \lambda_N$$

A few Eigenvalues\Eigenvectors provide meaningful information on dataset geometry

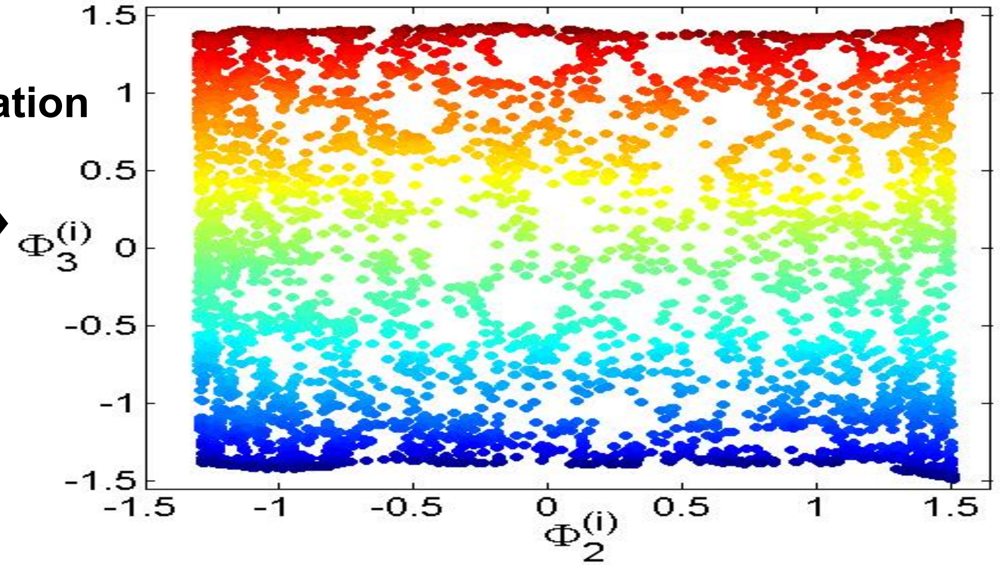
Diffusion Maps

Dataset in x, y, z



eigencomputation

Dataset Diffusion Map



N datapoints

$$\boldsymbol{x}^{(i)} = \left(x_i, y_i, z_i\right),\ i = 1, N$$

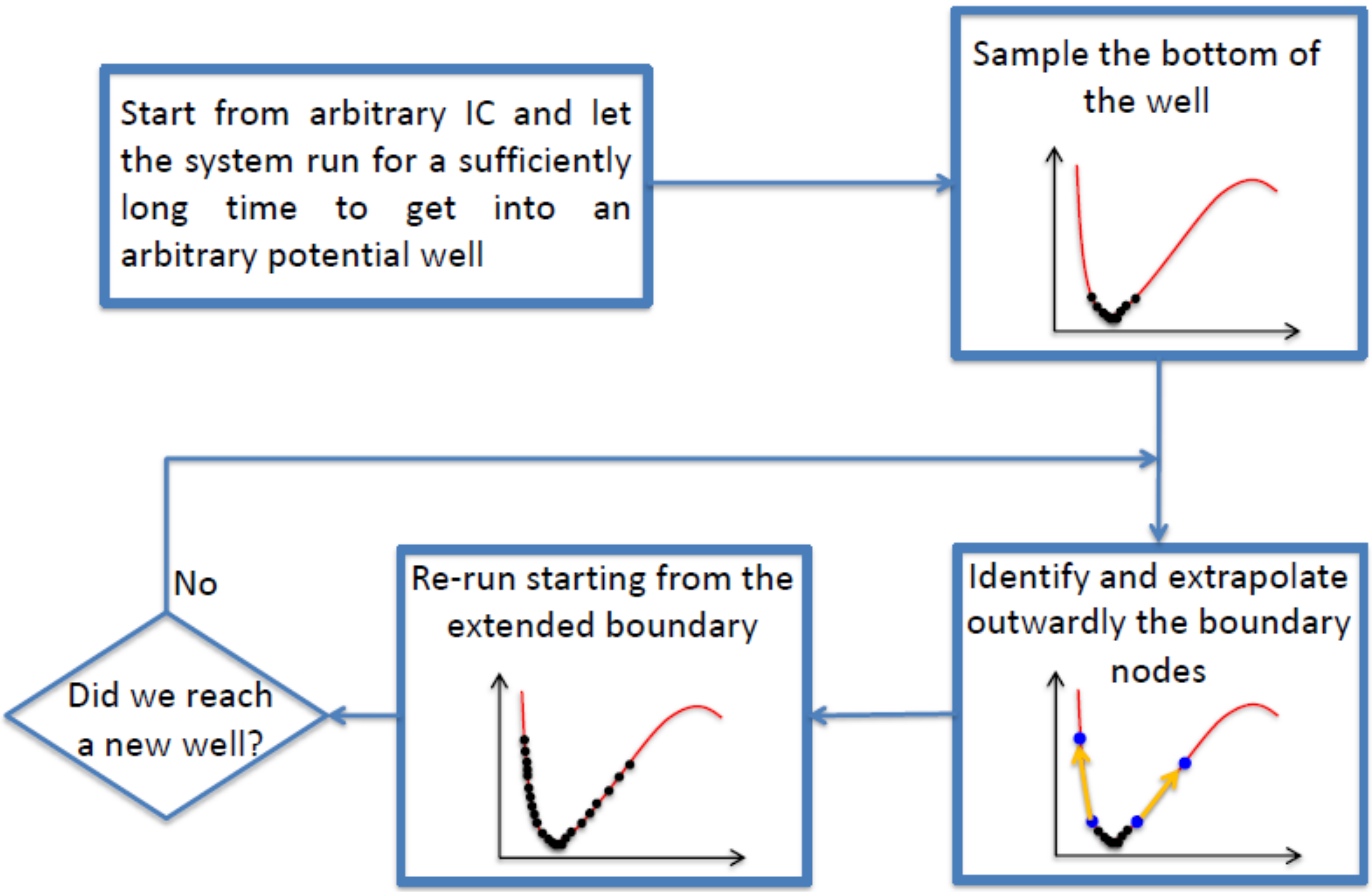
N datapoints

$$\boldsymbol{\Phi}^{(i)} = \left(\Phi_2^{(i)}, \Phi_3^{(i)}\right),\ i = 1, N$$

R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS* 102 (2005).

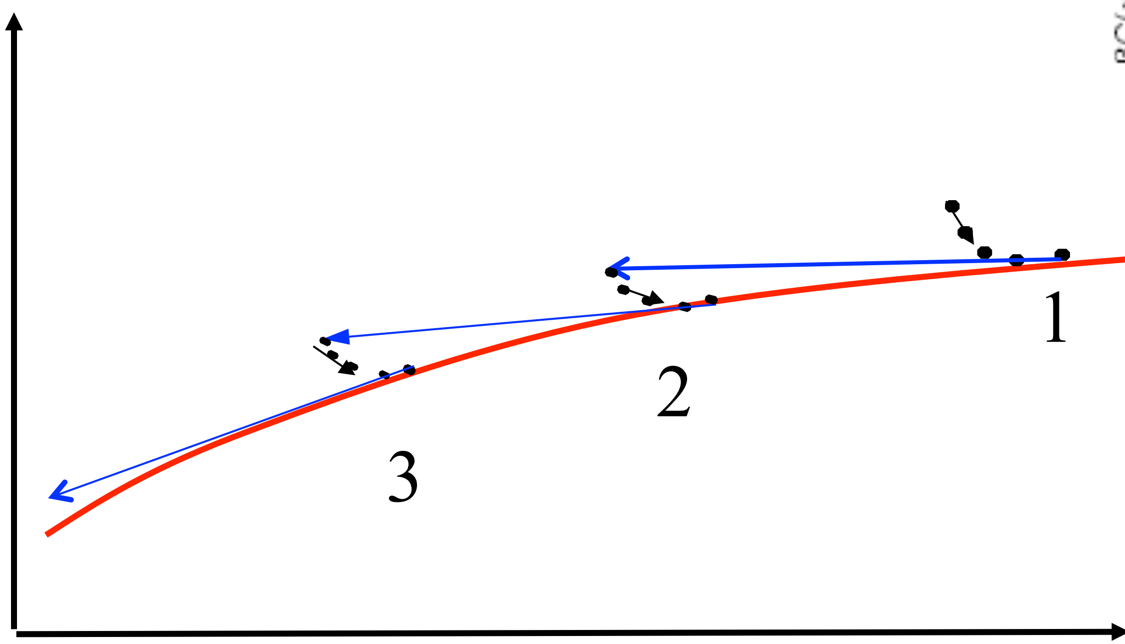
B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis, Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.* 21 (2006).

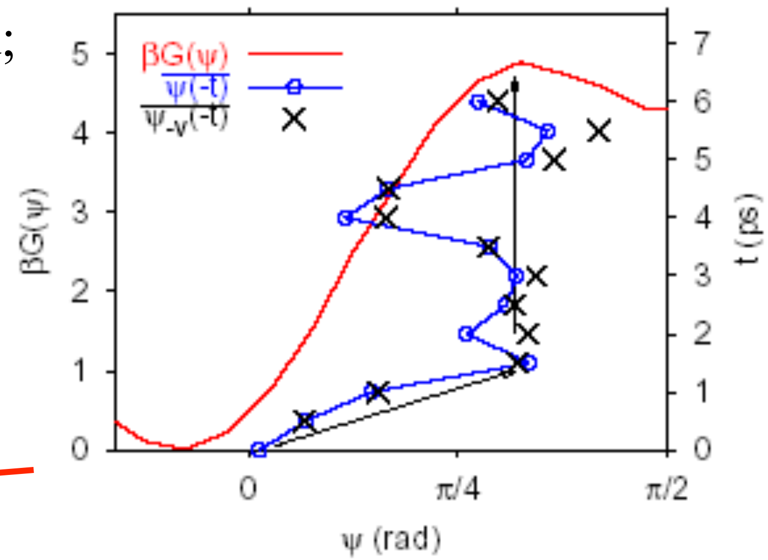
Systematic exploration: Basic idea

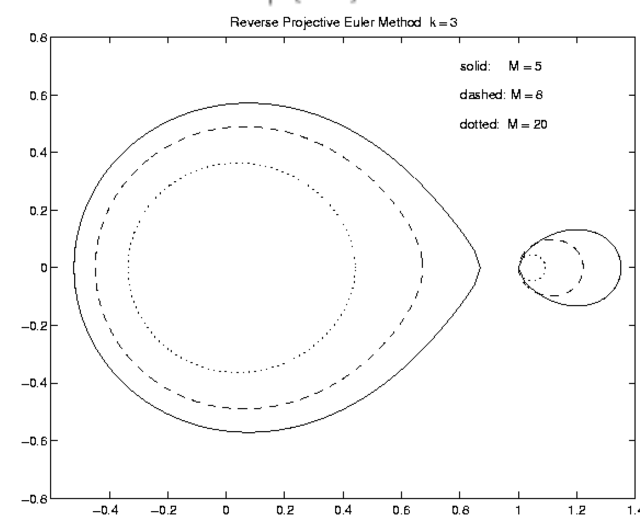


An oooooold slide – 2002– w/ G. Hummer (then NIH, now MPI Biophysik)

Reverse Projective Integration –
a sequence of outer integration steps backward;
based on forward steps + estimation

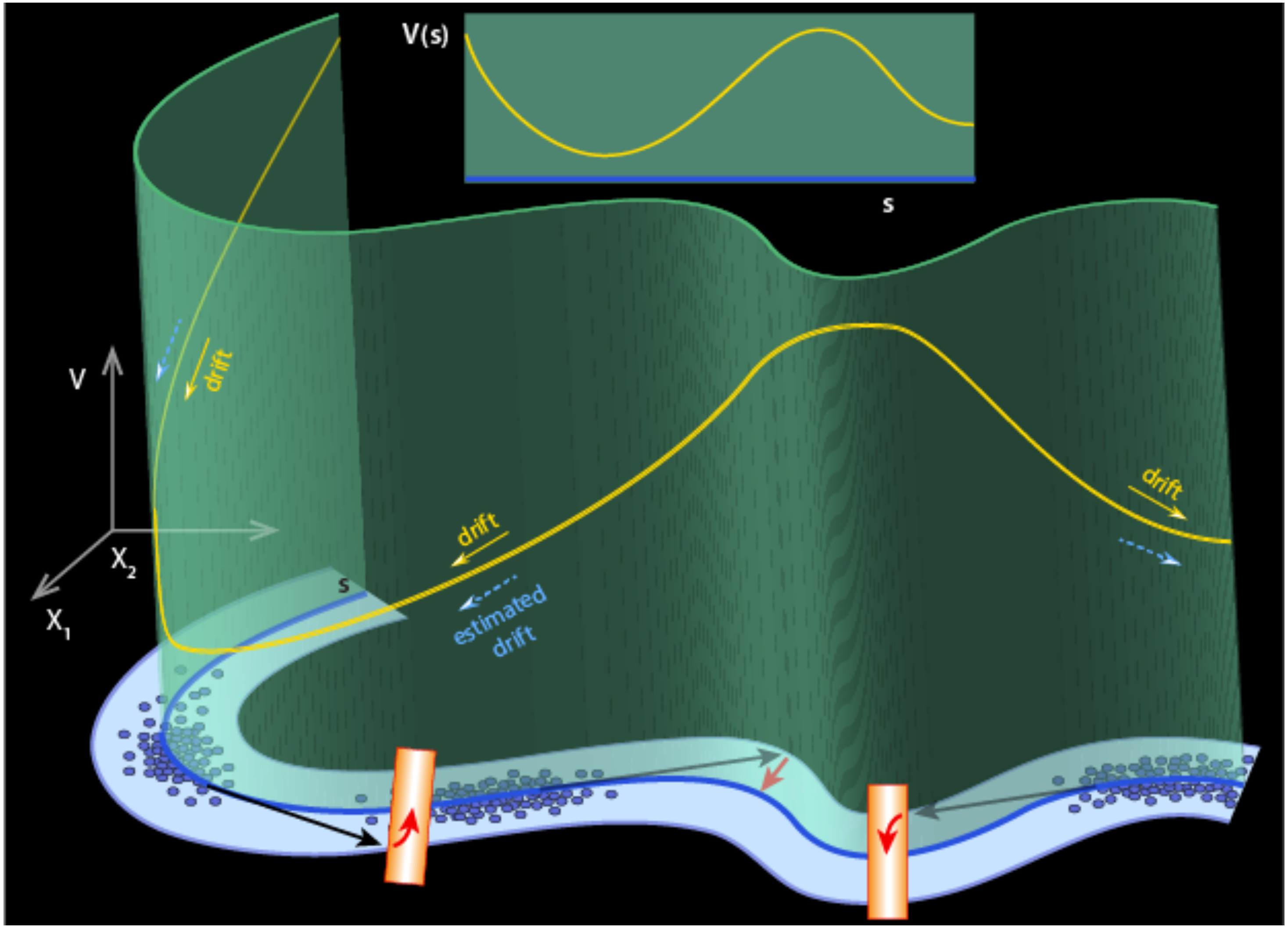




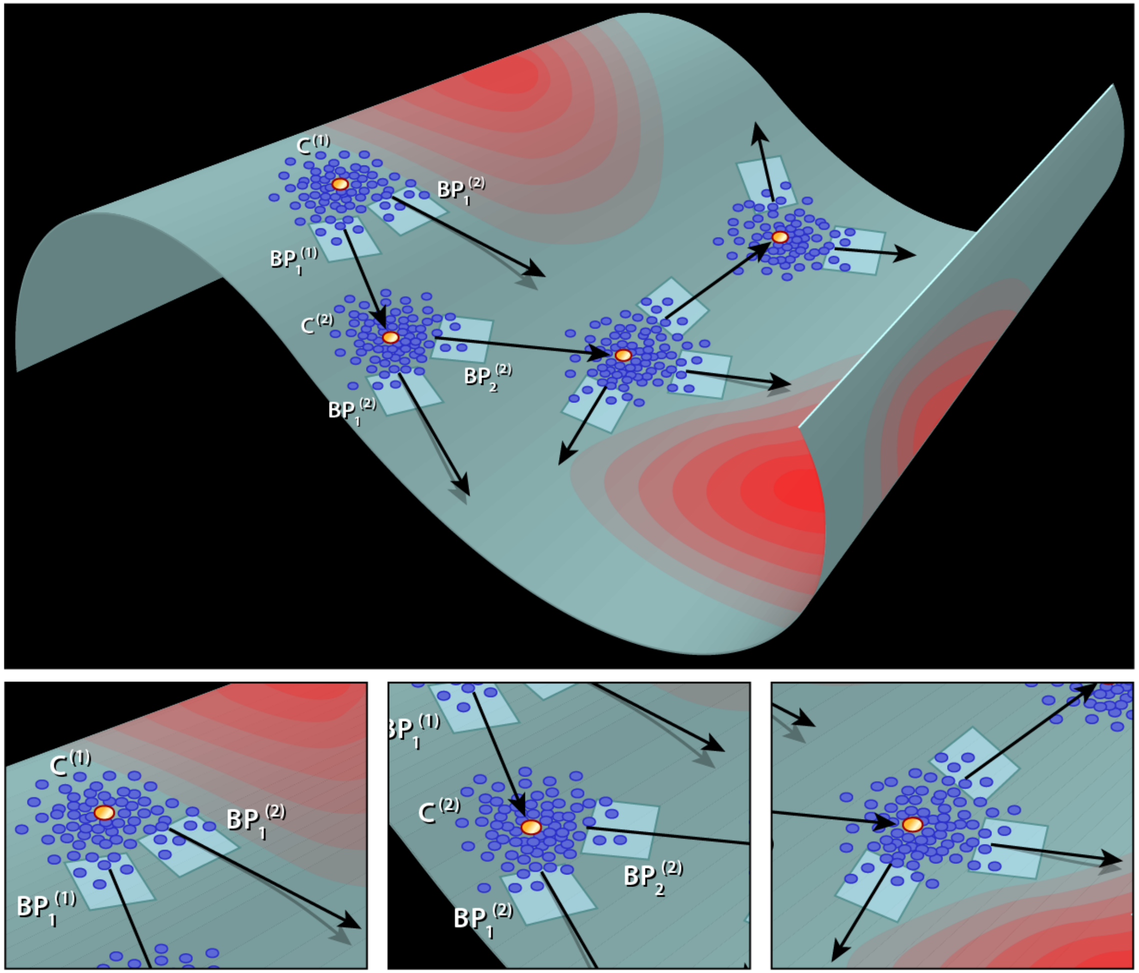


Reverse Integration: **a little forward,**
and then **a lot backward !**

We are studying the accuracy and stability of these methods

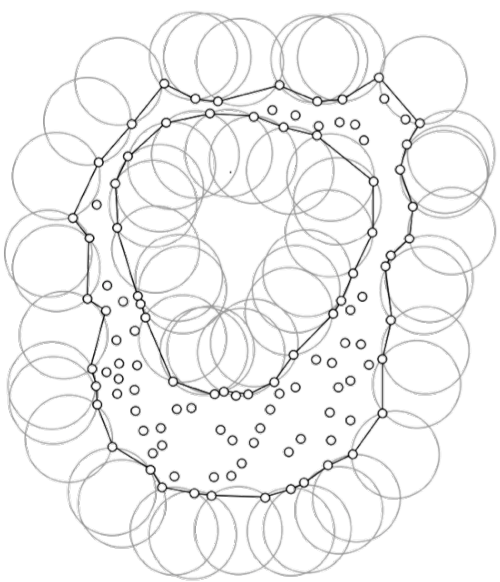


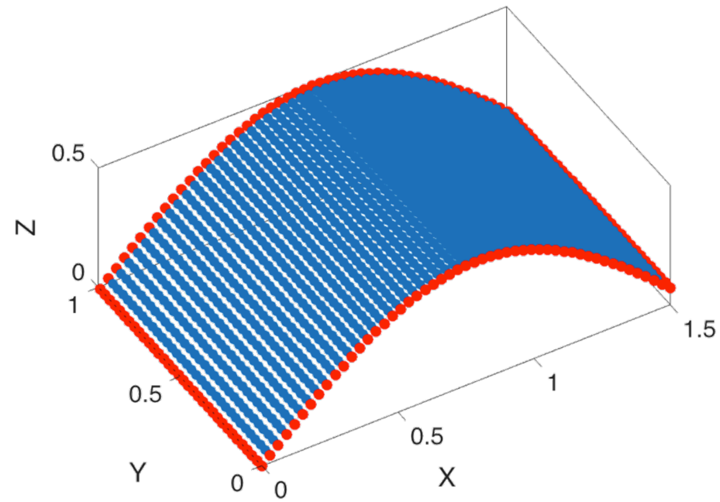
V(s)
s
V
X2
X1
s
drift
drift
drift
estimated drift


C (1)
BP 1 (2)
BP 1 (1)
C (2)
BP 2 (2)
BP 1 (2)

Boundary Detection: Alpha-Shapes

- Fit circles of radius α^{-1} such that (a) no interior points & (b) 2 points on perimeter
- Points on the perimeter of the circle make up the boundary
- Performed in reduced space





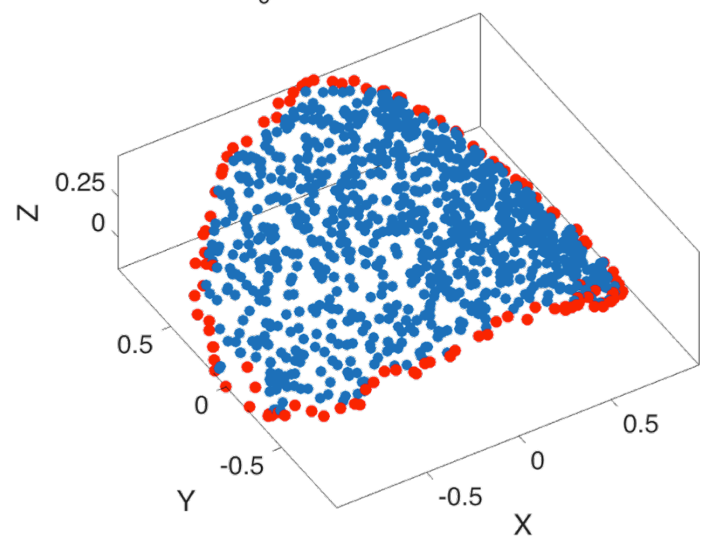
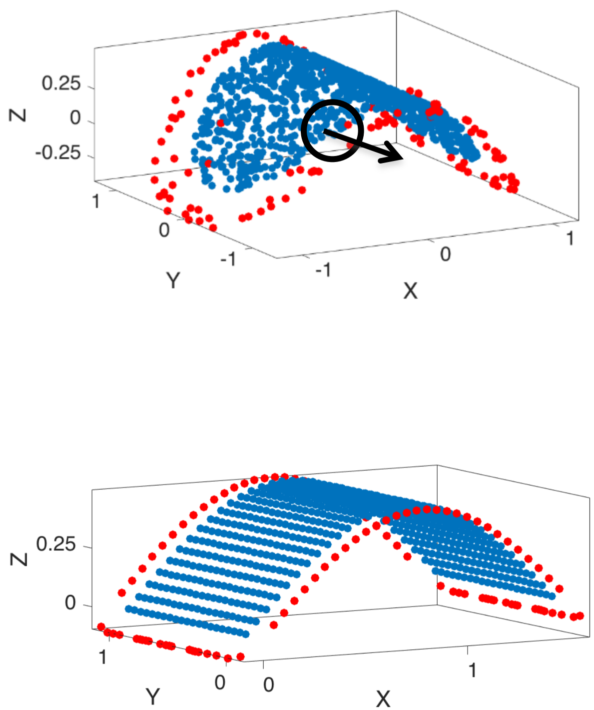


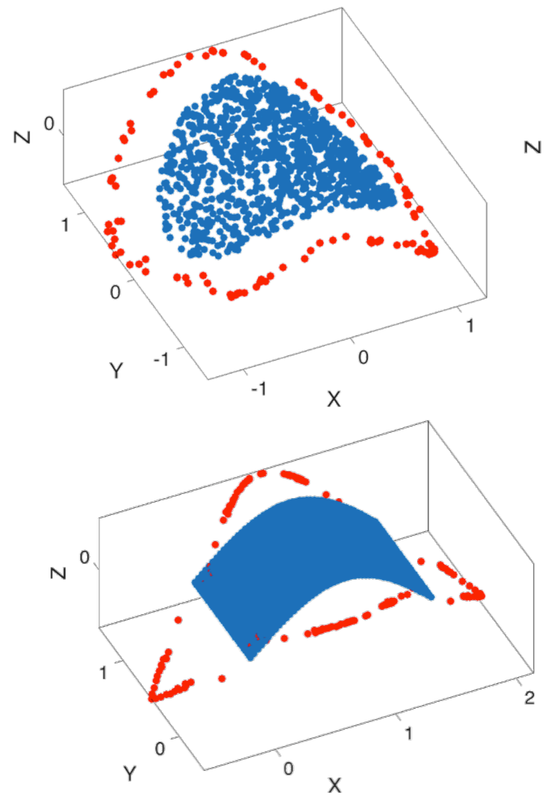
Image from H. Edelsbrunner and E. P. Mucke, "Three-dimensional alpha shapes," ACM Transactions on Graphics (TOG), vol. 13, no. 1, pp. 43–72, 1994.

Different Ways of Extending

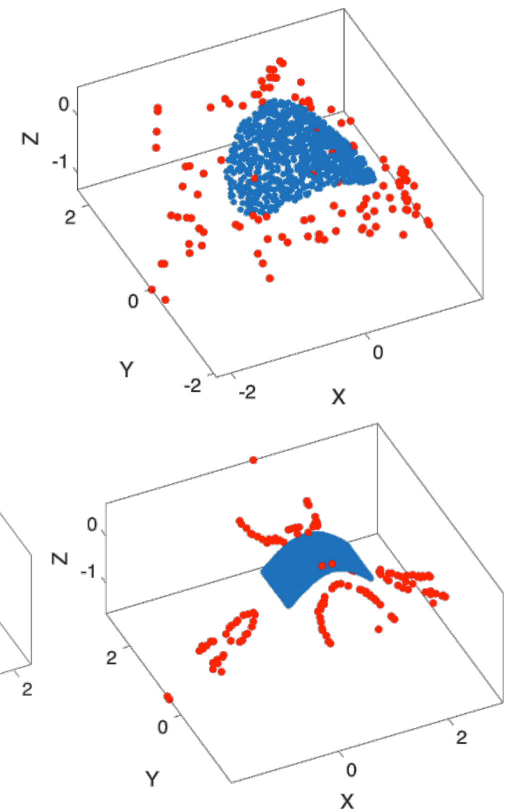
Local PCA



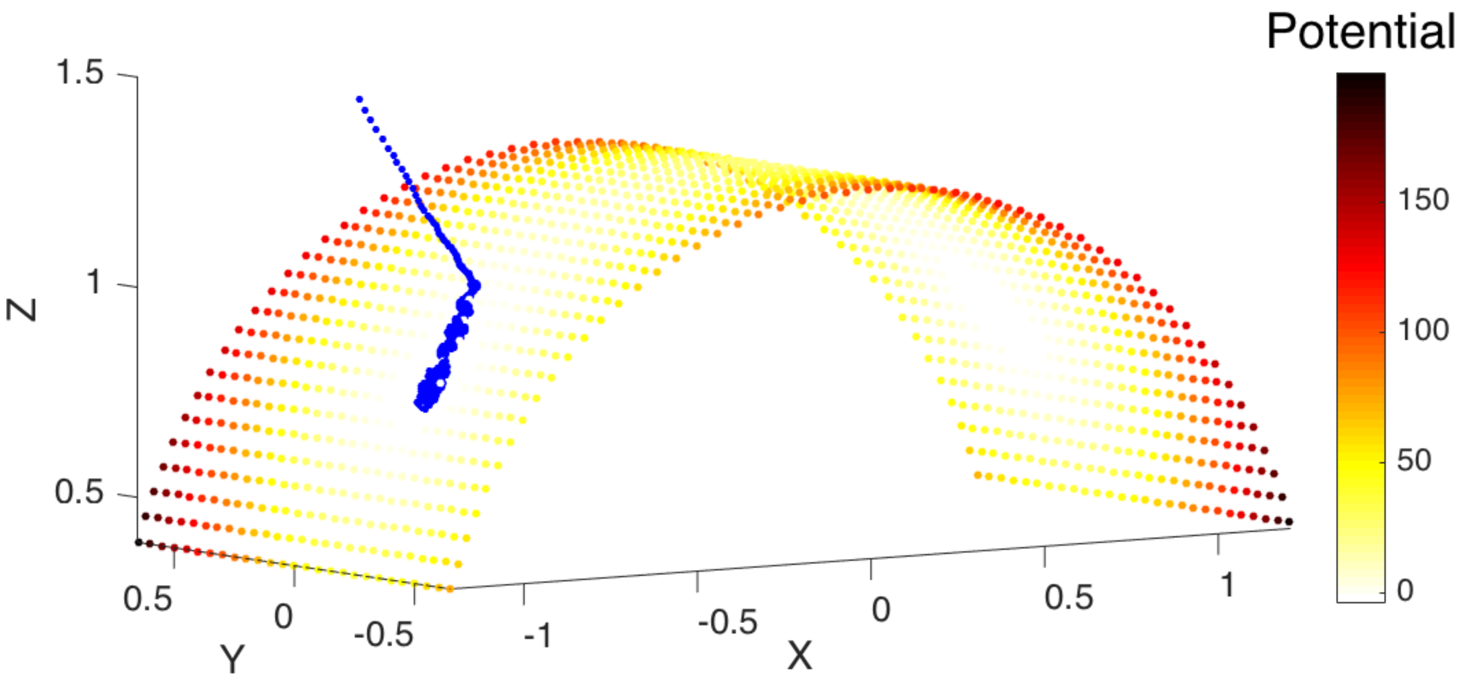
Cosine-DMAPs & Geometric Harmonics

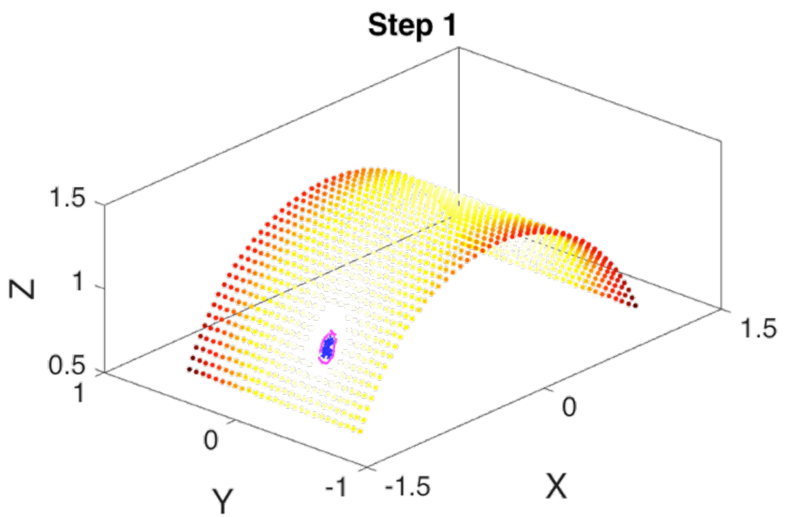


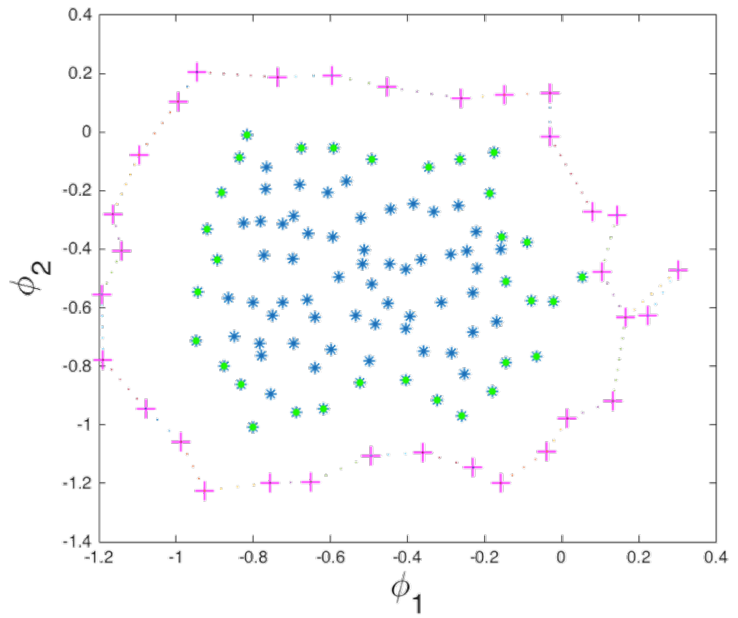
Sine-DMAPs & Geometric Harmonics



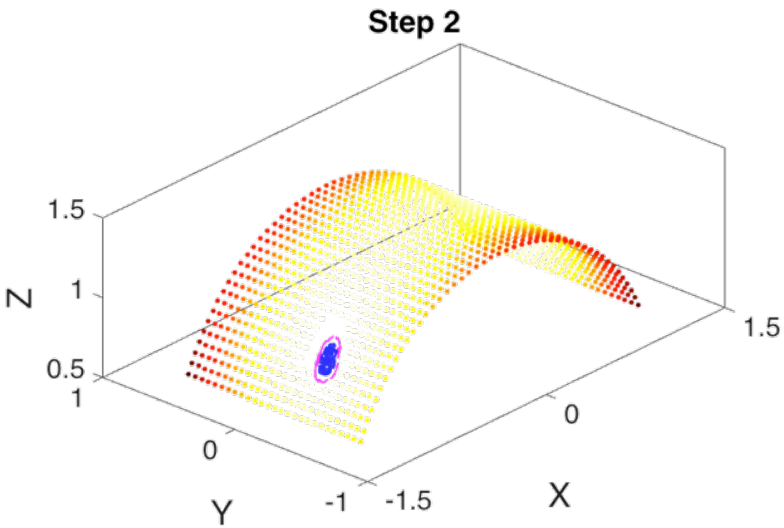
A Toy Example

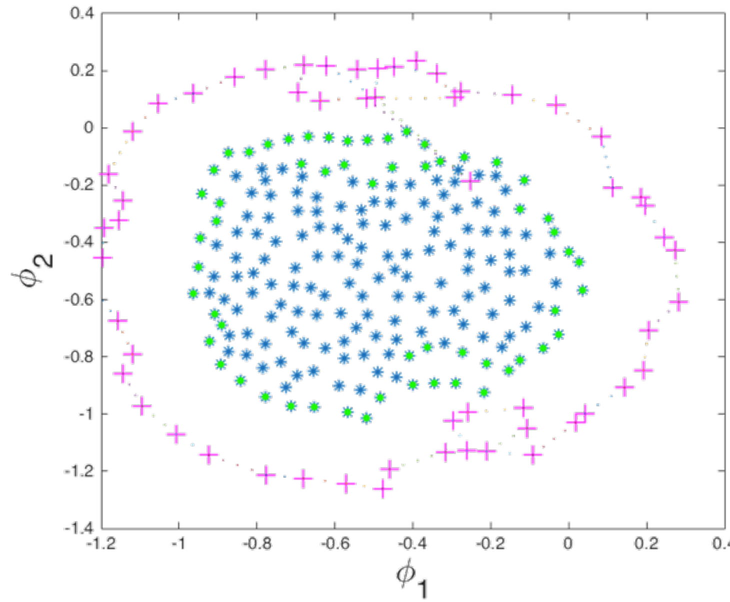




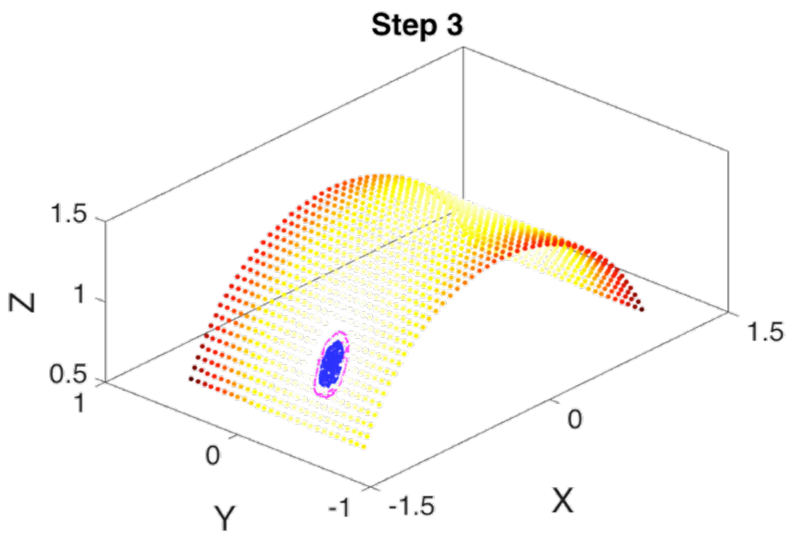


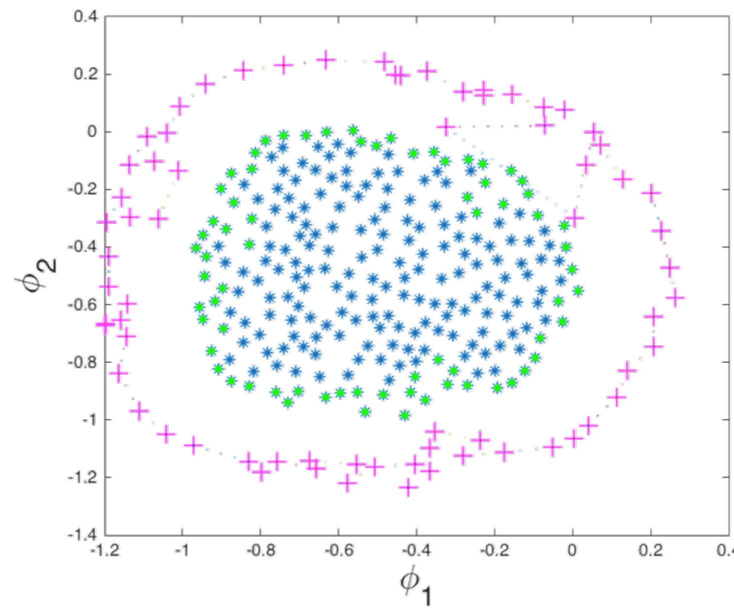
1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat



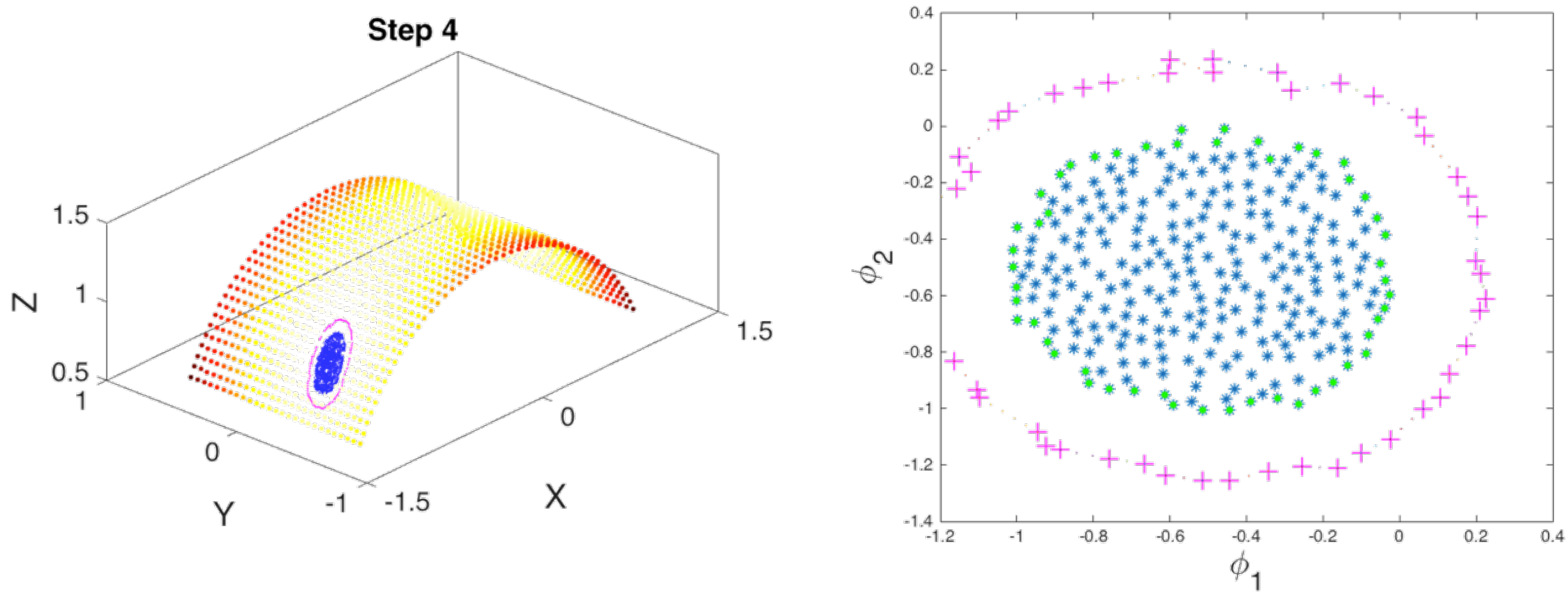


1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat

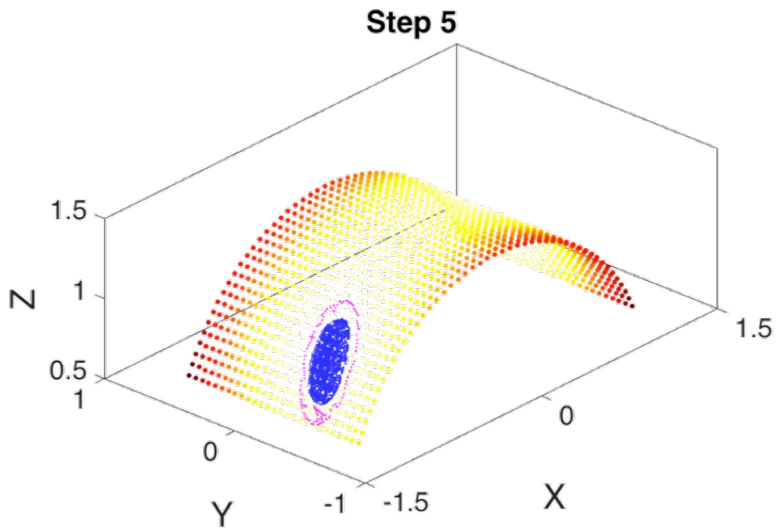


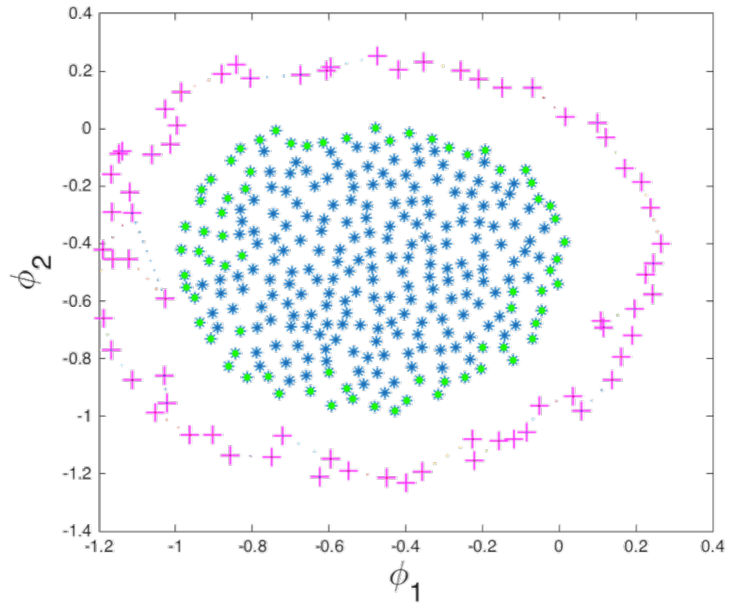


1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat

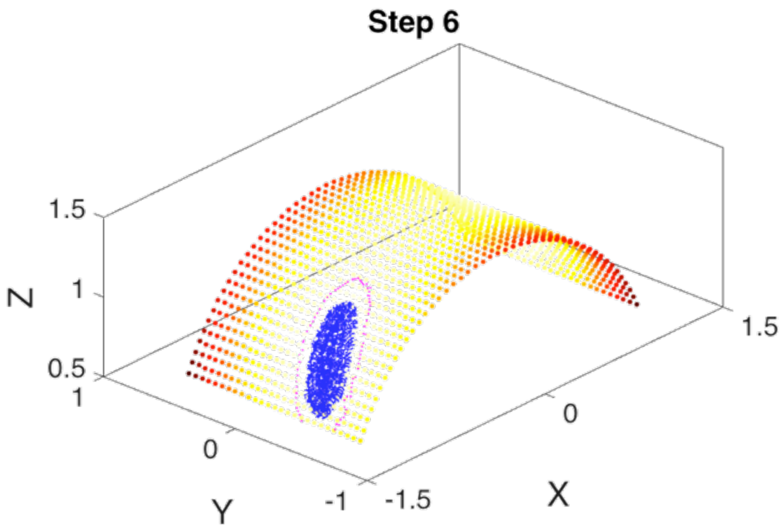


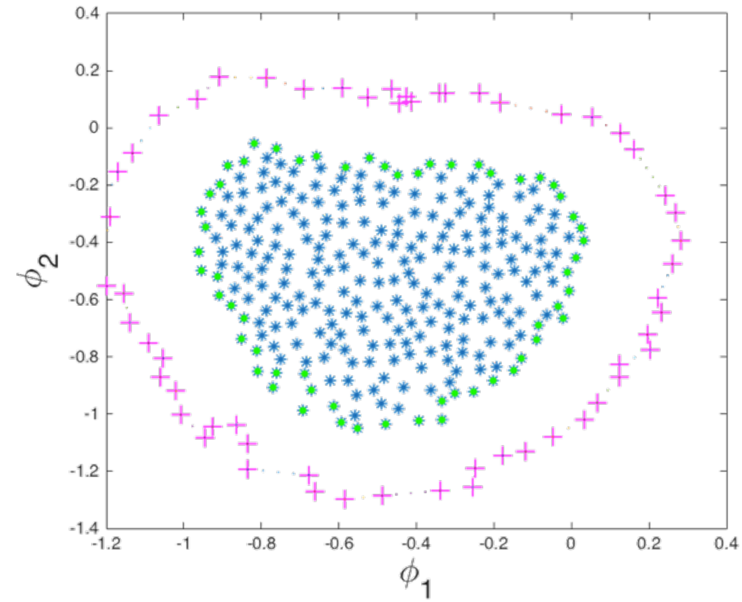
1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat



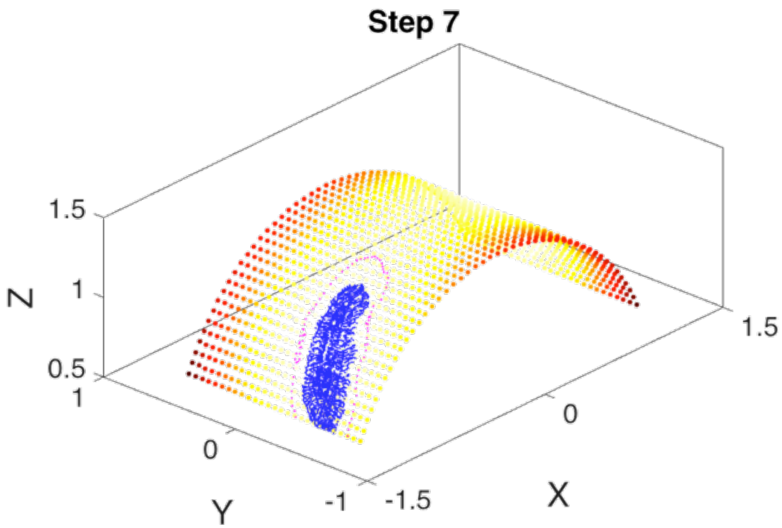


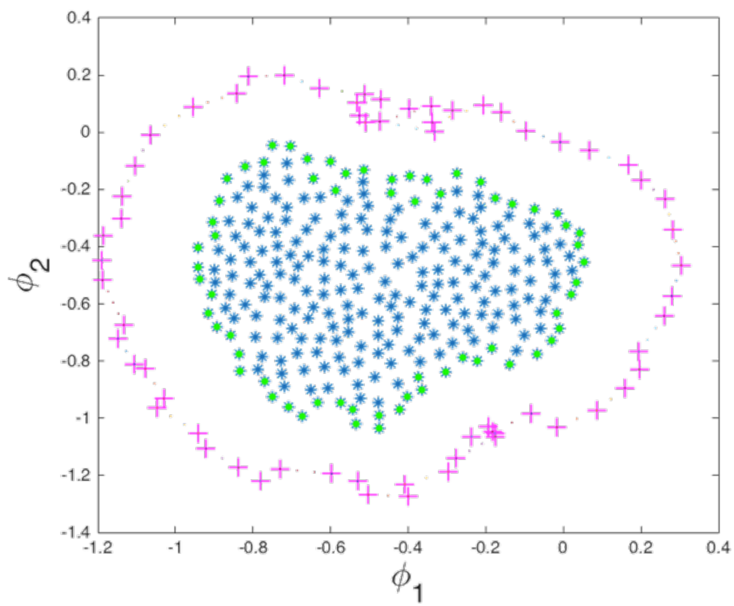
1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat





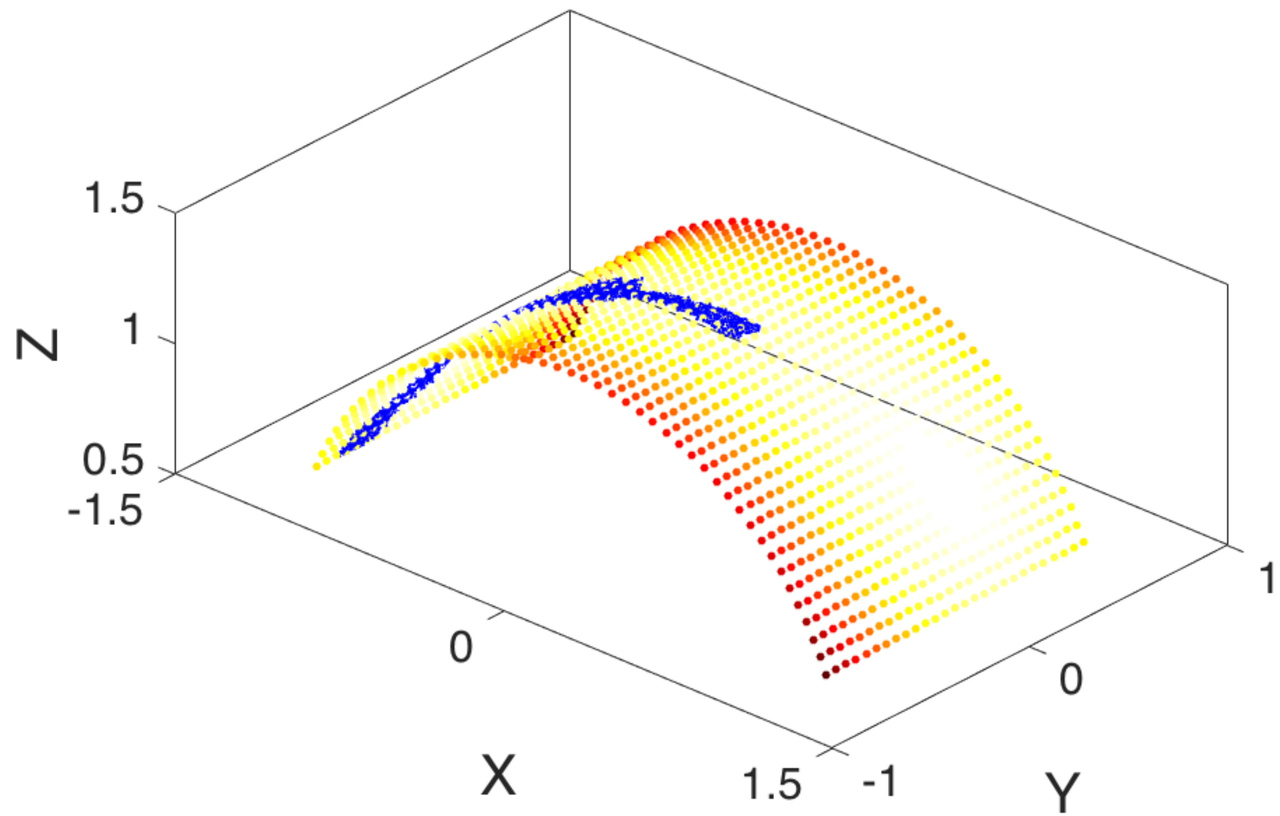
1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat



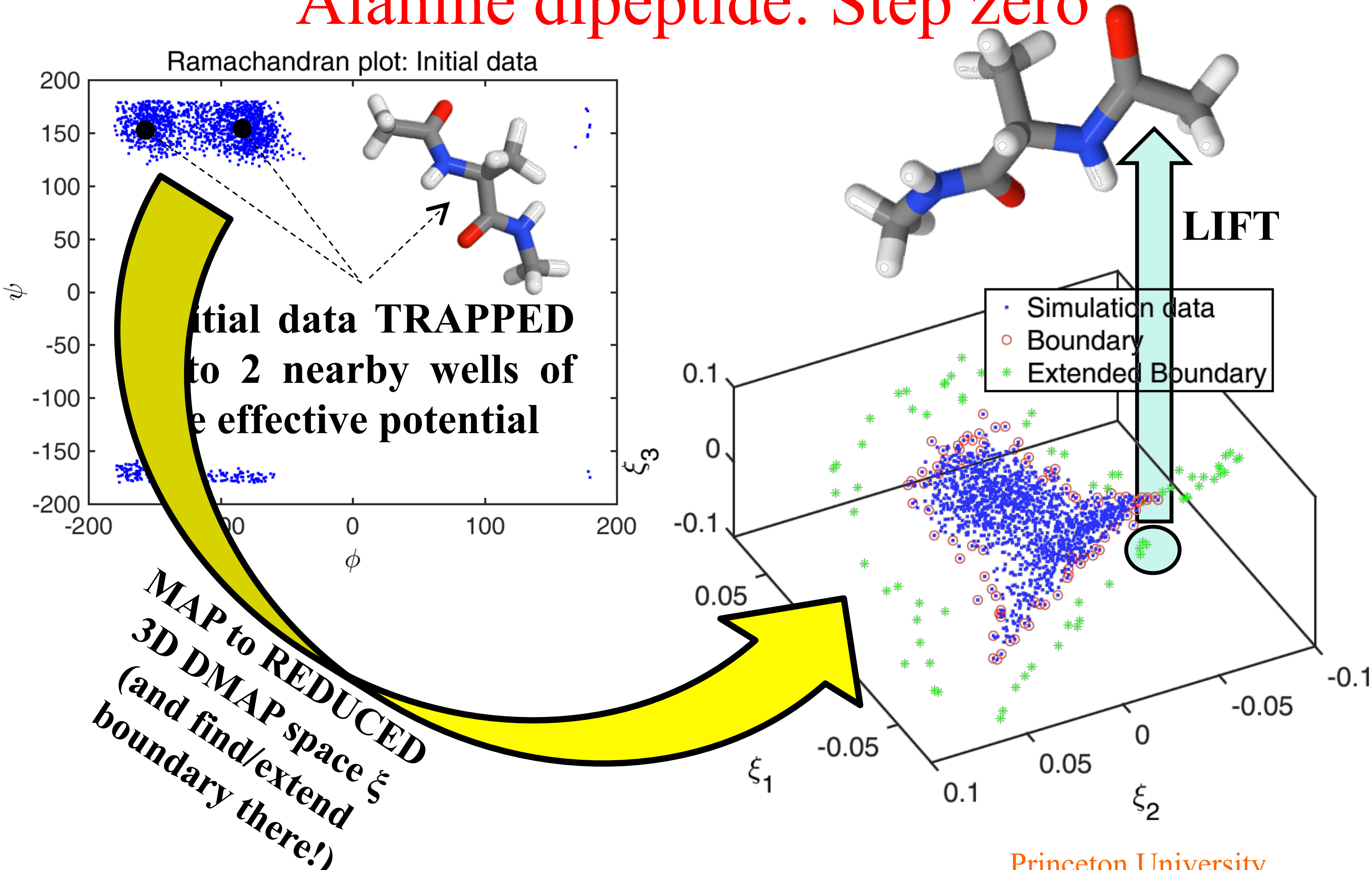


1. Obtain manifold sample
2. Reduce dimensions
3. Find boundary
4. Extend boundary
5. Lift into ambient space
6. Reinitialize & repeat

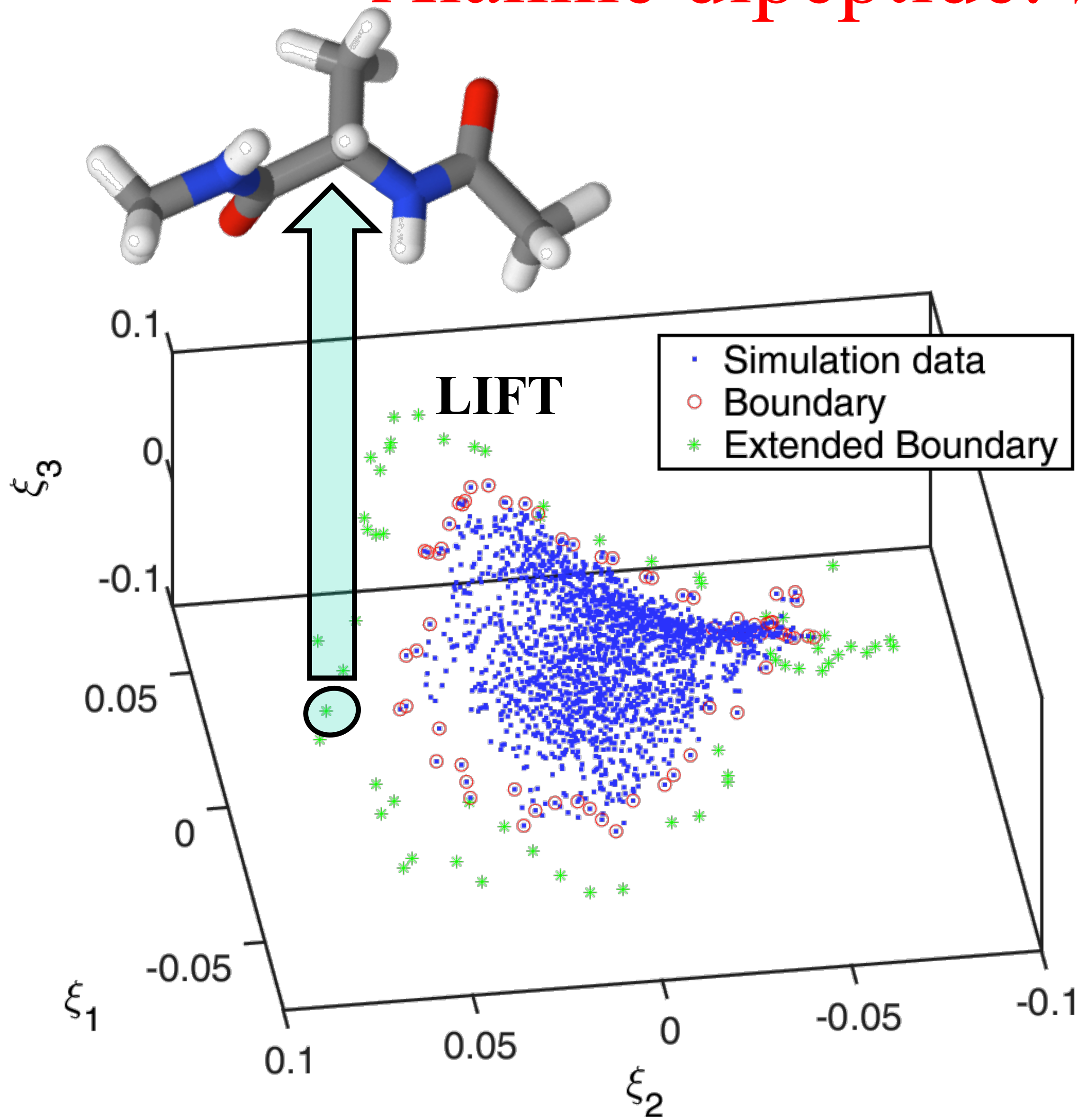
Finds 2nd Potential Well After 14 Iterations



Alanine dipeptide: Step zero

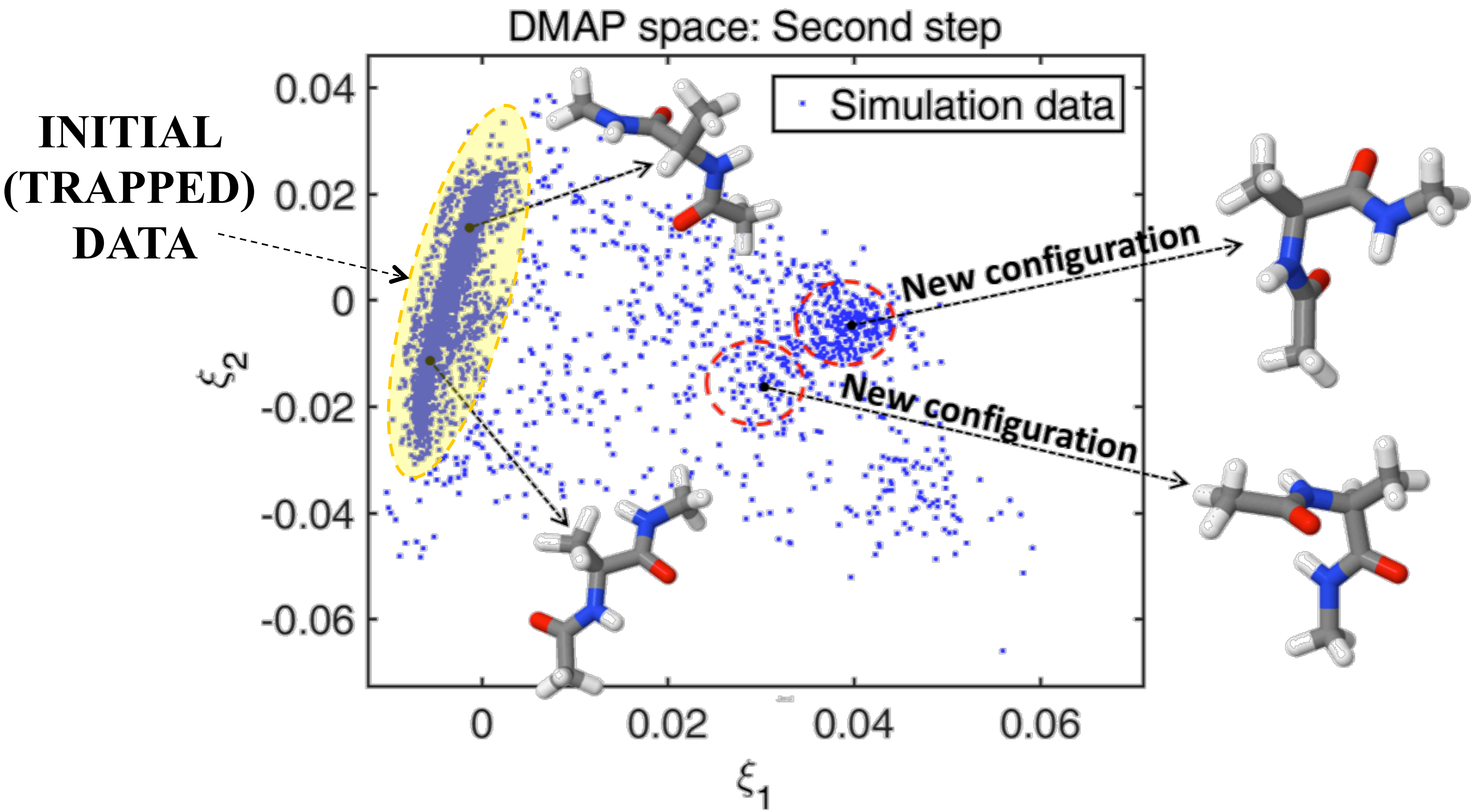


Alanine dipeptide: Step one

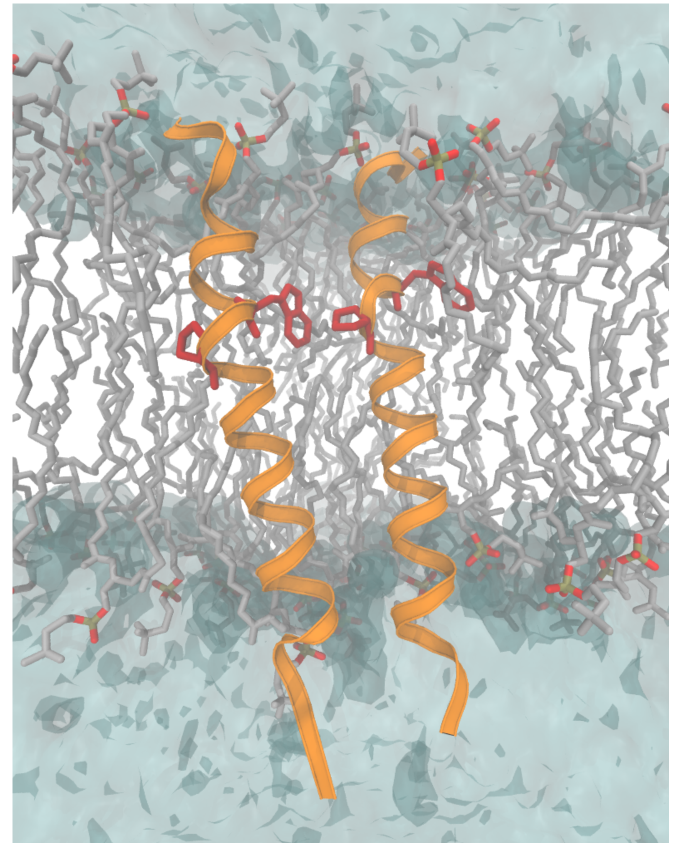


- **Re-run short simulations from previously lifted configurations;**
- **Down AGAIN to the 3D reduced DMAP space;**
- **Extend boundary in DMAP space;**
- **Lift-up into configuration space, towards UNKNOWN regions, and re-run short simulations AGAIN…**

Alanine dipeptide: After two steps only



A lipid saturation sensor in Euk. cells

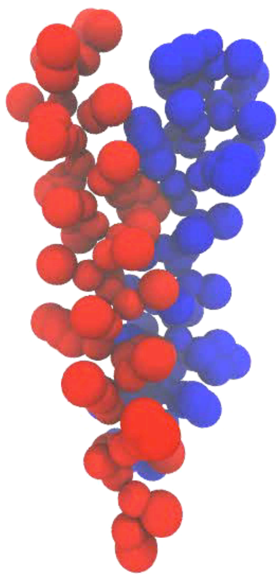


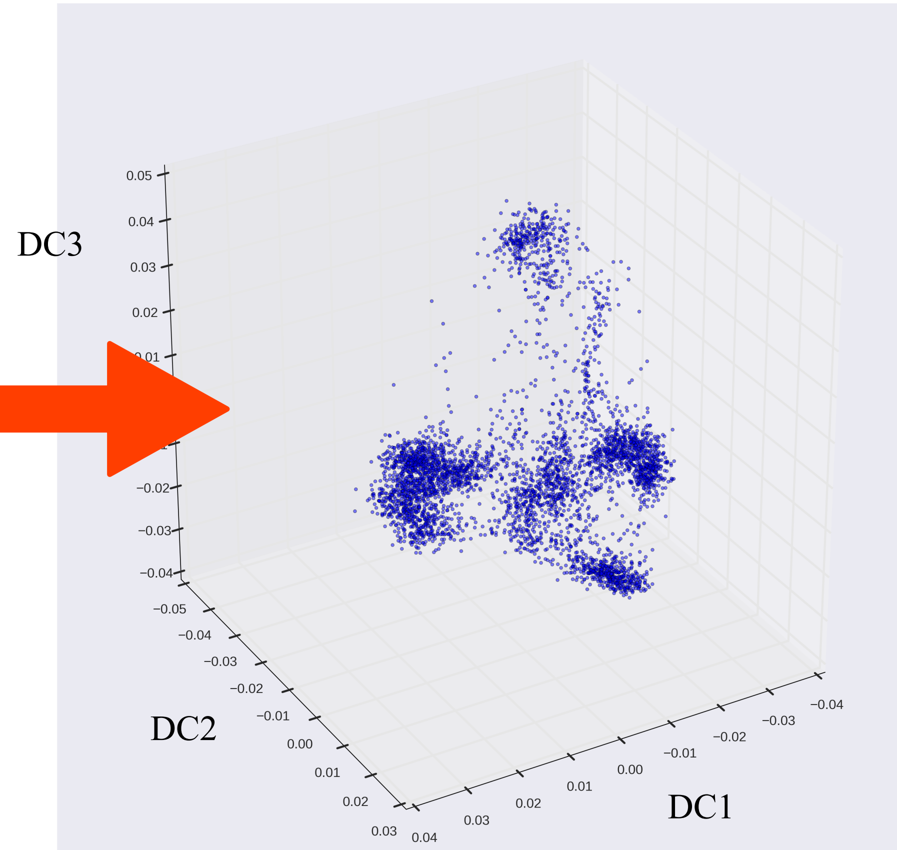
Roberto Covino
and Gerhard Hummer
Max-Planck-Institute of Biophysics

mpibp
max-planck-institut
für biophysik

Mga2 dimer in the membrane of the Endoplasmic Reticulum of S. cerevisiae

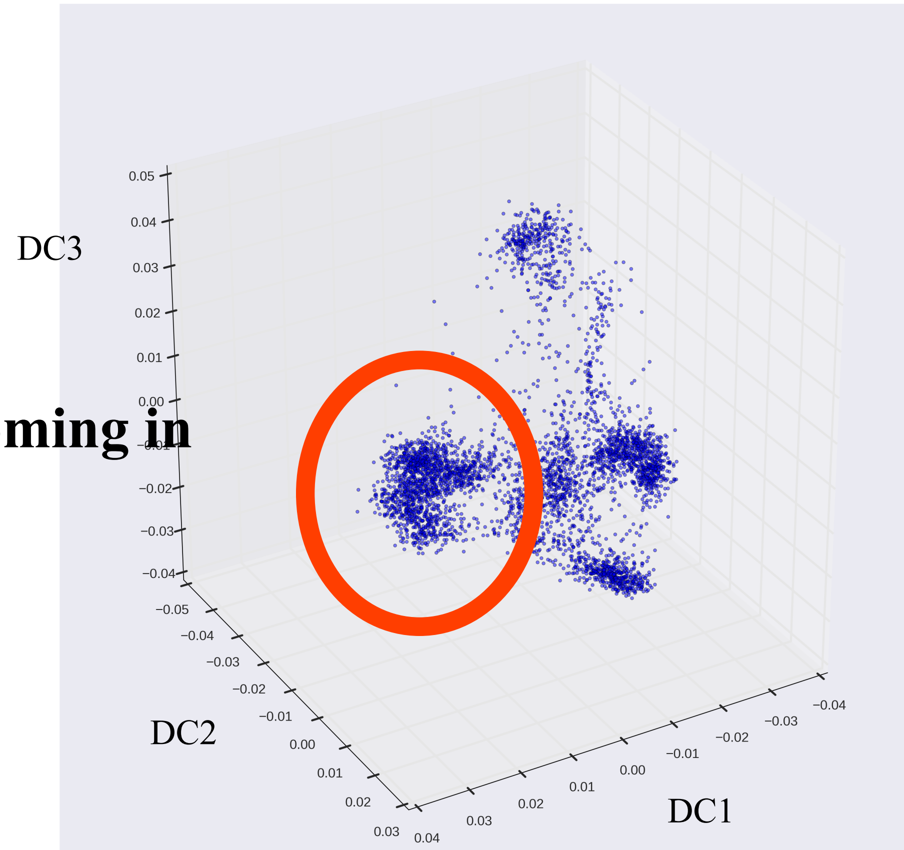
Mga2 dimer dynamics





Dynamics from long MD simulation projected on the first 3 diffusion coordinates

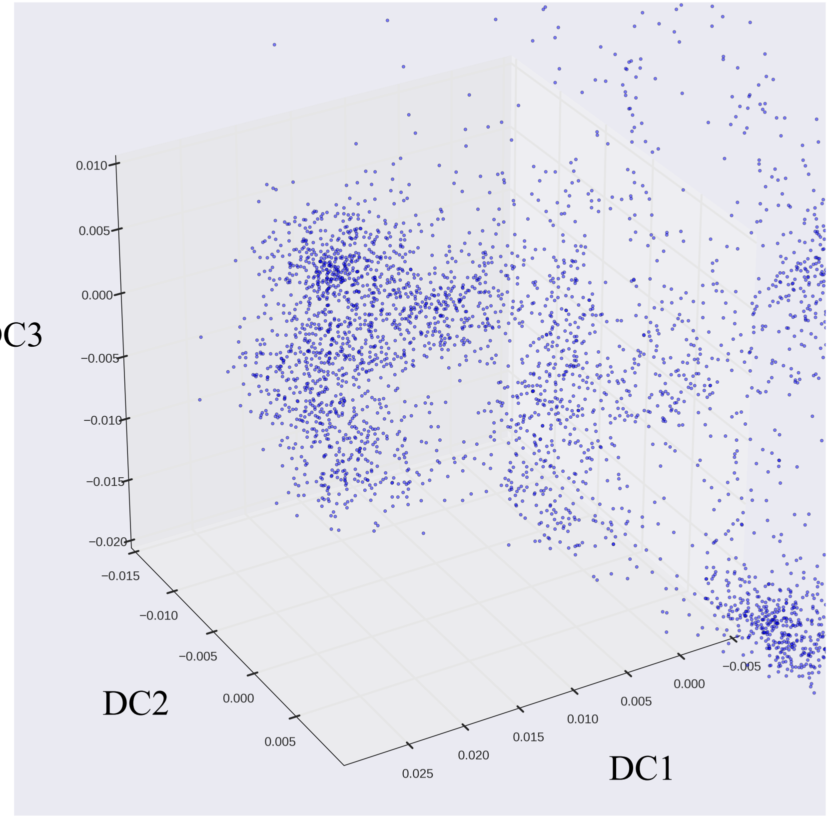
Mga2 dimer



Zooming in

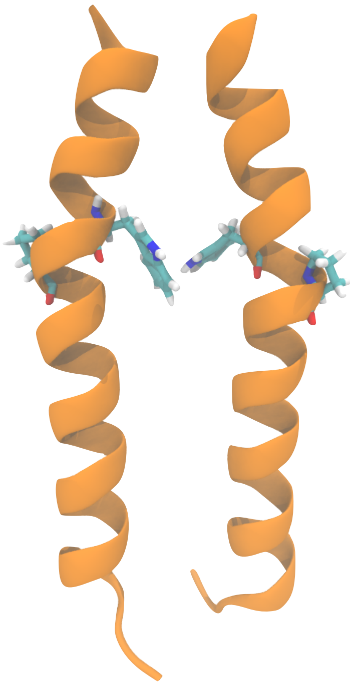
Dynamics from long MD simulation projected on the first 3 diffusion coordinates

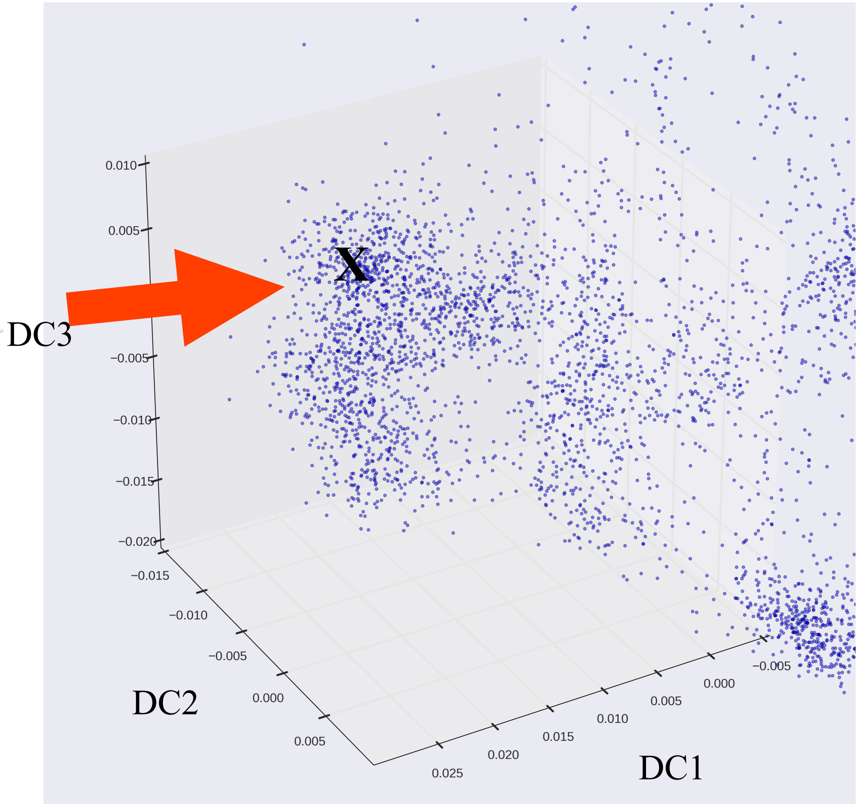
What if we visited only a small region?



We focus only on a small subset and pretend we do not know about the rest of the "world"

What if we visited only a small region?

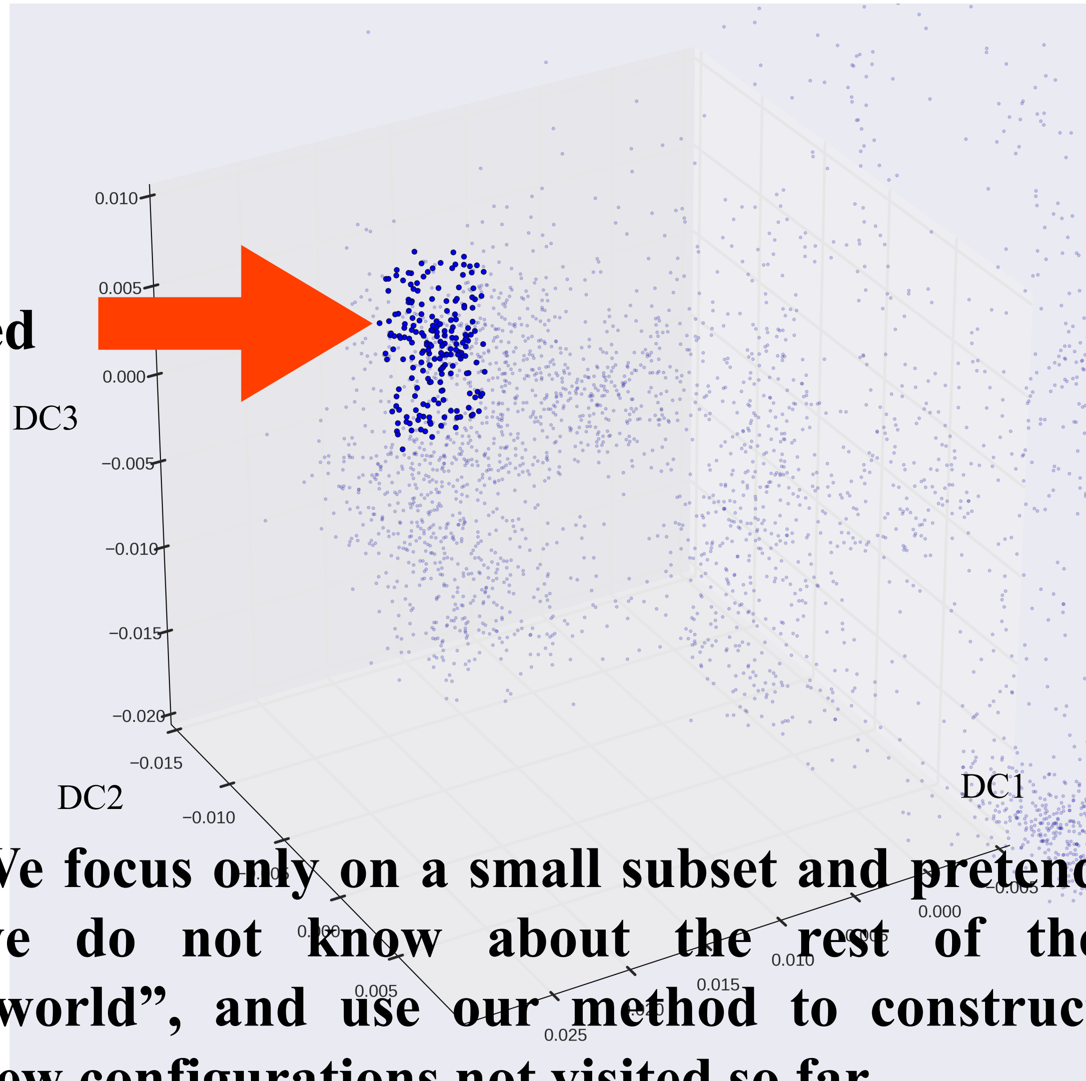




We focus only on a small subset and pretend we do not know about the rest of the "world"

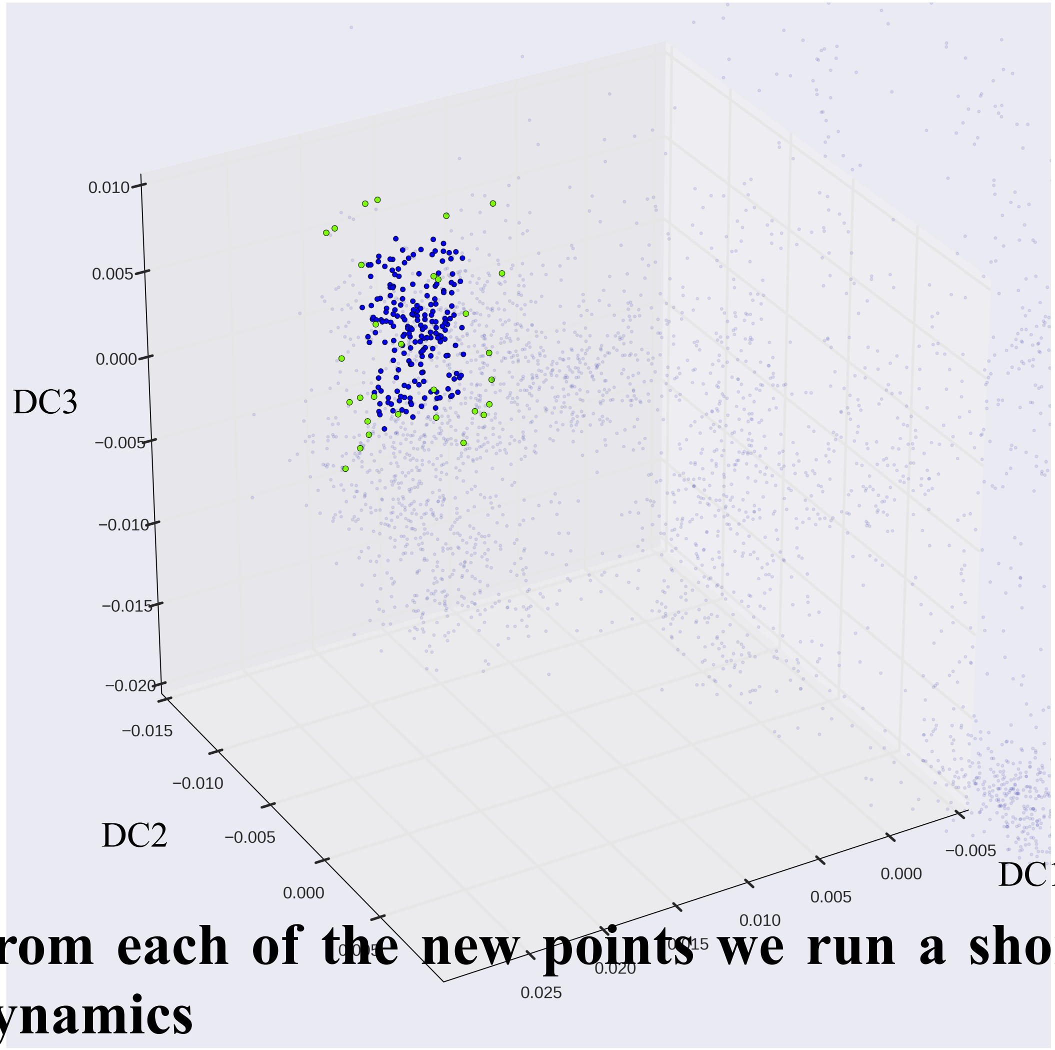
What if we visited only a small region?

Let us pretend we have explored only this region



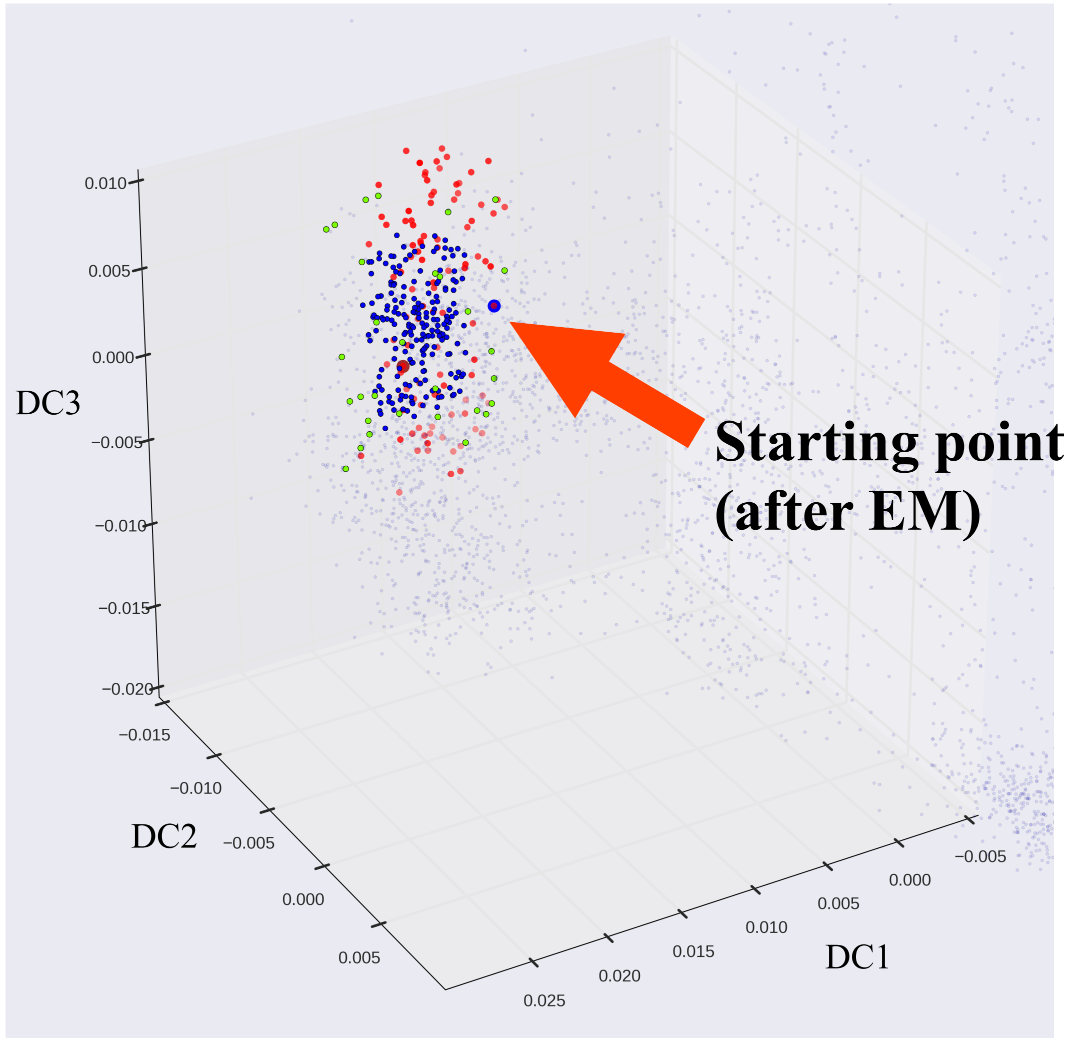
We focus only on a small subset and pretend we do not know about the rest of the "world", and use our method to construct new configurations not visited so far.

We can "predict" new structures

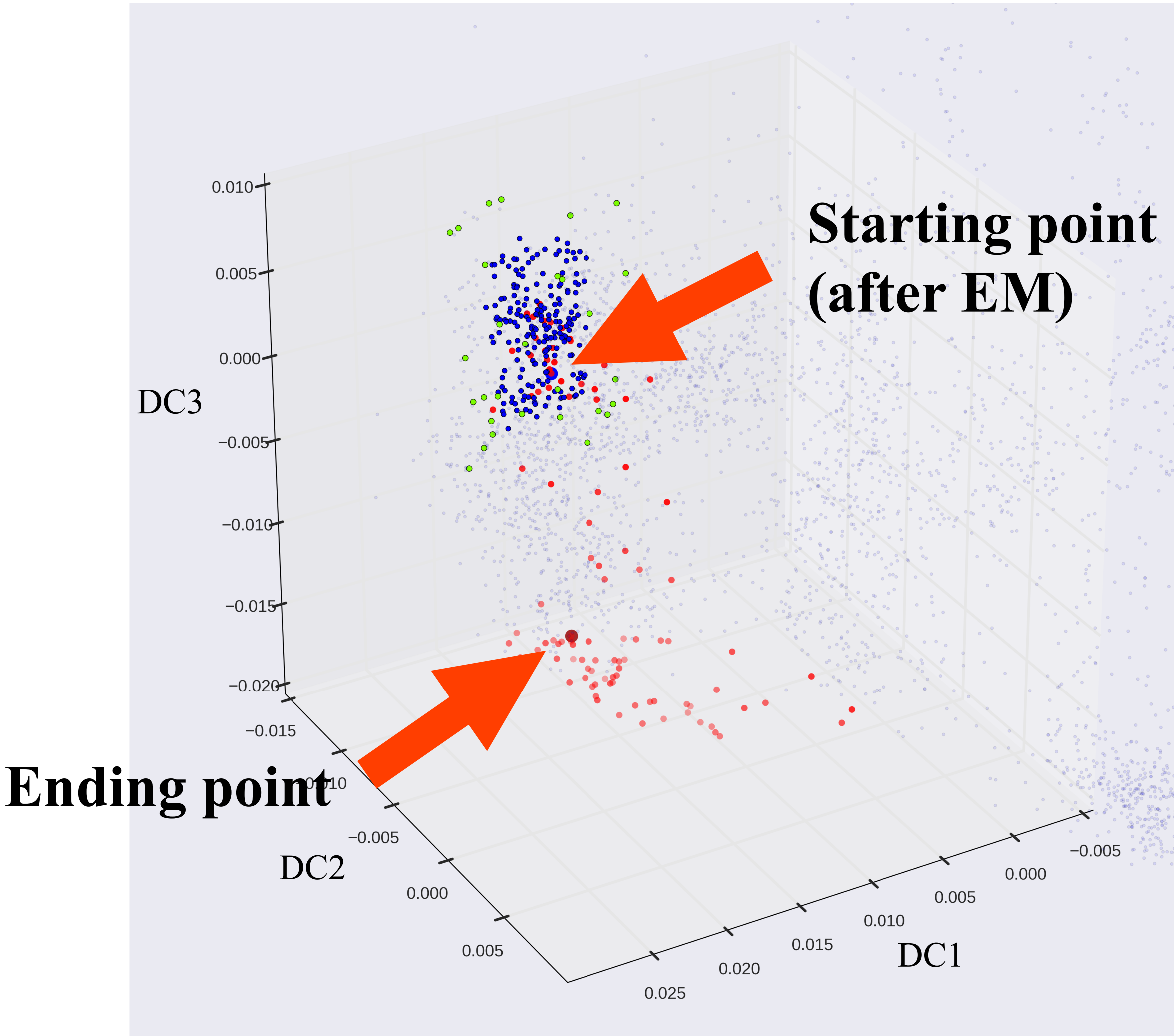


From each of the new points we run a short dynamics

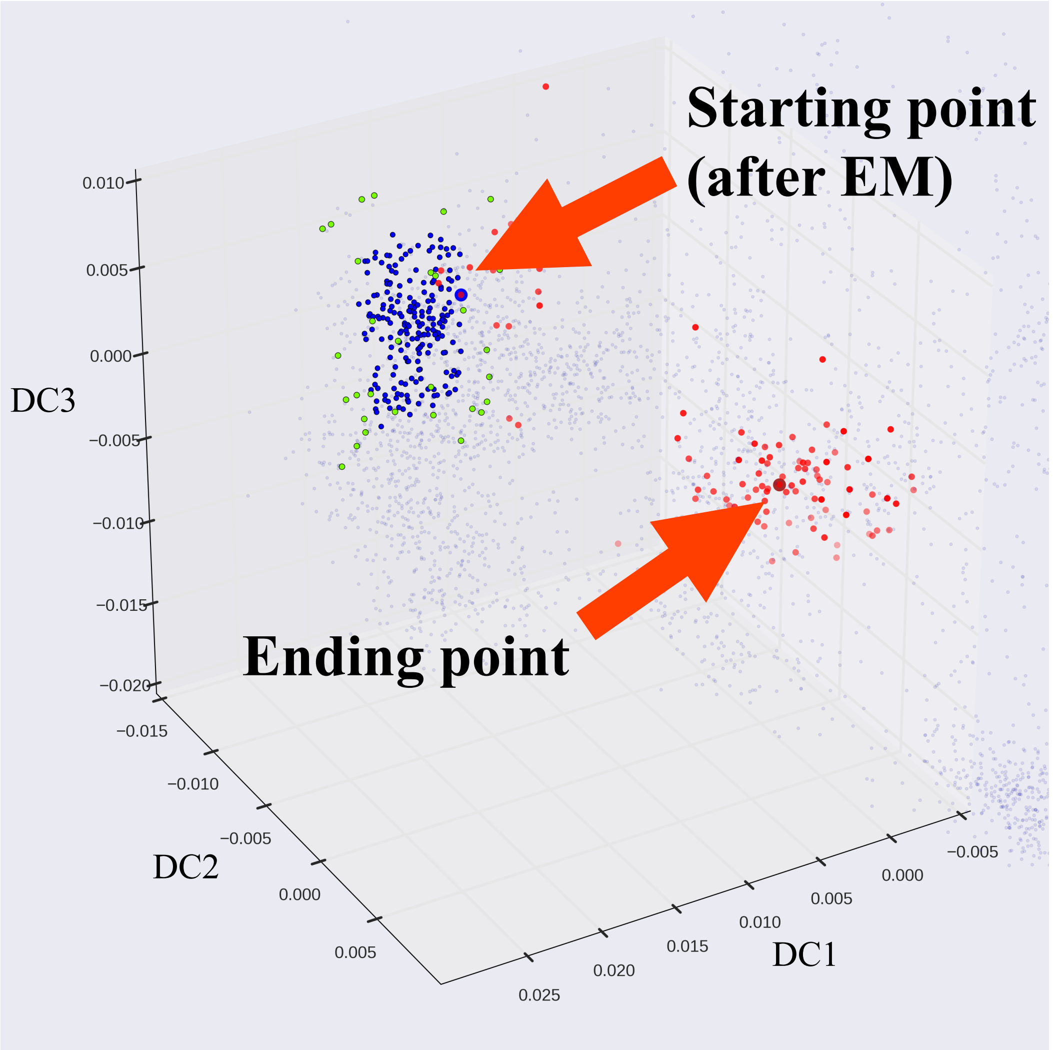
Some will just come back



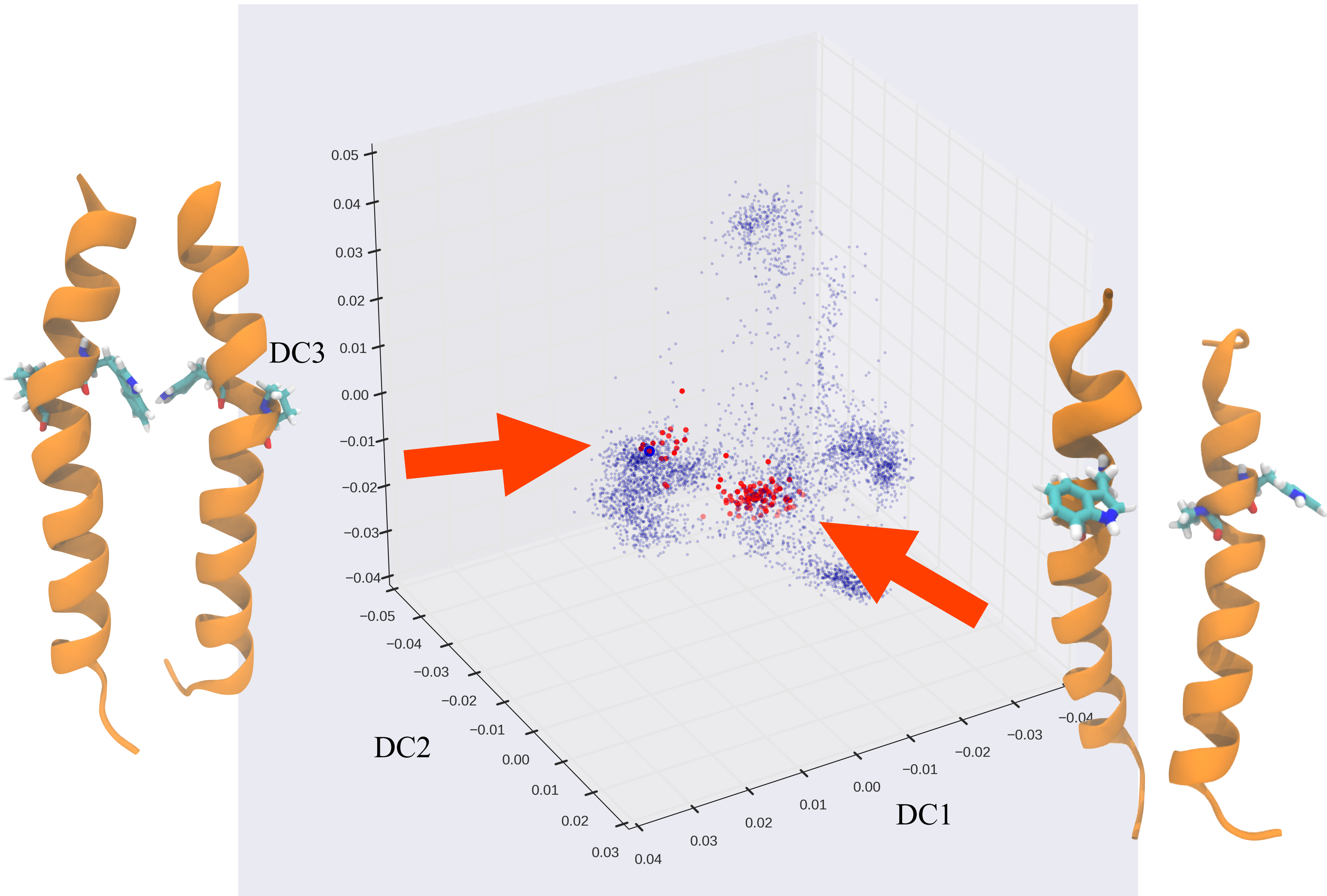
Some will stay in the local region



Some will leave and explore further

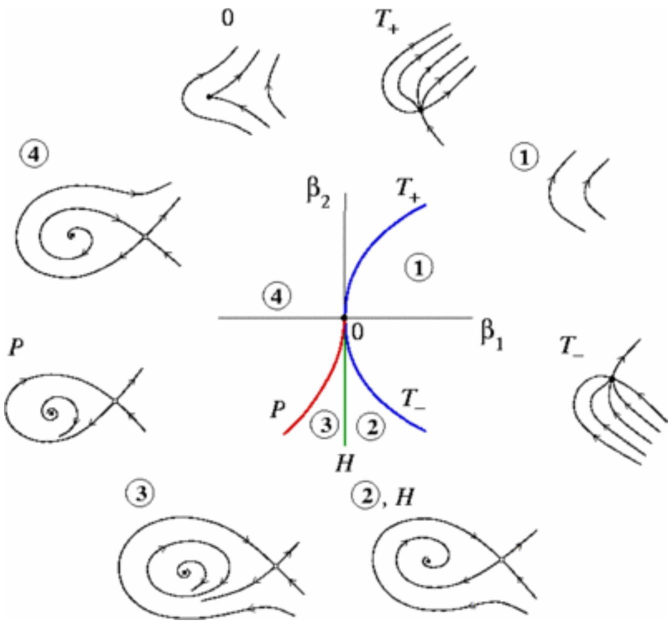


And we will “discover” new structures



Leap of Faith

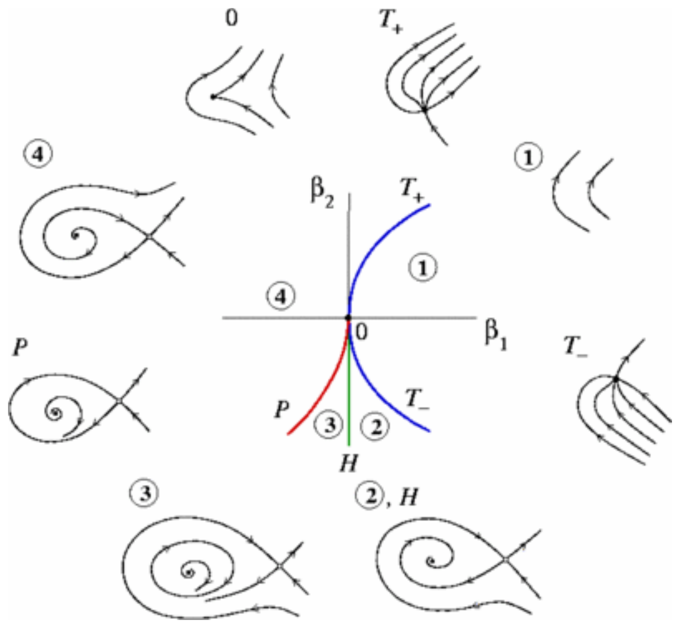
Takens-Bogdanov (double zero) singularity



- planar system

- dx/dt = y
- dy/dt = $\beta 1+\beta 2$ * x + x ** 2 − x * y

Takens-Bogdanov (double zero) singularity



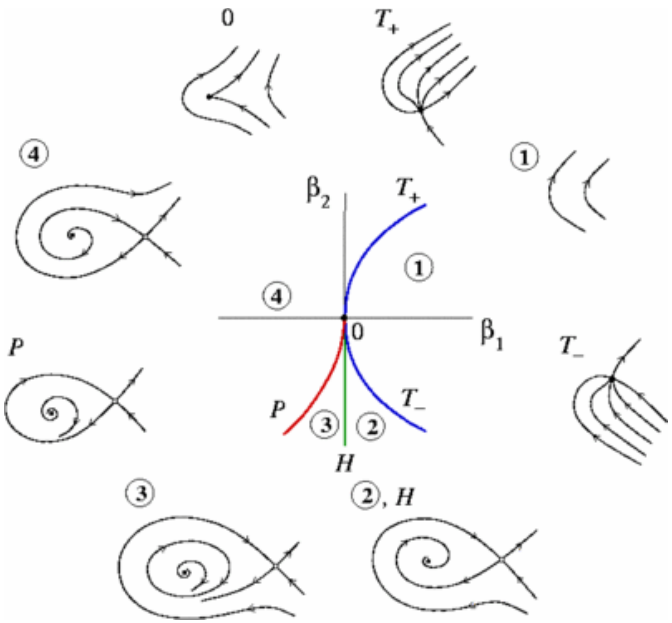
INPUTS

dx/dt = f(x,p)

y=G(x)

OUTPUTS, y(t)

Takens-Bogdanov (double zero) singularity



INPUTS

black box

OUTPUTS, y(t)

Working for Dr. Reid

"Read this and lets talk about it."

J.S.I.A.M. Control
Ser. A, Vol. 1, No. 2
Printed in U.S.A., 1963

Burns

MATHEMATICAL DESCRIPTION OF LINEAR DYNAMICAL SYSTEMS*

R. E. KALMAN†

Abstract. There are two different ways of describing dynamical systems: (i) by means of state variables and (ii) by input/output relations. The first method may be regarded as an axiomatization of Newton's laws of mechanics and is taken to be the basic definition of a system.

It is then shown (in the linear case) that the input/output relations determine only one part of a system, that which is completely observable and completely controllable. Using the theory of controllability and observability, methods are given for calculating irreducible realizations of a given impulse-response matrix. In particular, an explicit procedure is given to determine the minimal number of state variables necessary to realize a given transfer-function matrix. Difficulties arising from the use of reducible realizations are discussed briefly.

1. Introduction and summary. Recent developments in optimal control system theory are based on vector differential equations as models of physical systems. In the older literature on control theory, however, the same systems are modeled by transfer functions (i.e., by the Laplace transforms of the differential equations relating the inputs to the outputs). Two different languages have arisen, both of which purport to talk about the same problem. In the new approach, we talk about state variables, transition equations, etc., and make constant use of abstract linear algebra. In the old approach, the key words are frequency response, pole-zero patterns, etc., and the main mathematical tool is complex function theory.

Is there really a difference between the new and the old? Precisely what are the relations between (linear) vector differential equations and transfer-functions? In the literature, this question is surrounded by confusion [1]. This is bad. Communication between research workers and engineers is impeded. Important results of the "old theory" are not yet fully integrated into the new theory.

In the writer's view—which will be argued at length in this paper—the difficulty is due to insufficient appreciation of the concept of a *dynamical system*. Control theory is supposed to deal with physical systems, and not merely with mathematical objects such as a differential equation or a transfer function. We must therefore pay careful attention to the relationship between physical systems and their representation via differential equations, transfer functions, etc.

* Received by the editors July 7, 1962 and in revised form December 9, 1962. Presented at the Symposium on Multivariable System Theory, SIAM, November 1,

To clear up these issues, we need first of all a precise, abstract definition of a (physical) dynamical system. (See sections 2–3.) The axioms which provide this definition are generalizations of the Newtonian world-view of causality. They have been used for many years in the mathematical literature of dynamical systems. Just as Newtonian mechanics evolved from differential equations, these axioms seek to abstract those properties of differential equations which agree with the "facts" of classical physics. It is hardly surprising that under special assumptions (finite-dimensional state space, continuous time) the axioms turn out to be equivalent to a system of ordinary differential equations. To avoid mathematical difficulties, we shall restrict our attention to linear differential equations.

In section 4 we formulate the central problem of the paper:

Given an (experimentally observed) impulse response matrix, how can we identify the linear dynamical system which generated it?

We propose to call any such system a *realization* of the given impulse response. It is an *irreducible realization* if the dimension of its state space is minimal.

Section 5 is a discussion of the "canonical structure theorem" [2, 14] which describes abstractly the coupling between the external variables (input and output) and the internal variables (state) of any linear dynamical system. As an immediate consequence of this theorem, we find that *a linear dynamical system is an irreducible realization of an impulse-response matrix if and only if the system is completely controllable and completely observable.* This important result provides a link between the present paper and earlier investigations in the theory of controllability and observability [3–5].

Explicit criteria for complete controllability and complete observability are reviewed in a convenient form in section 6.

Section 7 provides a constructive computational technique for determining the canonical structure of a constant linear dynamical system.

In section 8 we present, probably for the first time, a complete and rigorous theory of how to define the state variables of a multi-input/multi-output constant linear dynamical system described by its transfer-function matrix. Since we are interested only in irreducible realizations, there is a certain unique, well-defined number n of state variables which must be used. We give a simple proof of a recent theorem of Gilbert [5] concerning the value of n. We give canonical forms for irreducible realizations in simple cases. We give a constructive procedure (with examples) for finding an irreducible realization in the general case.

Many errors have been committed in the literature of system theory by carelessly regarding transfer functions and systems as equivalent concepts. A list of these has been collected in section 9.

The field of research outlined in this paper is still wide open, except

In 1981, a miracle happens

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. AC-26, NO. 1, FEBRUARY 1981

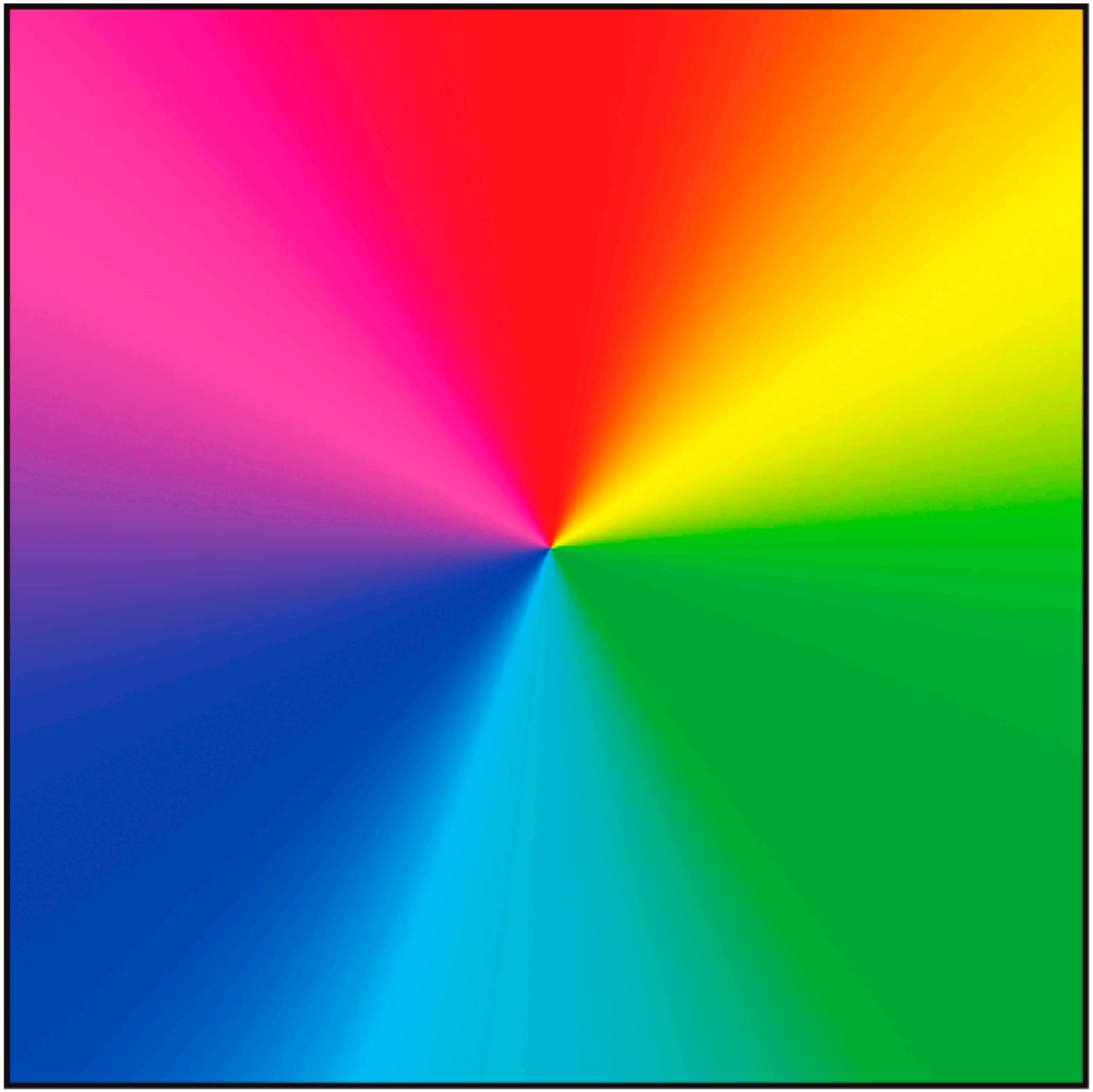
Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction

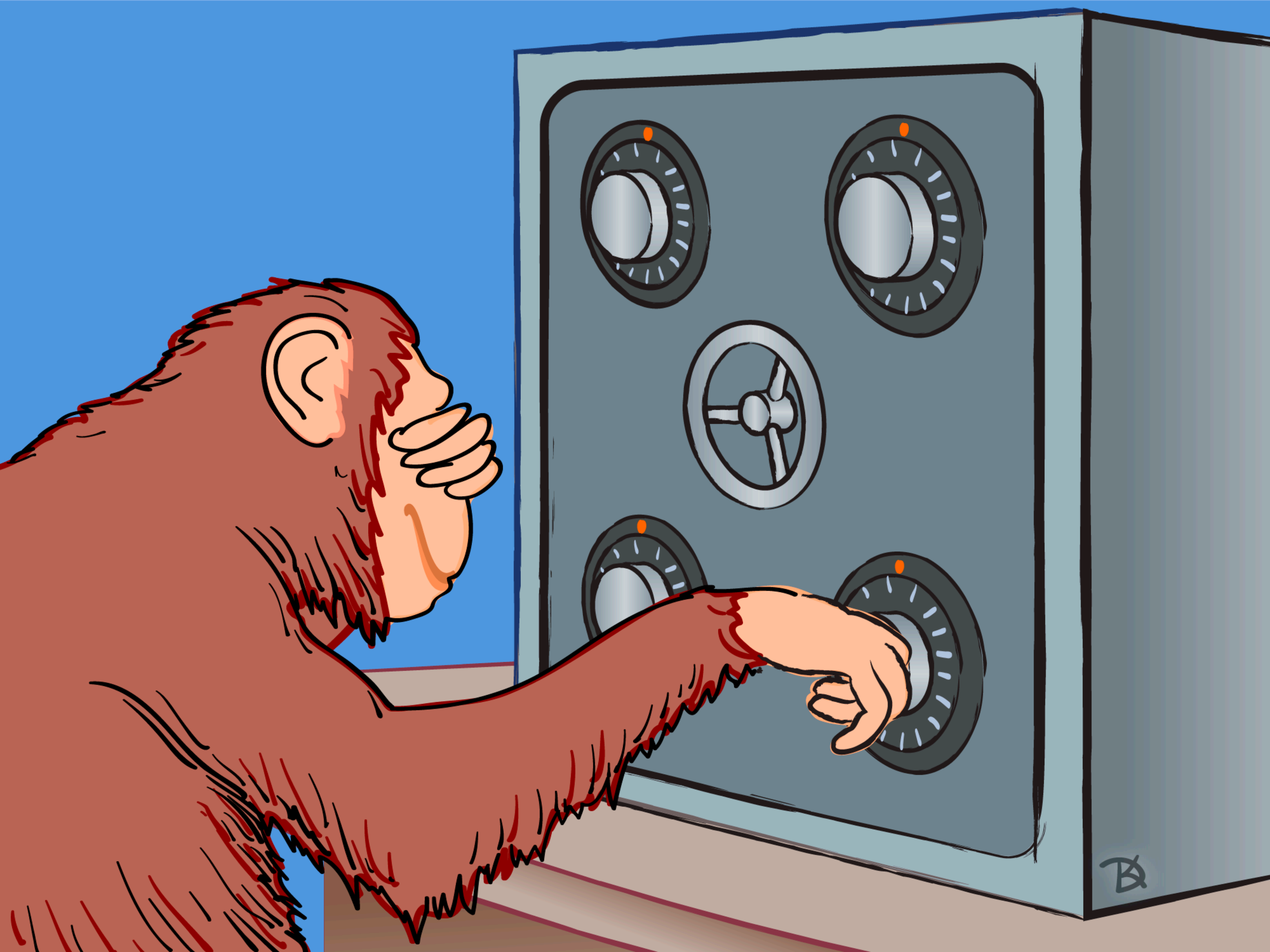
BRUCE C. MOORE

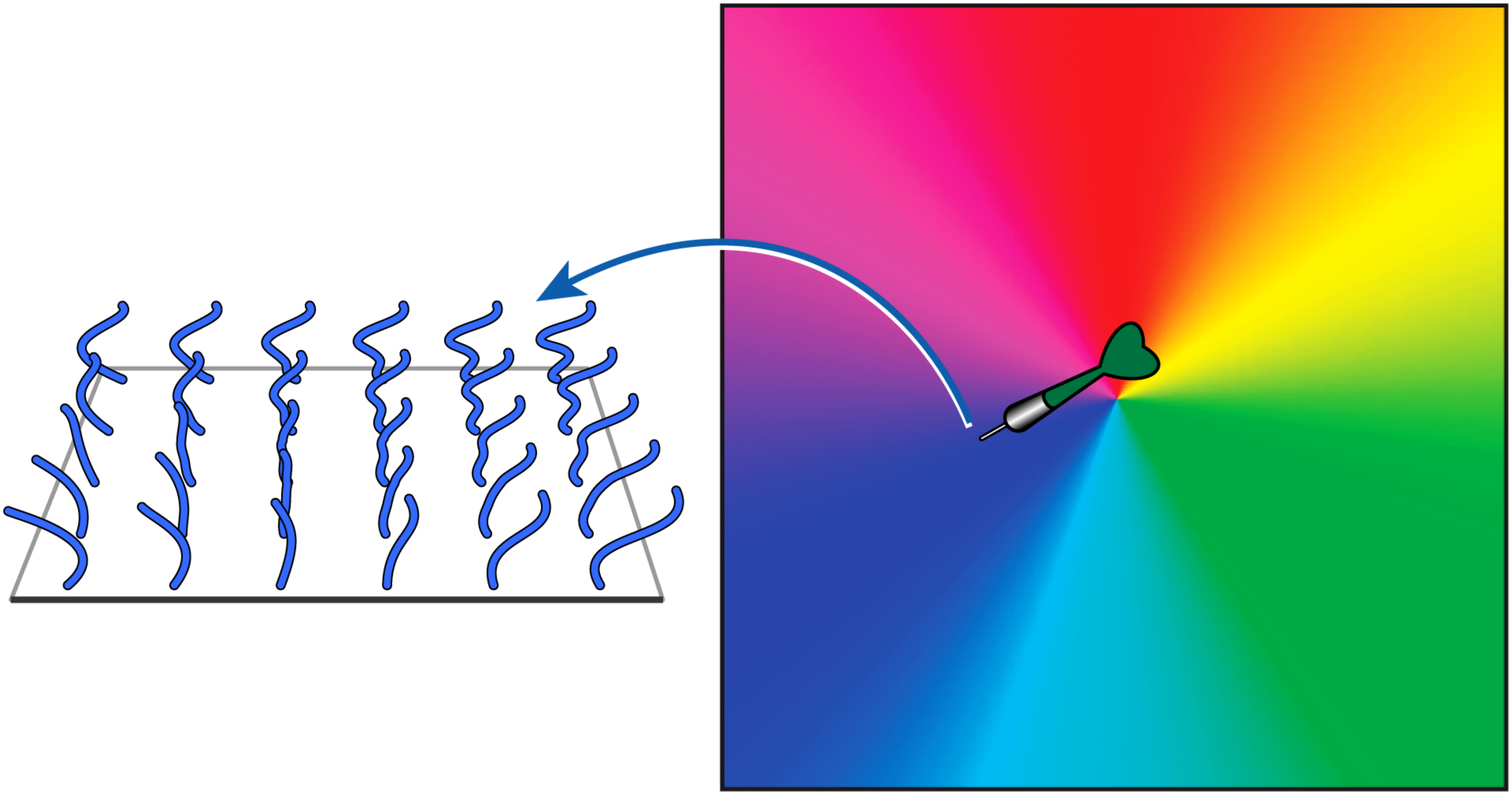
Abstract—Kalman's minimal realization theory involves geometric objects (controllable, unobservable subspaces) which are subject to structural instability. Specifically, arbitrarily small perturbations in a model may cause a change in the dimensions of the associated subspaces. This situation is manifested in computational difficulties which arise in attempts to apply textbook algorithms for computing a minimal realization.

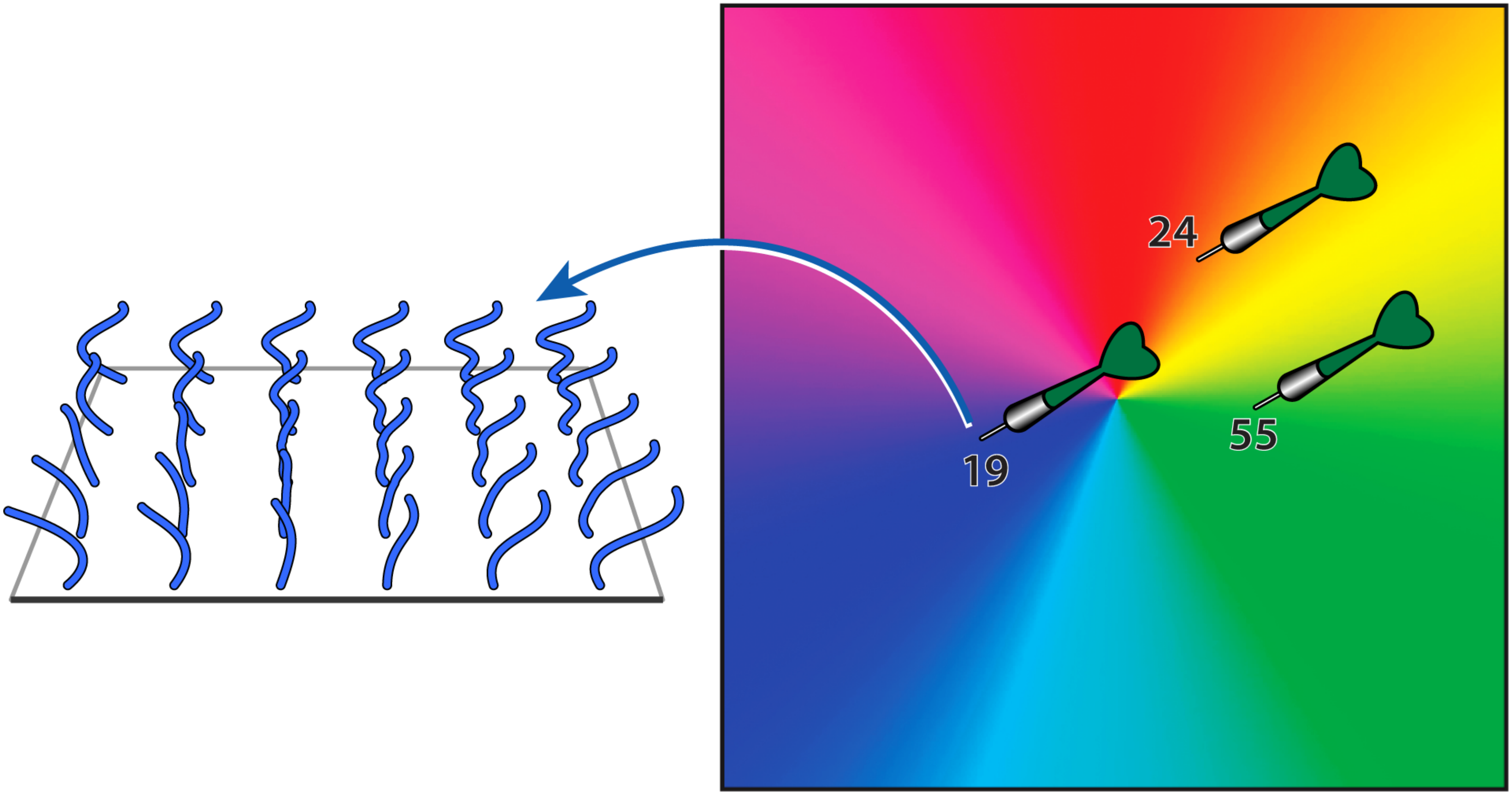
Structural instability associated with geometric theories is not unique to control; it arises in the theory of linear equations as well. In this setting, the computational problems have been studied for decades and excellent tools have been developed for coping with the situation. One of the main goals of this paper is to call attention to *principal component analysis* (Hotelling, 1933), and an algorithm (Golub and Reinsch, 1970) for computing the *singular value decomposition* of a matrix. Together they form a powerful tool for coping with structural instability in dynamic systems.

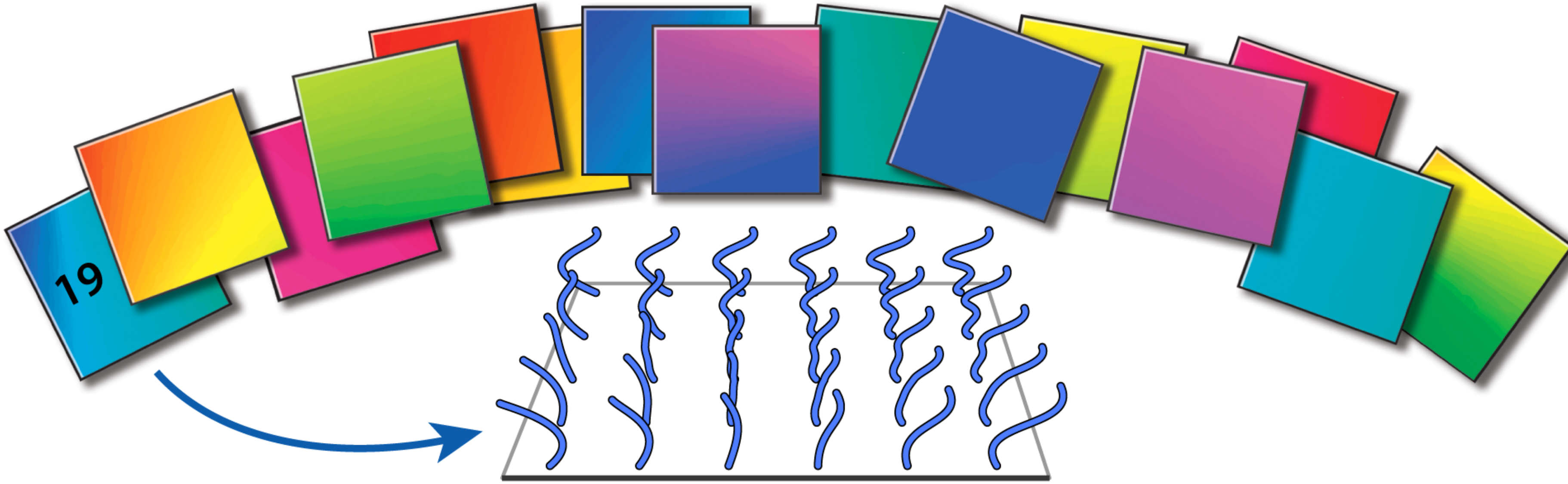
As developed in this paper, principal component analysis is a technique for analyzing signals. (Singular value decomposition provides the computational machinery.) For this reason, Kalman's minimal realization theory is recast in terms of responses to injected signals. Application of the signal analysis to controllability and observability leads to a coordinate system in which the "internally balanced" model has special properties. For asymptotically stable systems, this yields working approximations of X_c, $X_{\bar{o}}$, the controllable and unobservable subspaces. It is proposed that a natural first step in model reduction is to apply the mechanics of minimal realization using these working subspaces.

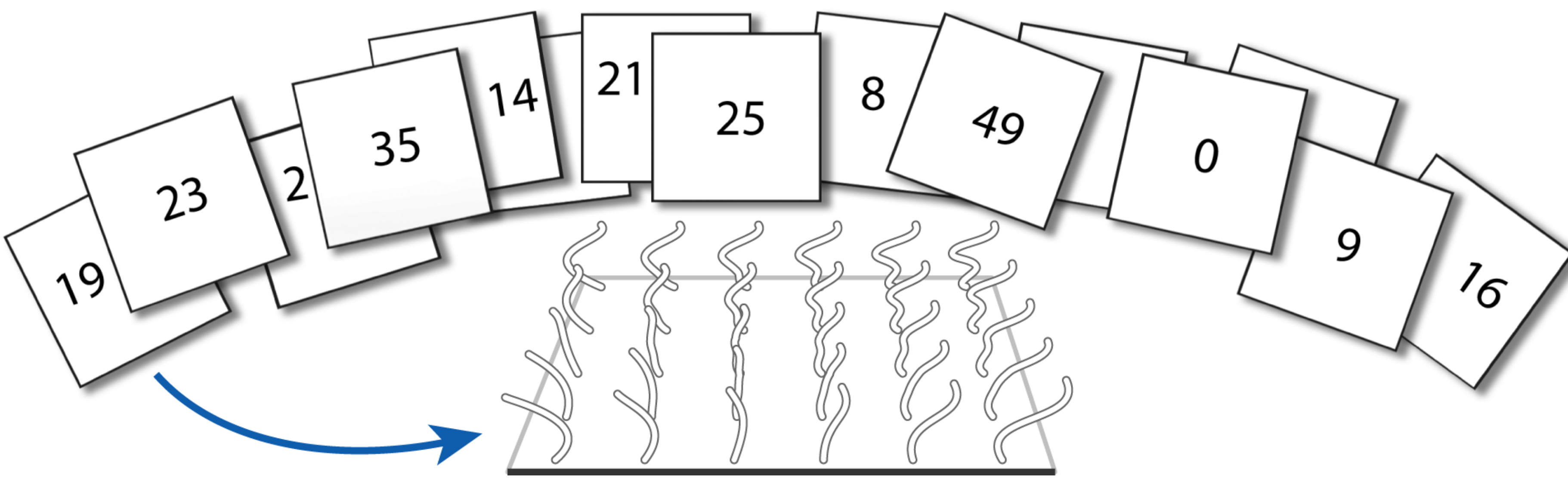


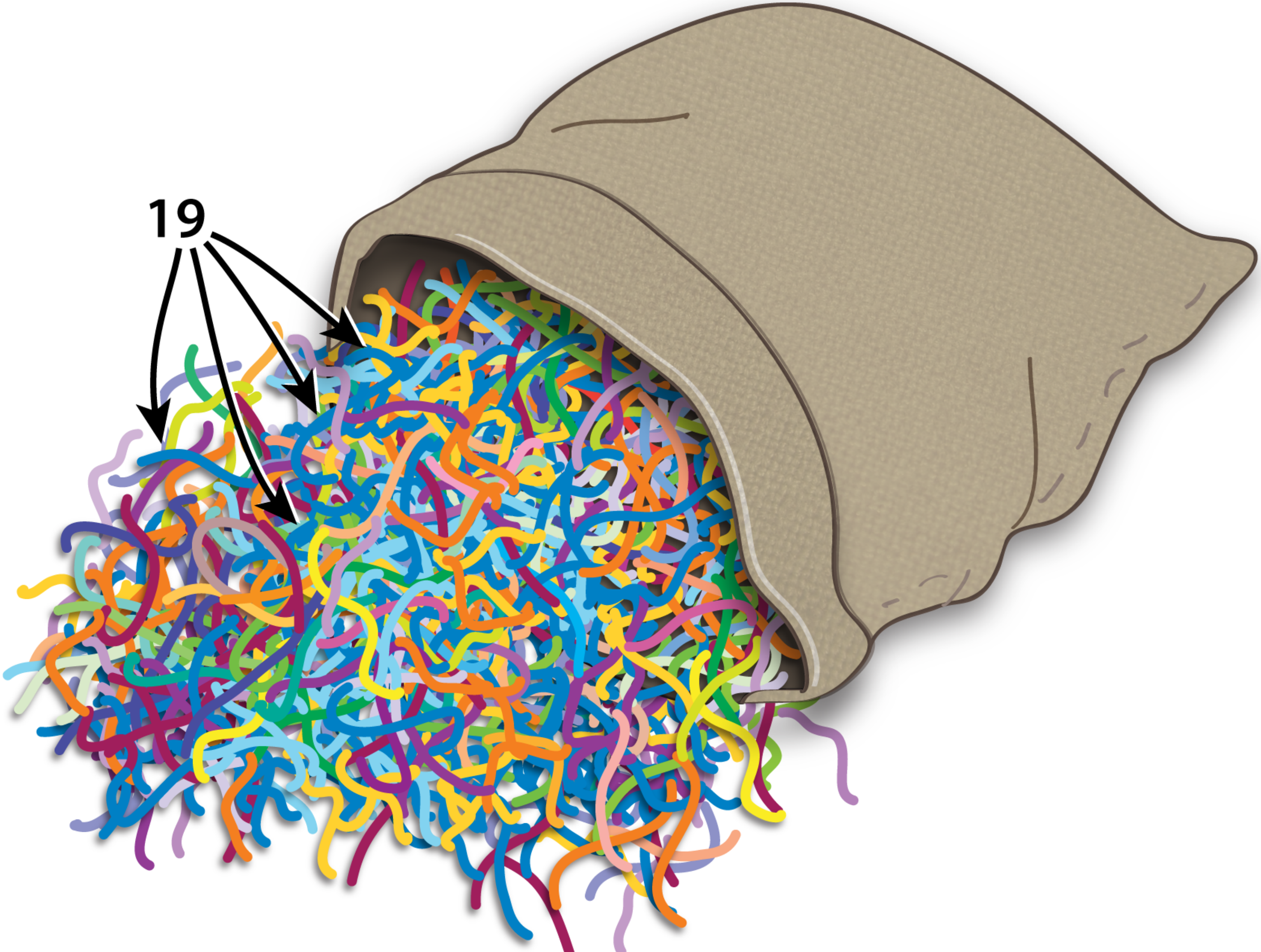


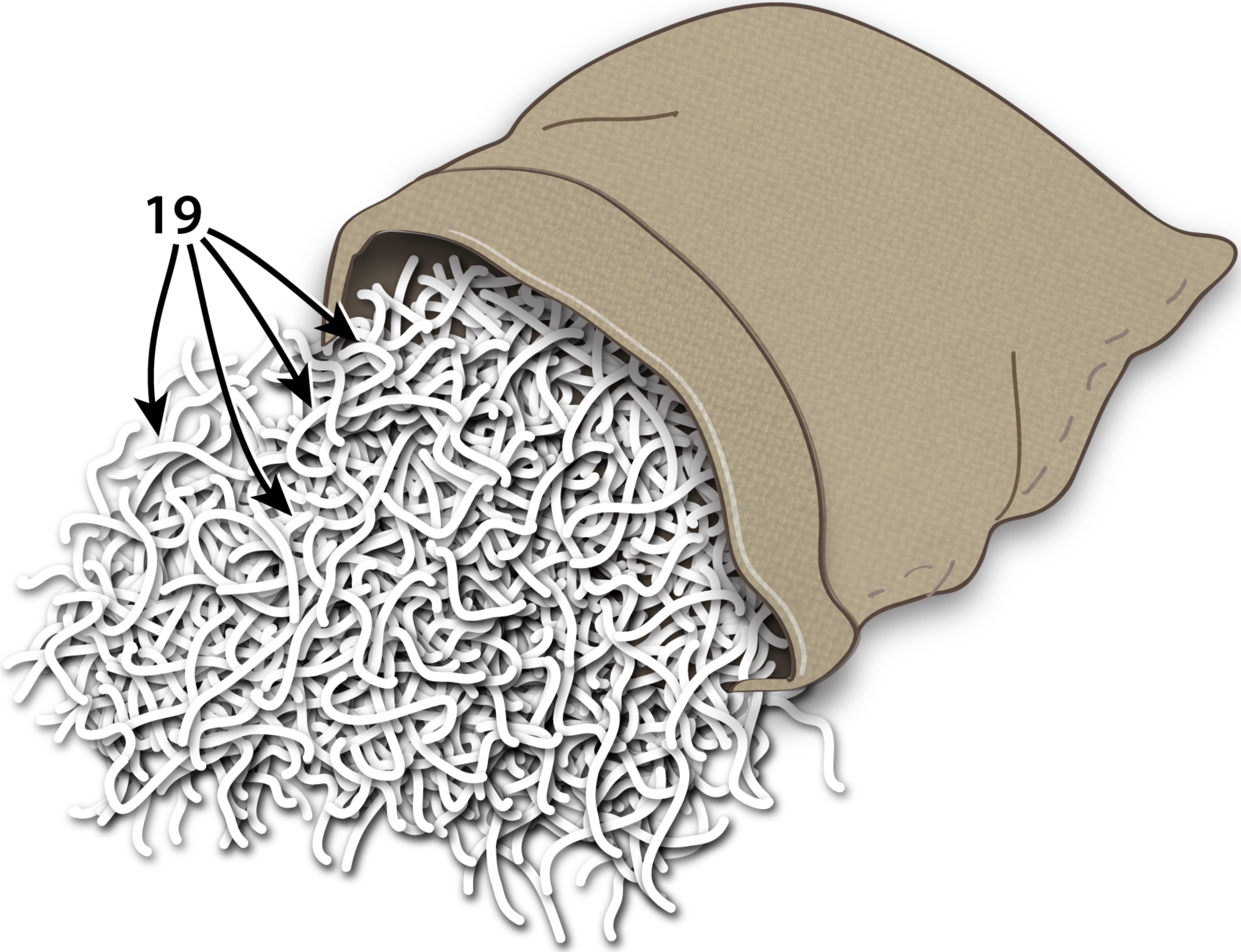


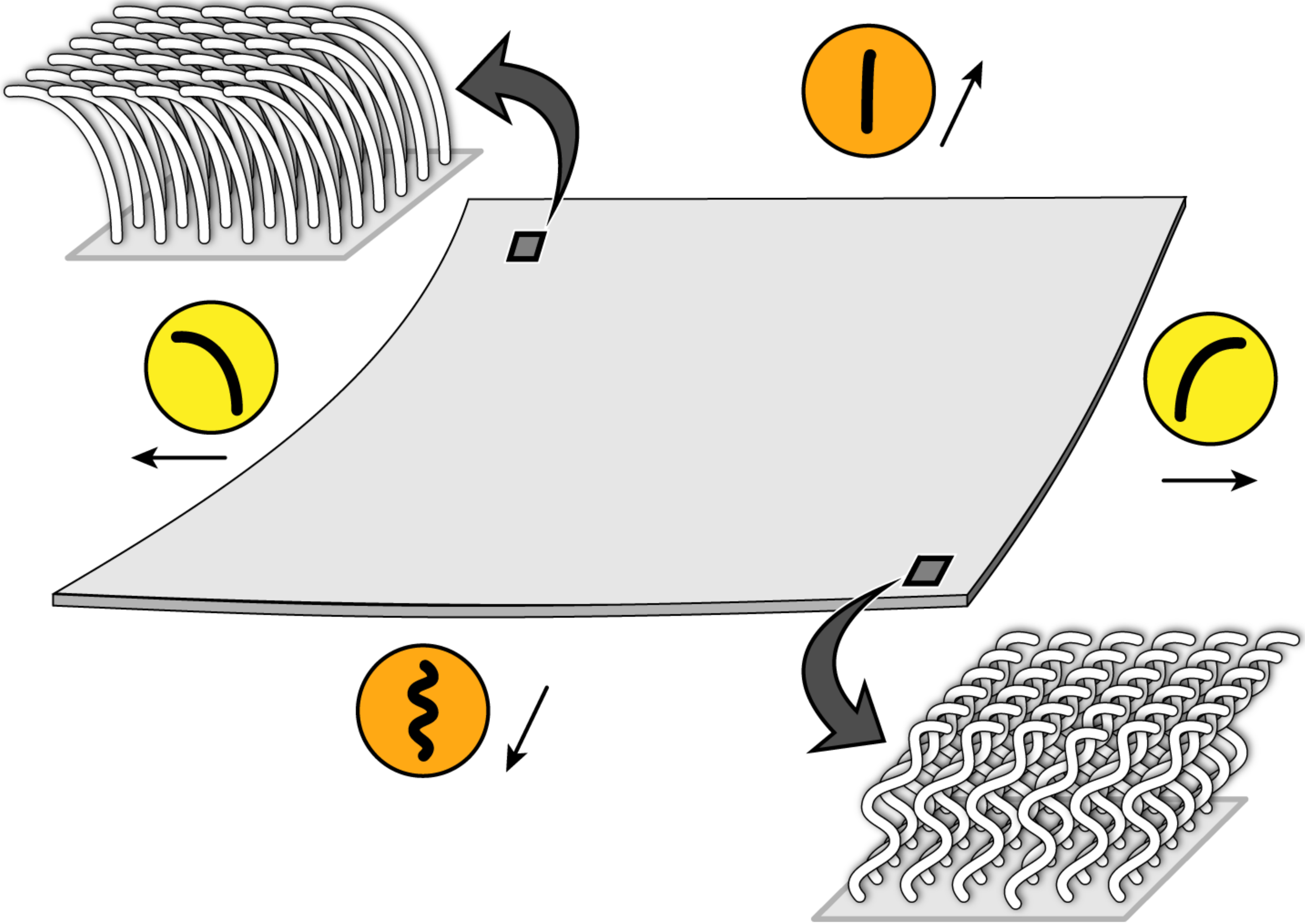

24
55
19

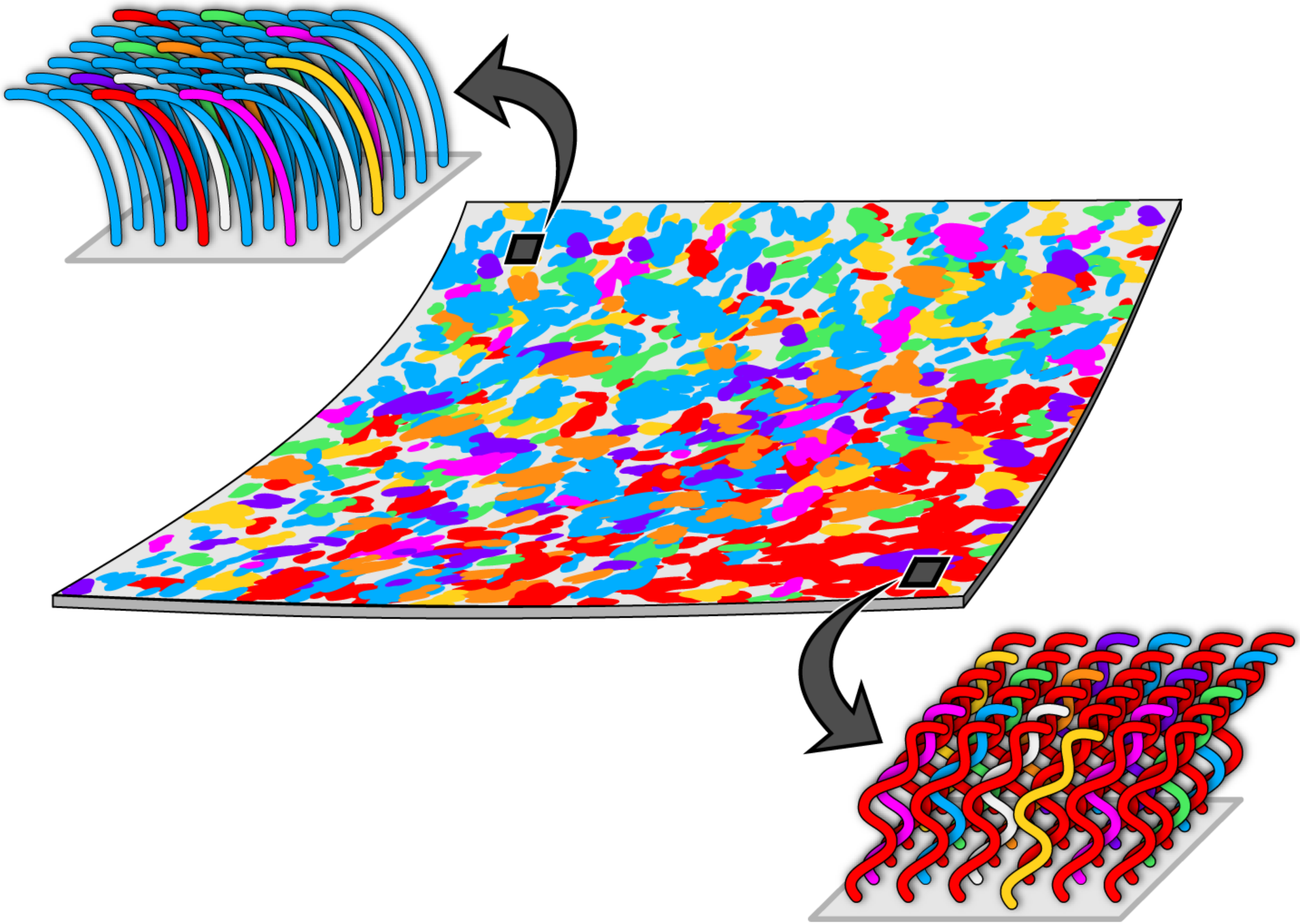

19

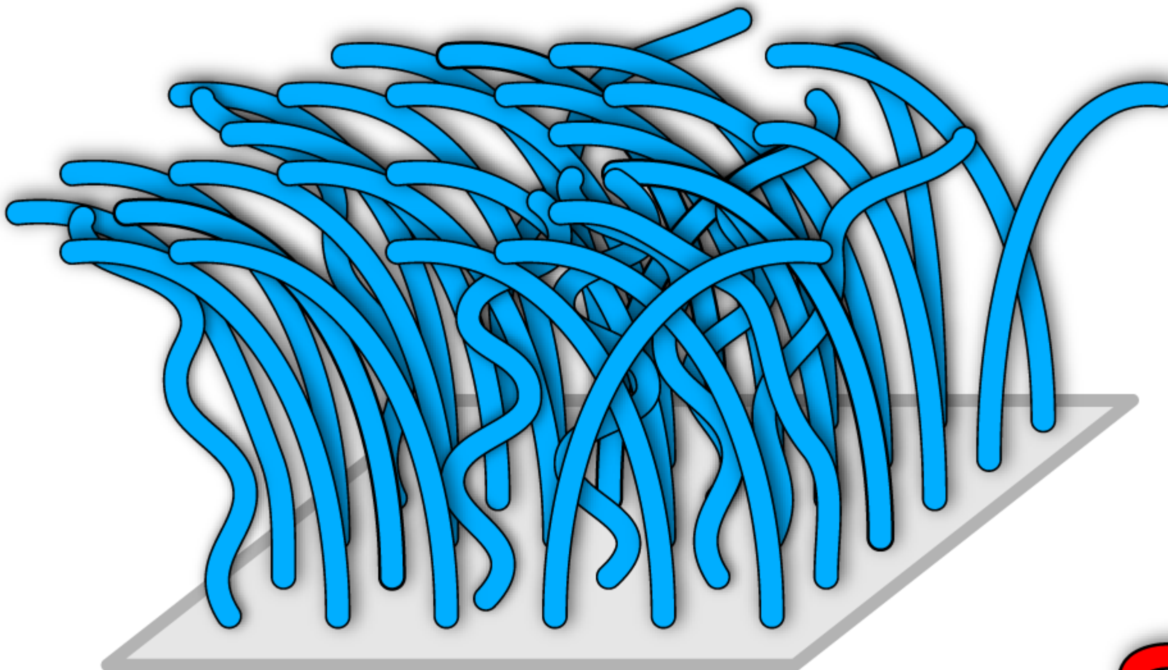

19
23
2
35
14
21
25
8
49
0
9
16

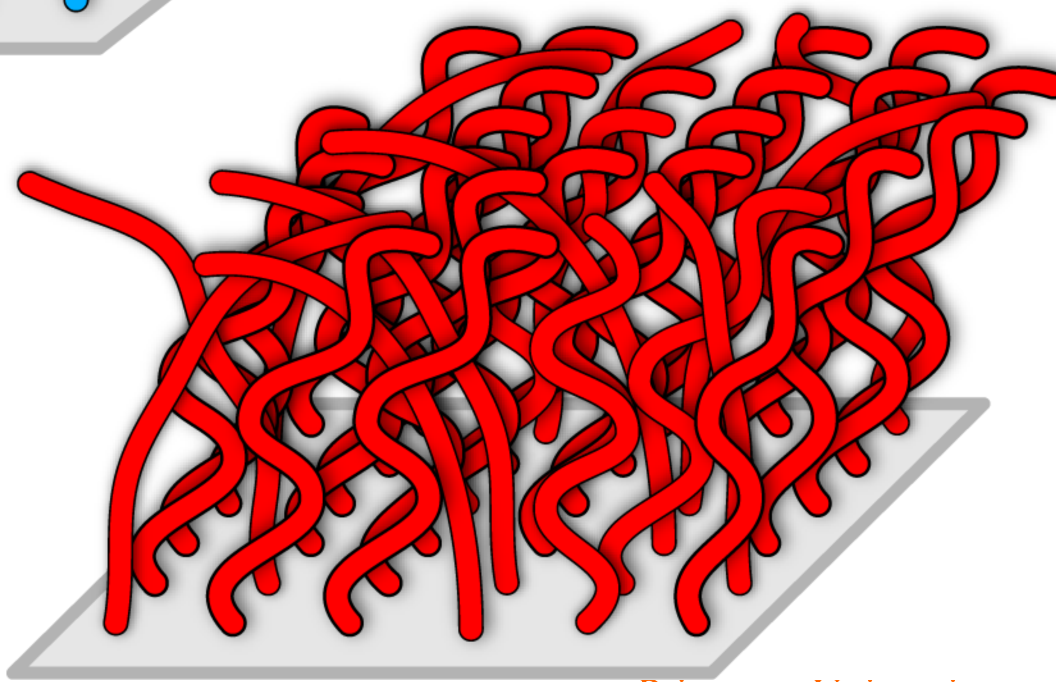

19


19

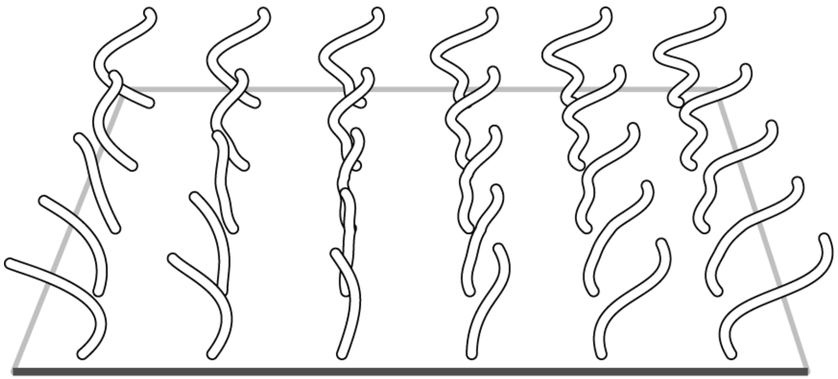




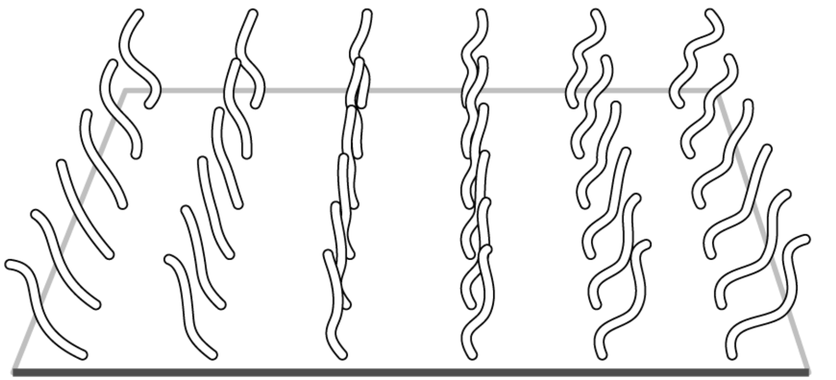




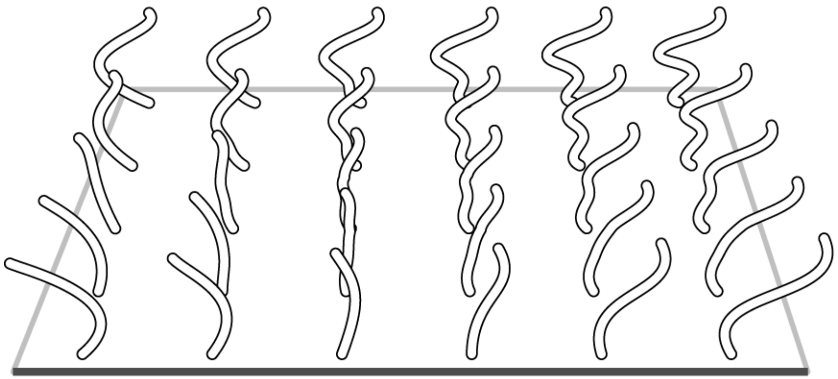
19



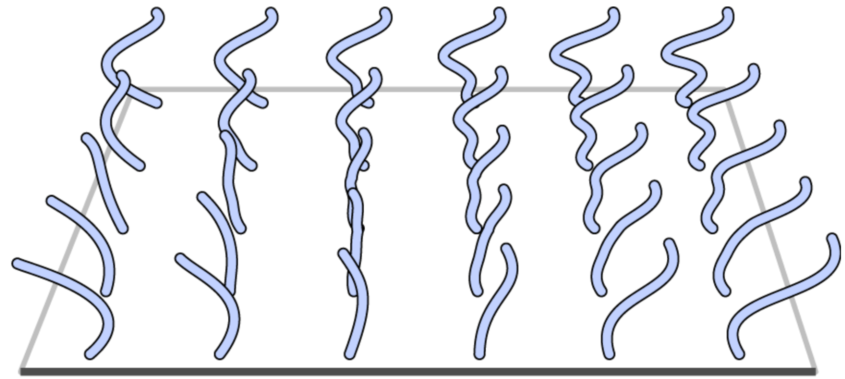
24



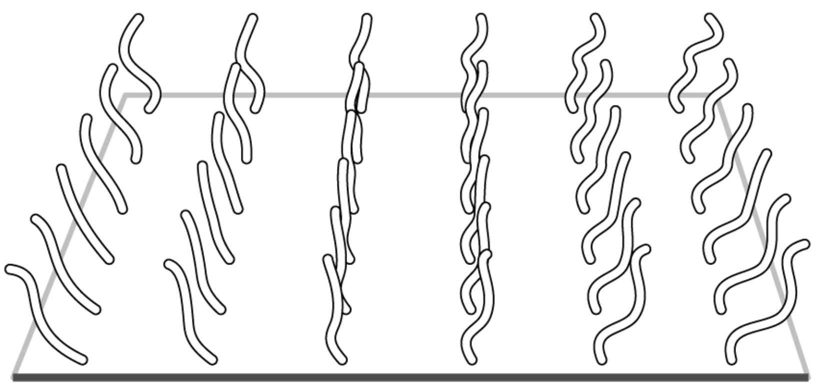
19



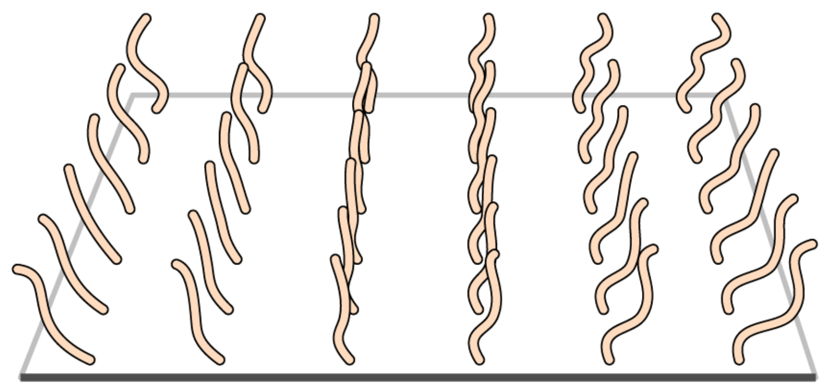
19



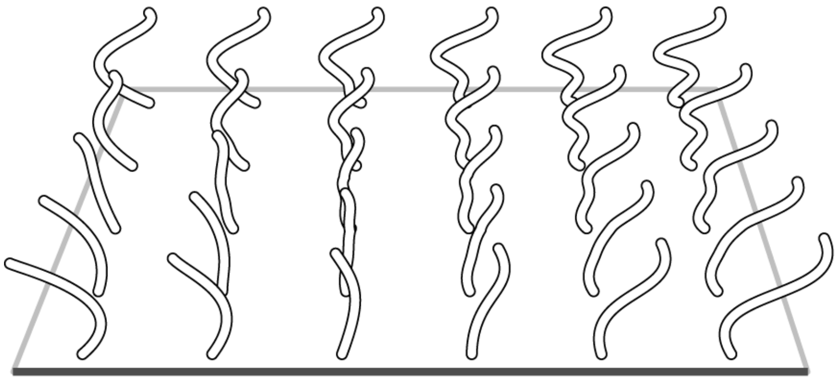
24



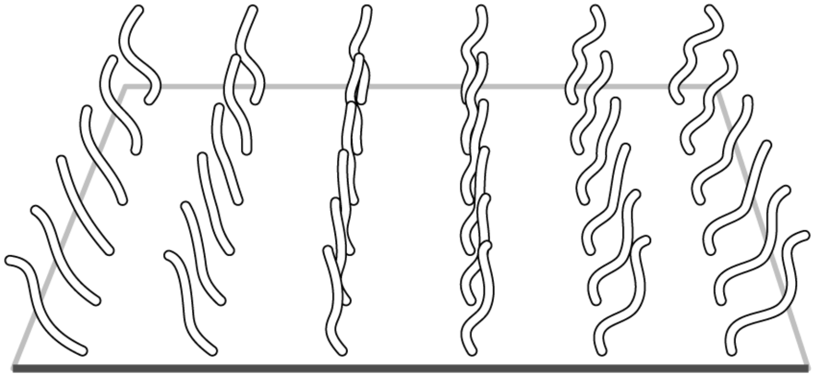
24



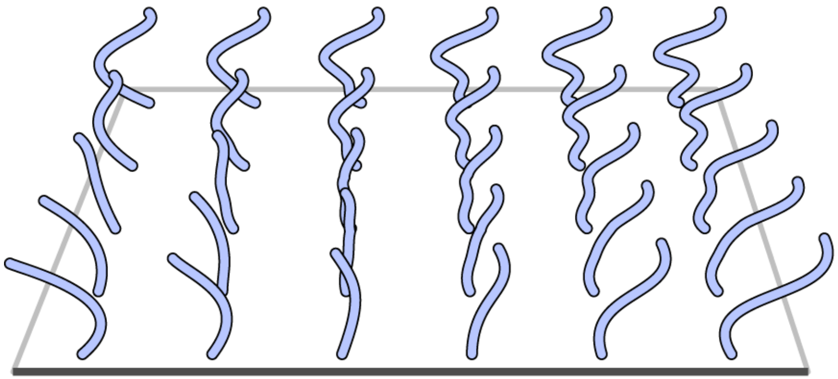
19



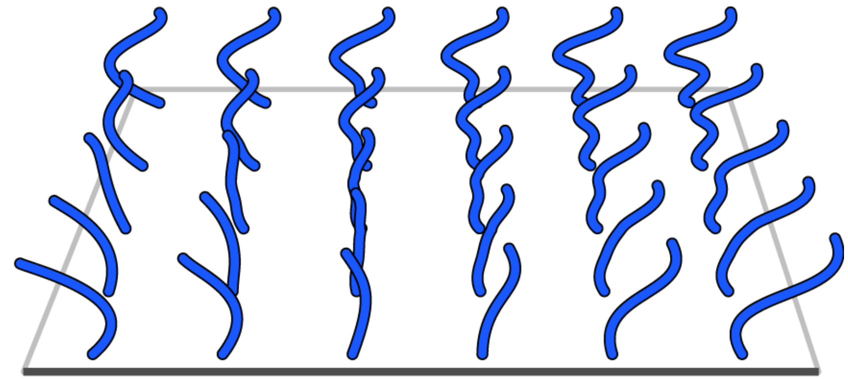
24



19

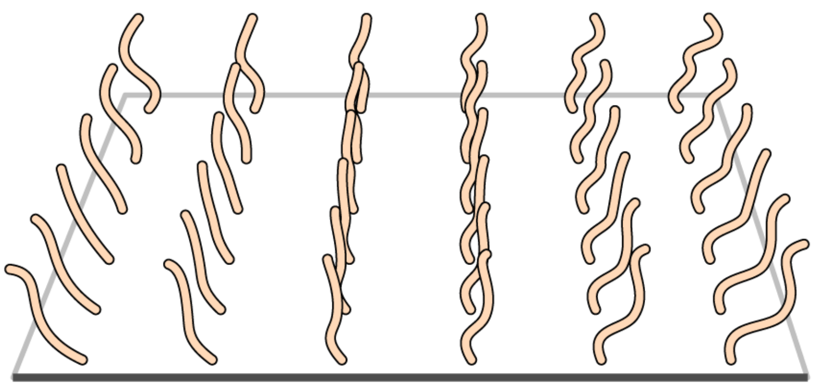


19

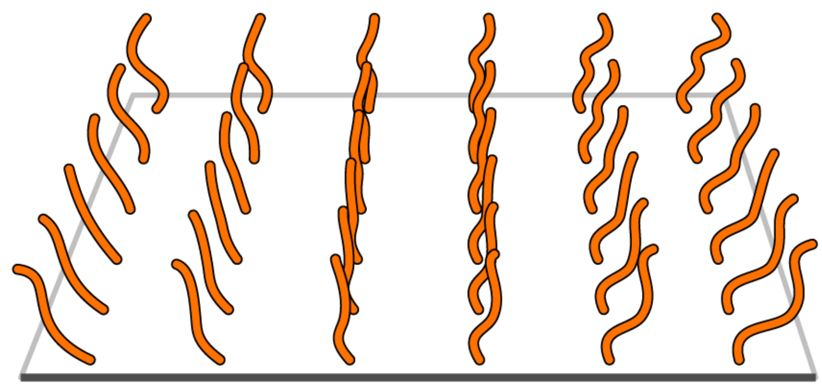


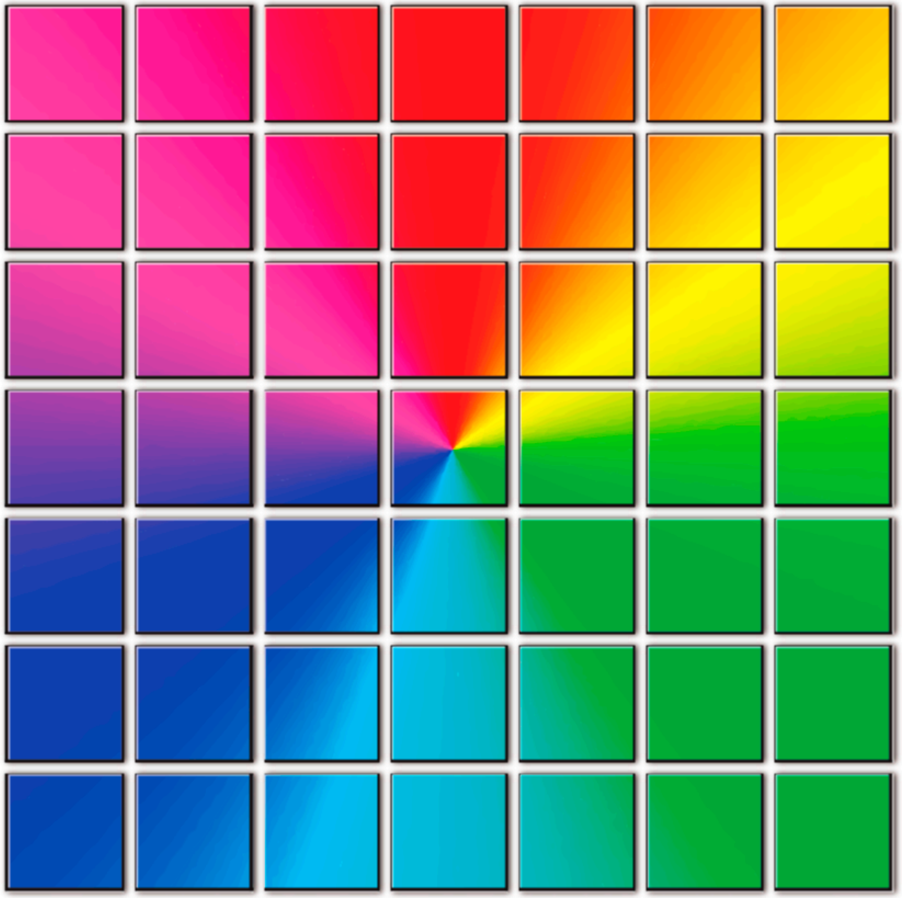
• • •

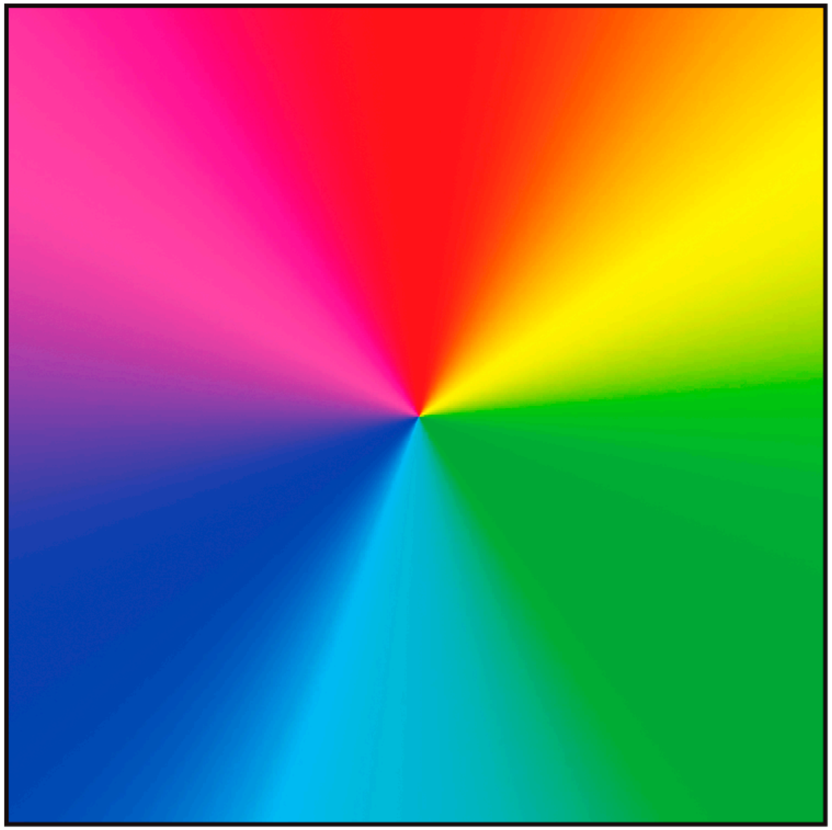
24



24





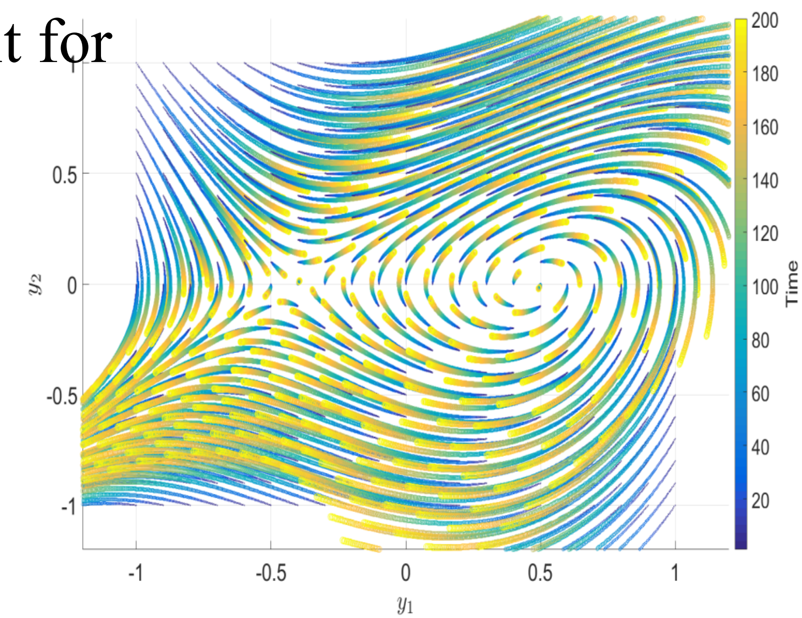


Bogdanov-Takens

- The dynamical system:

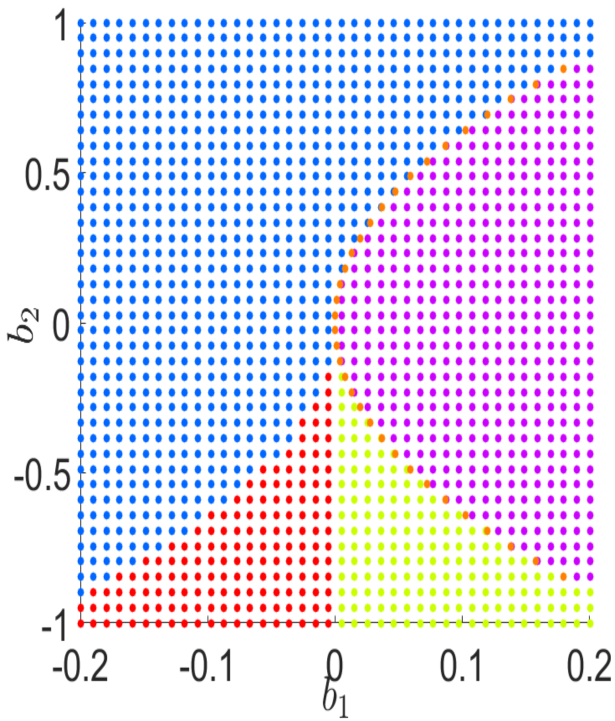
$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = \beta_1 + \beta_2 y_1 + y_1^2 - y_1 y_2 \end{cases}$$

- An example of a phase-portrait for fixed parameter values

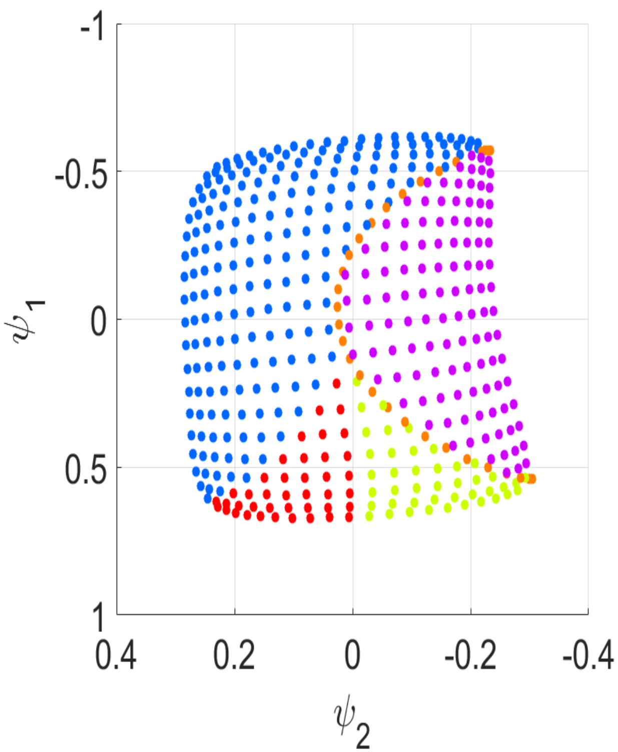


Bogdanov-Takens

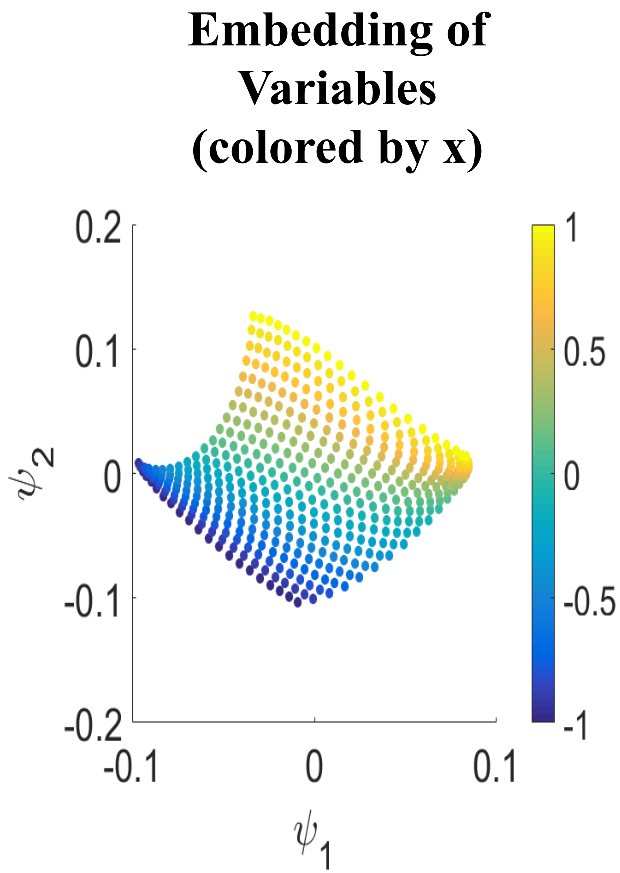
Bifurcation Map

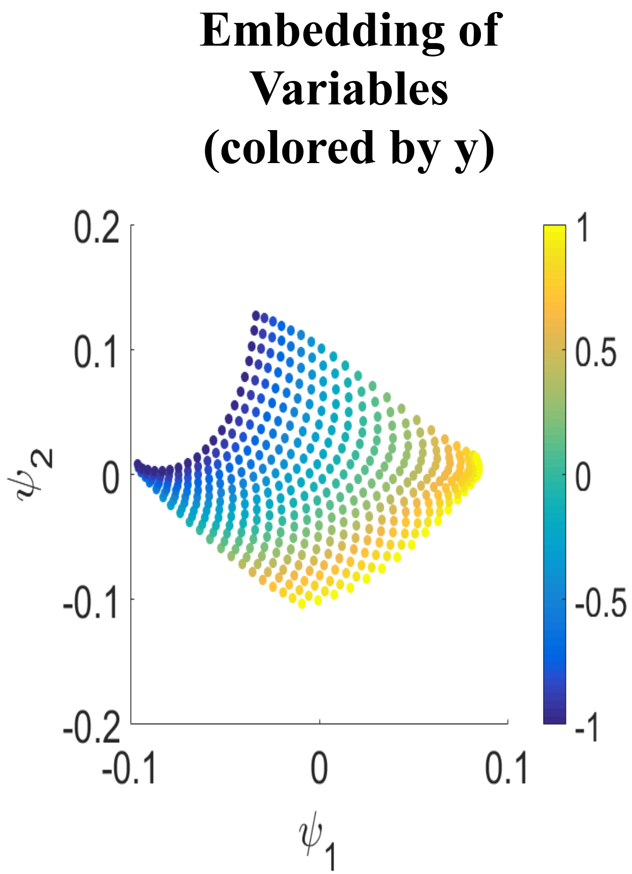


Embedding of the Parameters



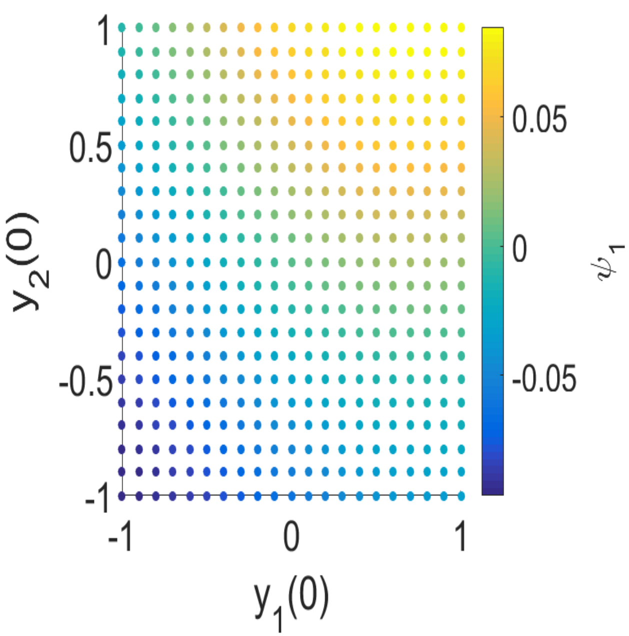
Bogdanov-Takens



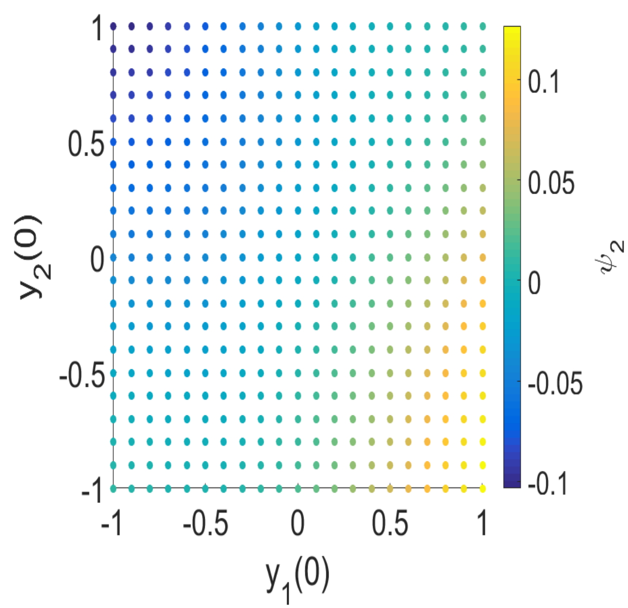


Bogdanov-Takens

Variables colored by the first embedding coordinate



Variables colored by the second embedding coordinate



And the most amazing thing is

We have an algorithm that establishes relations between

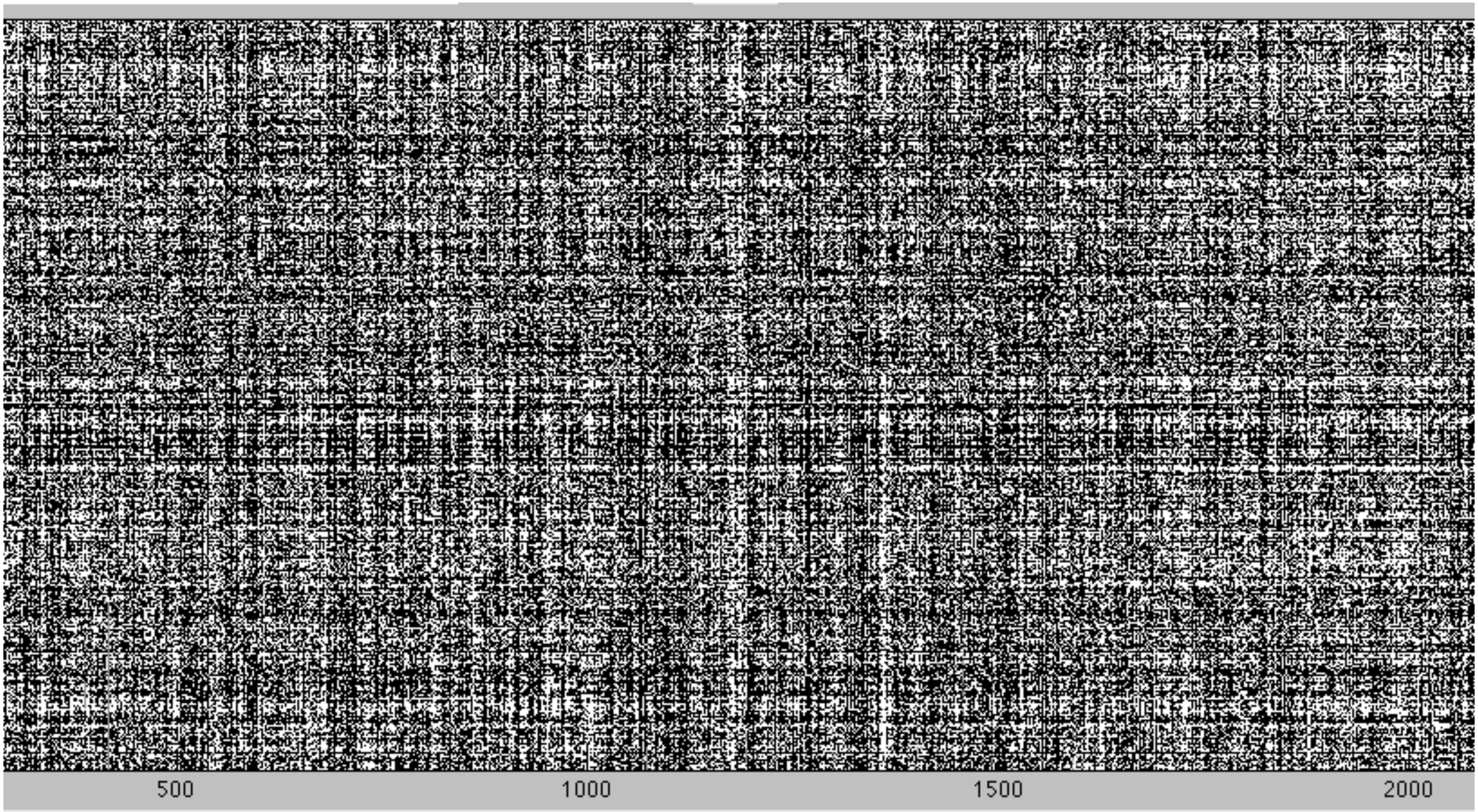
"discovered" parameters p
and "discovered" state variables, short trajectories x(t)
and also "discovered/estimated" dy1/dt and dy2/dt

NOW

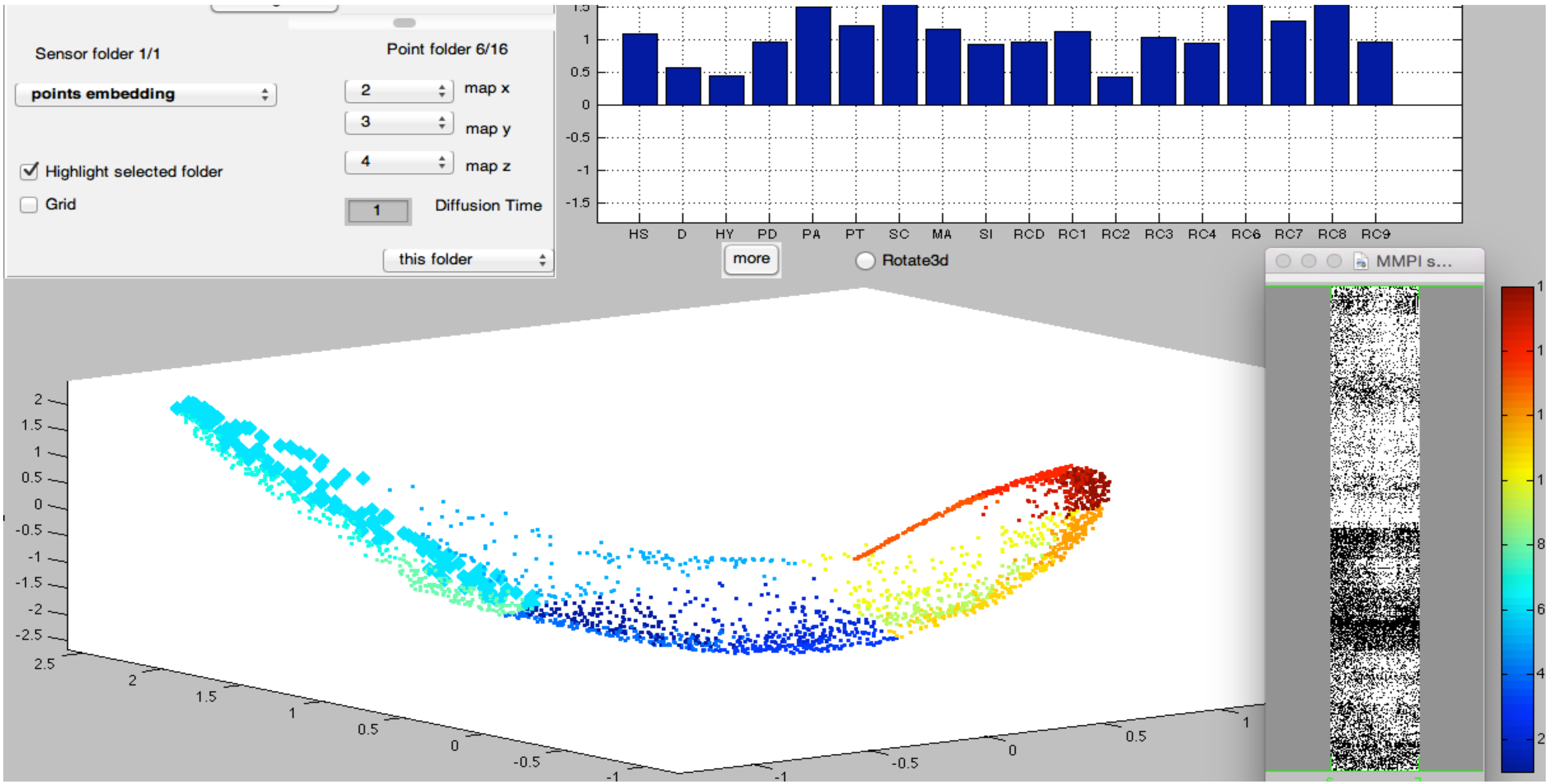
Use as inputs p, y1(t) , y2(t)
And as OUTPUTS dy1(t)/dt and d2(t)/dt
from data

We use the SAME methodology
And we now find the GENERATOR of the relation
The EQUATION ! We get a REALIZATION !

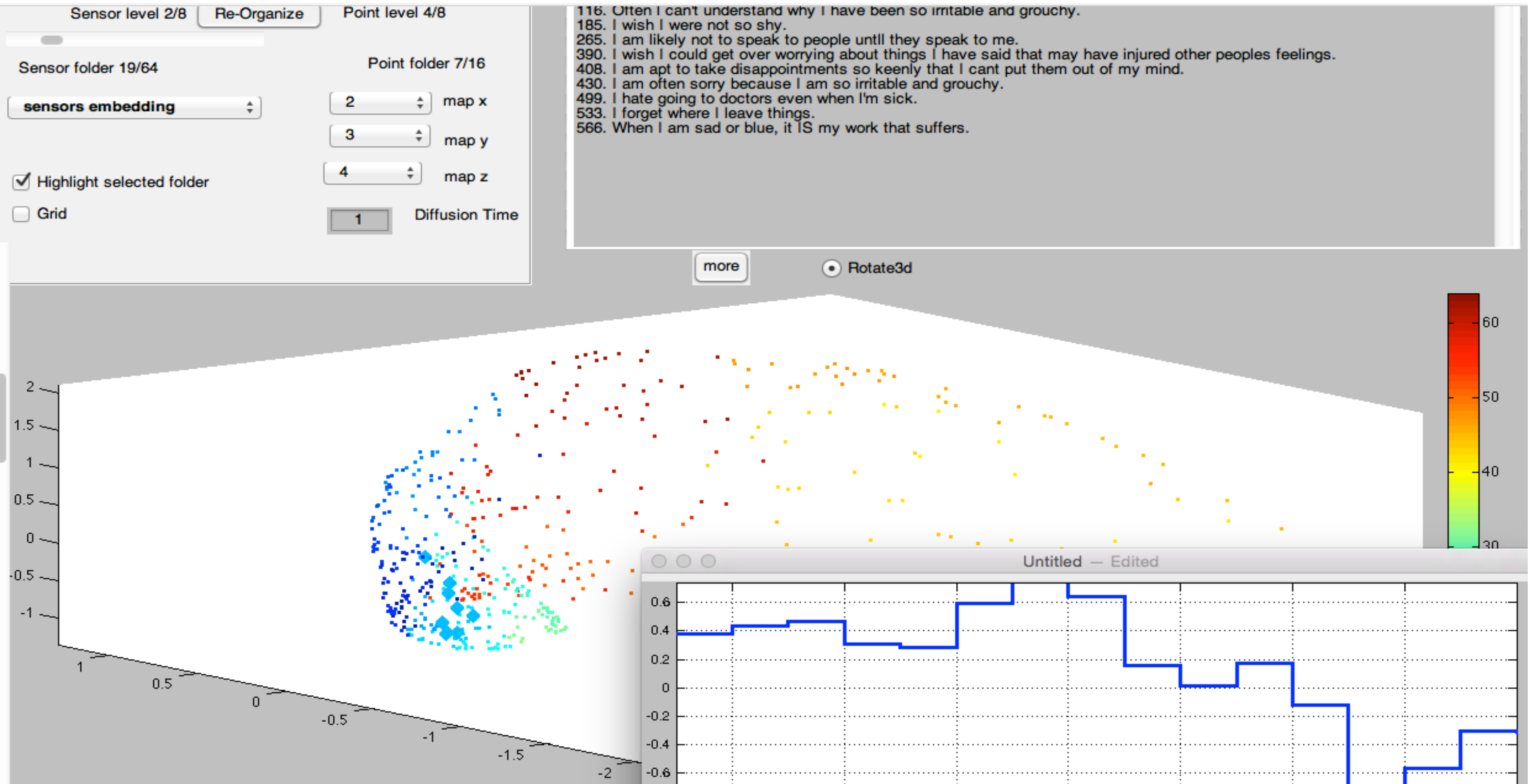
We now describe a generic method to organize a database . This example is a matrix of yes or no responses in a psychological questionnaire (MMPI), The data is undocumented ,(every column is a profile, every row is a question). Nevertheless we want to separate dysfunctional profiles .



The columns in the database have been organized into contextual groups in the geometric scatter plot the light blue group, left extreme tip ,representing the position of the gray list of similar profiles, and evaluated to be extremely dysfunctional (top right corner has Psychological scores)



conceptual group of questions , which are particularly informative , for the group selected before. All the questions (rows) are organized into "conceptual" codependent classes (shyness), their response in the different population groups are provided in the bottom right diagram.

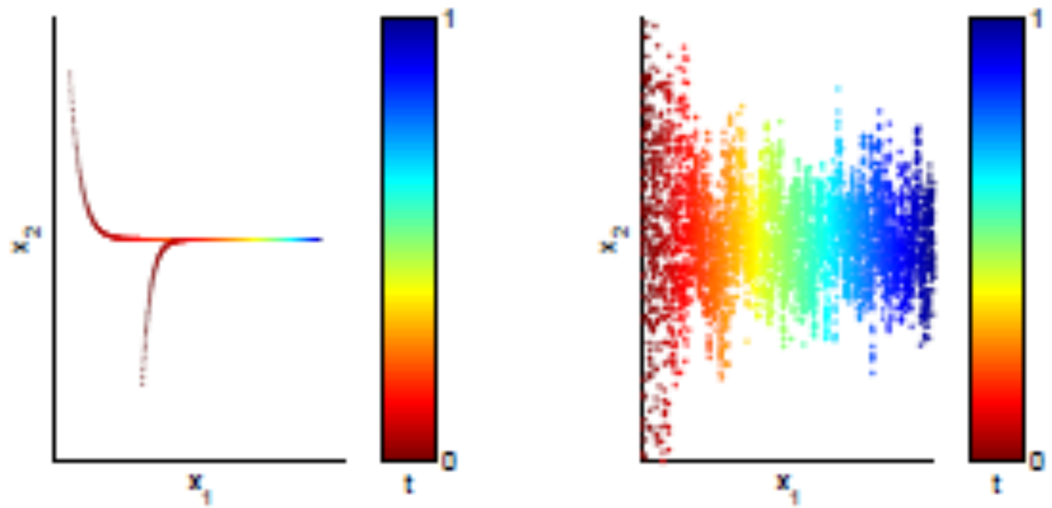


Another data problem, and a small miracle

- Two kinds of model/data reduction:
- one in which the value of the fast variable (its QSS) becomes slaved to the slow variable
- and one in which the statistics of the fast observable (its quasi-invariant measure) is slaved to the slow observable

Two types of reduction (“ODE” and “SDE”)

DATA-DRIVEN REDUCTION OF SDES 3



(a) (b)

Manifold Learning Applied to Reducible SDEs

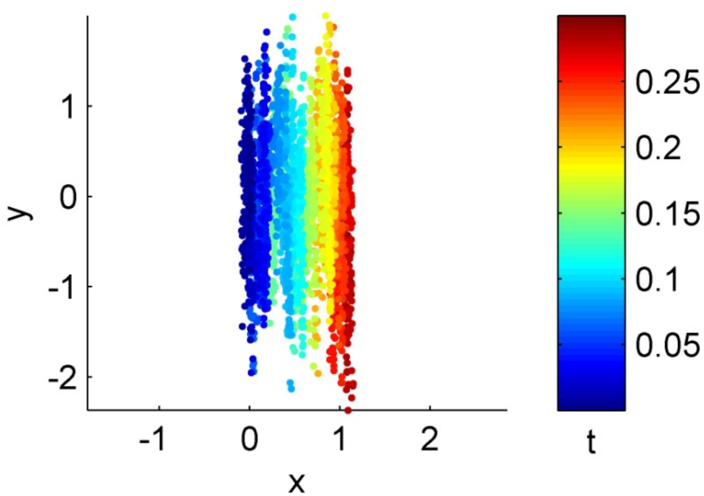
Consider the SDE

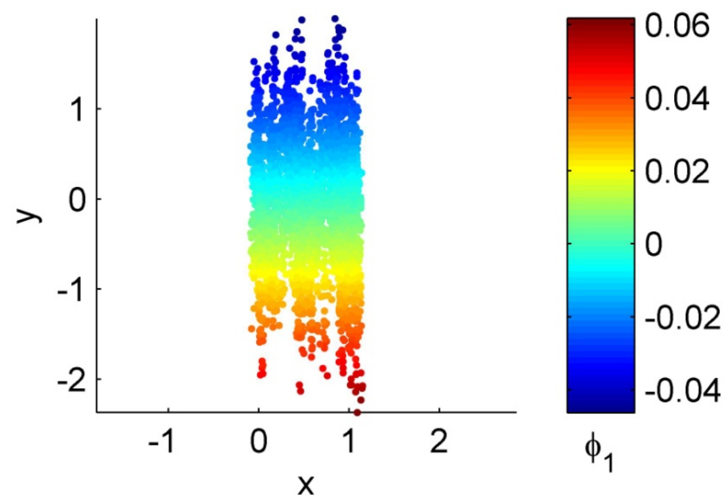
$$dx = a\,dt + dW_1$$

$$dy = -\frac{y}{\varepsilon}dt + \frac{1}{\sqrt{\varepsilon}}dW_2$$

Issue: *y* is still *O(1)*, so we cannot recover only *x* using DMAPS

If *ε* <





***a* = 3, *ε*=1e-3**

Rescaling to Recover the Slow Variables

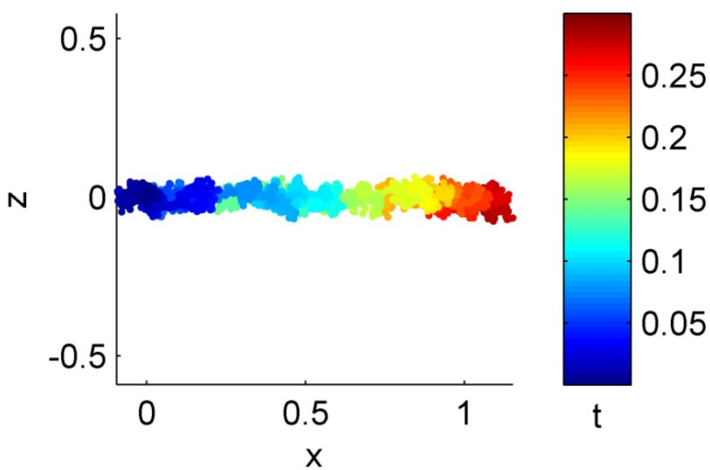
Consider the transformation

$$z = y\sqrt{\varepsilon}$$

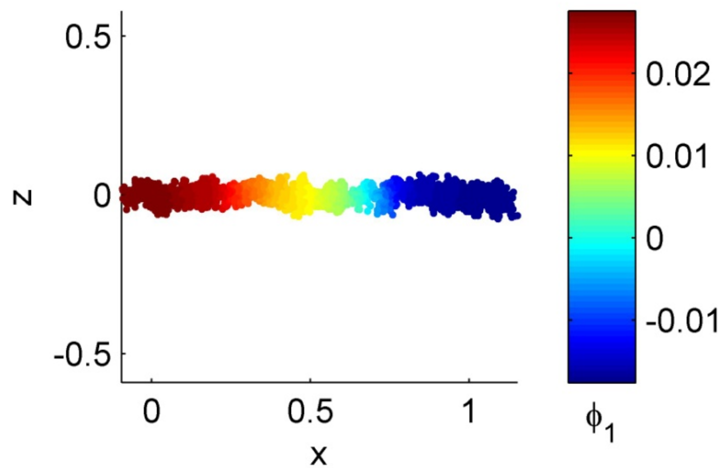
Then the SDEs become

$$dx = a\,dt + dW_1$$

$$dz = -\frac{z}{\varepsilon}dt + dW_2$$

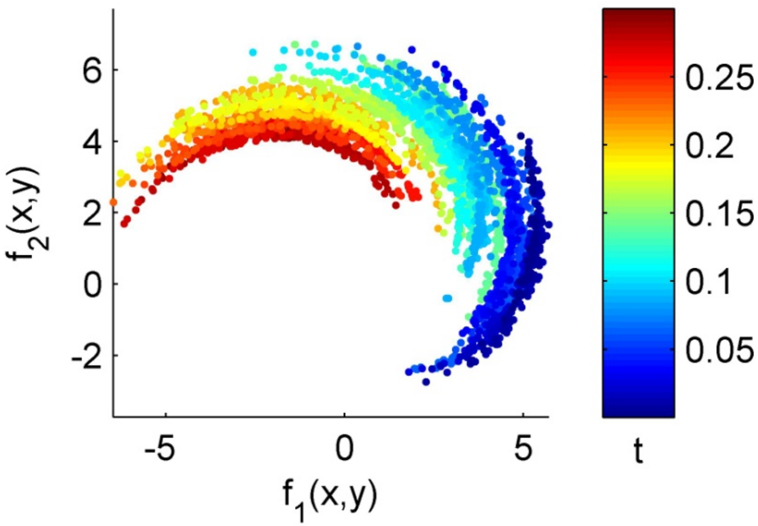


Now z is of a much smaller scale than x, and DMAPS will recover x



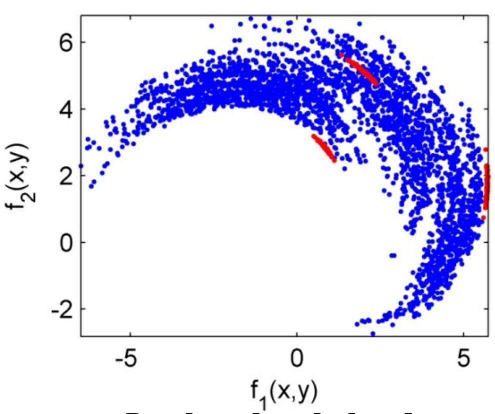
NIVs to Recover the Slow Variables with Nonlinearities

We can also recover NIVs when the data is obscured by a *nonlinear* measurement fun

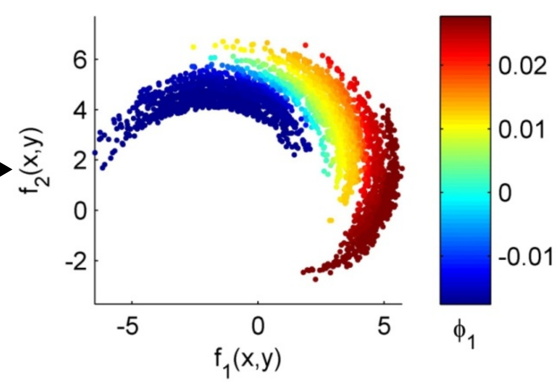


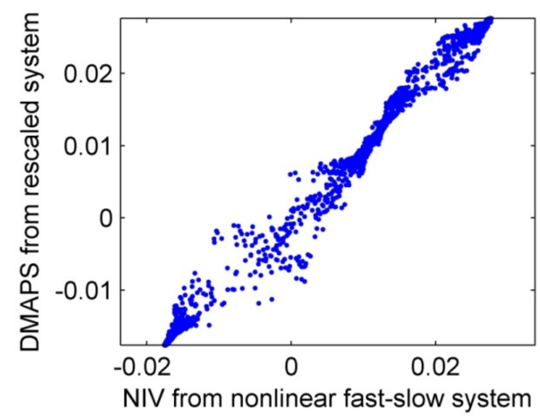
$$f_1(x,y) = (y+5)\cos\left(2x + \frac{y}{2}\right)$$

$$f_2(x,y) = (y+5)\sin\left(2x + \frac{y}{2}\right)$$



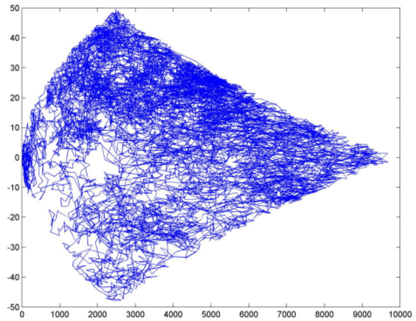
Look at local clouds in the data



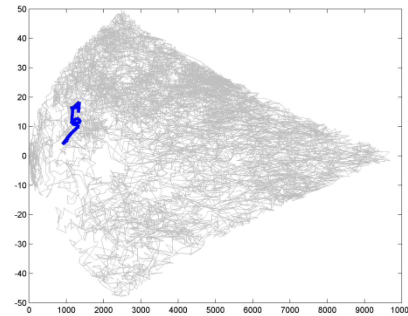


We again recover the slow variable *x*

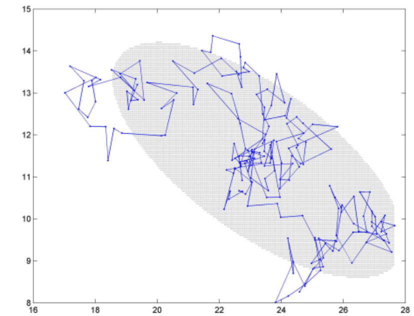
Computing the Distance Metric from Empirical Data



We have data from a trajectory or time series



For each data point, we look at the local trajectory in order to sample the local noise



We estimate the local covariance using the local trajectory

We then consider the distance metric

$$d^2(x_i, x_j) = 2(x_i - x_j)^T (C(x_i) + C(x_j))^\dagger (x_i - x_j)$$

where C is the estimated noise covariance.

This is the Mahalanobis distance and is invariant to the observation function (provided the observation function is invertible).

We then use the Mahalanobis distance in a DMAPS computation.

We call the resulting DMAPS variables Nonlinear Intrinsic Variables (NIV).

C. J. Dsilva *et al.*, *The Journal of chemical physics*, 2013.

- We assume the underlying **intrinsic variables** $\mathbf{x}(t) \in \mathbb{R}^d$ can be described by uncoupled stochastic differential equations

$$dx_i(t) = a_i(\mathbf{x}(t))dt + dw_i(t), \ \ i = 1, \dots, d,$$

where a_i are unknown drift functions and $\dot{w}_i(t)$ are independent white noises.
- We assume that we **observe** some function $\mathbf{f}(\mathbf{x}(t)) : \mathbb{R}^d \to \mathcal{M}$ of the intrinsic variables, where $\mathcal{M} \subset \mathbb{R}^n$ is a d-dimensional manifold.
- We assume that $\mathbf{f}$ is bi-Lipschitz and smooth.
- $\mathbf{f}$ can then be linearly approximated locally as $\mathbf{f}(\mathbf{x}(t)) = \mathbf{J}(t)\mathbf{x}(t) + \epsilon(t)$, where $\mathbf{J}(t)$ is the Jacobian of $\mathbf{f}$ and $\epsilon(t)$ contains higher-order terms.
- Let $\mathbf{C}(t)$ be the **local covariance** matrix. Then $\mathbf{C}(t) = \mathbf{J}(t)\mathbf{J}^T(t)$ and [8]

$$\begin{aligned}\|\mathbf{x}(t) - \mathbf{x}(\tau)\|^2 =& 2(\mathbf{f}(\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(\tau)))^T(\mathbf{C}(t) + \mathbf{C}(\tau))^\dagger(\mathbf{f}(\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(\tau))) \\ &+ \mathcal{O}(\|\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(\tau))\|^4)\end{aligned}$$

- Therefore, the distance

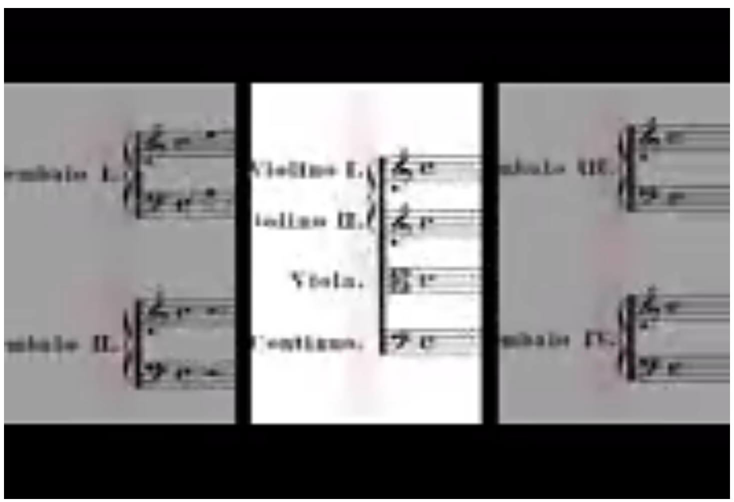
$$d^2(\mathbf{f}(\mathbf{x}(t)), \mathbf{f}(\mathbf{x}(\tau)) = 2(\mathbf{f}(\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(\tau))^T(\mathbf{C}(t) + \mathbf{C}(\tau))^\dagger(\mathbf{f}(\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(\tau))$$

approximates $\|\mathbf{x}(t) - \mathbf{x}(\tau)\|^2$ and is invariant to the measurement function $\mathbf{f}$.

[8] Singer, A. and R. R. Coifman, *Applied and Computational Harmonic Analysis*, 2008

The Music of Gauge Invariance

BWV 1065 Concerto for Four Harpsichords and Strings



Vivaldi – Concerto for Four Violins in B Minor RV 580 – Il Giardino Armonico

Data and the Science of Crystal Balls

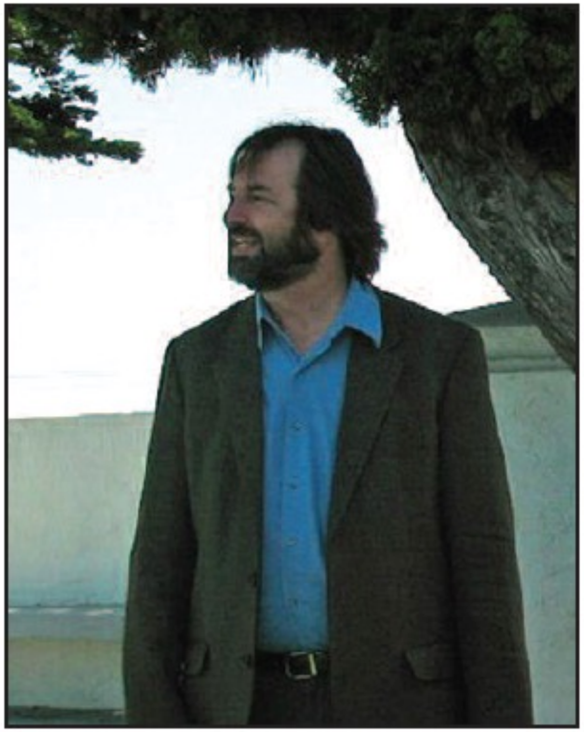
Thank you, over the years to

- AFOSR
- DOE
- NSF

But today, I want us to remember

a truly special man, scientist, visionary, program director

DENNIS HEALY, 1957-2009 (Dartmouth, Maryland, DARPA)



Dennis Healy, 1957–2009

But today, I want us to remember

a truly special man, scientist, visionary, program director

DENNIS HEALY, 1957-2009 (Dartmouth, Maryland, DARPA)

Many of Dennis's friends and colleagues have sent additional reminiscences, stories, and comments.

One of my strongest memories of Dennis comes from one quintessential New England winter's night. He and I had been in the office, working into the wee hours of the morning, finishing up some piece of work—either a proposal or a paper. It snowed heavily almost all night long, but by the time we had finished, the snowstorm had also come to an end. We turned out the lights and walked outside into a still and snow-covered campus. We decided to take a night-time stroll past the Green (now white!) and down Main Street to clear our heads before walking home. After the night of hard work, we were in high spirits, and clowned around, throwing snowballs, shaking snow off tree branches, and just generally goofing around in the local winter wonderland. At some point we stopped and stared up into the kind of beautiful, clear, and starry night that can be found only far from a city. I could hardly tell the difference between the North Star and the Dog Star, but Dennis gave me one of his excited and expert tours of the night sky. It was a night that I think stands for Dennis's career at Dartmouth—having fun, doing math, and delighting in the simple and sometimes hidden beauties of Nature and small-town New England."—*Dan Rockmore.*

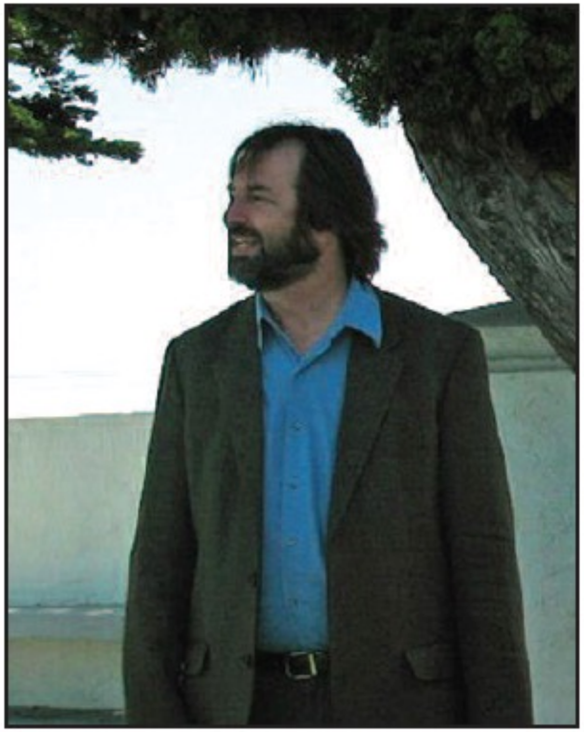
Thank you, over the years to

- AFOSR
- DOE
- NSF

But today, I want us to remember

a truly special man, scientist, visionary, program director

DENNIS HEALY, 1957-2009   (Dartmouth, Maryland, DARPA)



Dennis Healy, 1957–2009

But today, I want us to remember

a truly special man, scientist, visionary, program director

DENNIS HEALY, 1957-2009 (Dartmouth, Maryland, DARPA)

Many of Dennis's friends and colleagues have sent additional reminiscences, stories, and comments.

One of my strongest memories of Dennis comes from one quintessential New England winter's night. He and I had been in the office, working into the wee hours of the morning, finishing up some piece of work—either a proposal or a paper. It snowed heavily almost all night long, but by the time we had finished, the snowstorm had also come to an end. We turned out the lights and walked outside into a still and snow-covered campus. We decided to take a night-time stroll past the Green (now white!) and down Main Street to clear our heads before walking home. After the night of hard work, we were in high spirits, and clowned around, throwing snowballs, shaking snow off tree branches, and just generally goofing around in the local winter wonderland. At some point we stopped and stared up into the kind of beautiful, clear, and starry night that can be found only far from a city. I could hardly tell the difference between the North Star and the Dog Star, but Dennis gave me one of his excited and expert tours of the night sky. It was a night that I think stands for Dennis's career at Dartmouth—having fun, doing math, and delighting in the simple and sometimes hidden beauties of Nature and small-town New England."—*Dan Rockmore.*