Equation-free Computation For Complex Systems

or

Enabling Microscopic Time-Steppers to perform System Level Tasks

or

Solving Differential Equations Without the Equations

Or

Systems Engineering for Multiscale Simulations

I. G. Kevrekidis, C. W. Gear and many other good people

Department of Chemical Engineering, PACM & Mathematics
Princeton University, Princeton, NJ 08544

Multiscale / Complex System Modeling

"Textbook" engineering modeling:

macroscopic behavior *through* macroscopic models

(e.g. conservation equations augmented by closures)

Alternative (and increasingly frequent) modeling situation:

- Models
  - at a **FINE** / ATOMISTIC / STOCHASTIC level
  - MD, KMC, BD, LB (also CPMD…)
- Desired Behavior
  - At a **COARSER**, Macroscopic Level
  - E.g. Conservation equations, flow, reaction-diffusion, elasticity
- Seek a bridge
  - Between Microscopic/Stochastic Simulation
  - And "Traditional, Continuum" Numerical Analysis
  - When closed macroscopic equations are not available ***in closed form***

What I will tell you:

Solve the equations WITHOUT writing them down.

Write "software wrappers" around "fine level" microscopic codes

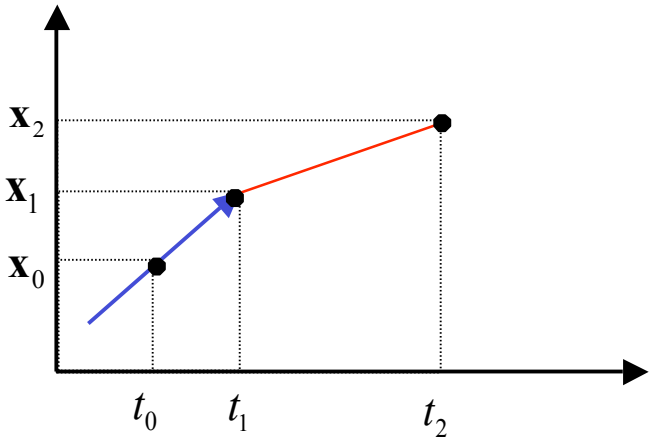
Top level: **all algorithms we know and love**
Bottom level: MD, kMC, LB, BD, heterogeneous/ discrete media, CPMD, hybrid
INTERFACE:

Trade Function Evaluation for "on demand" experimentaton and estimation

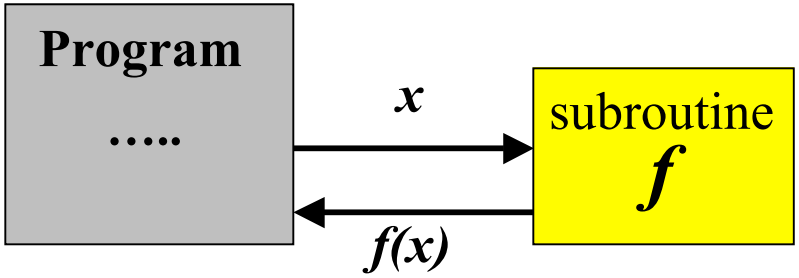
Think of the microscopic simulator AS AN EXPERIMENT That you can set up and run at will

"Equation Free" (motivated by "matrix free iterative linear algebra")
Algorithms (coarse integration, patch dynamics, coarse RPM…)
Tasks (stability/ bifurcation, control, optimization, dynamic renormalization)
Examples (LB, KMC, BD, MD), ***and some nebulous thoughts***

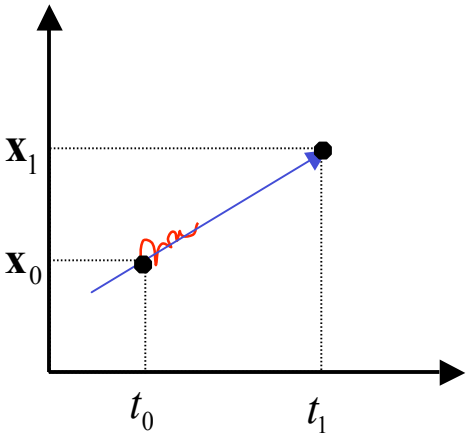
Equations $\frac{d\,\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$



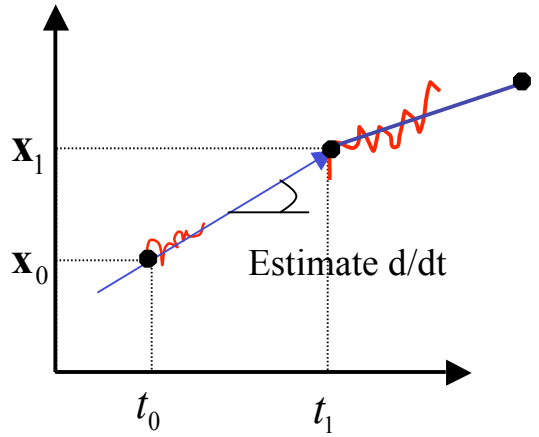
Determinism (in a practical way)



Experimental Experimentalist



Legacy Code
Computational Experimentalist

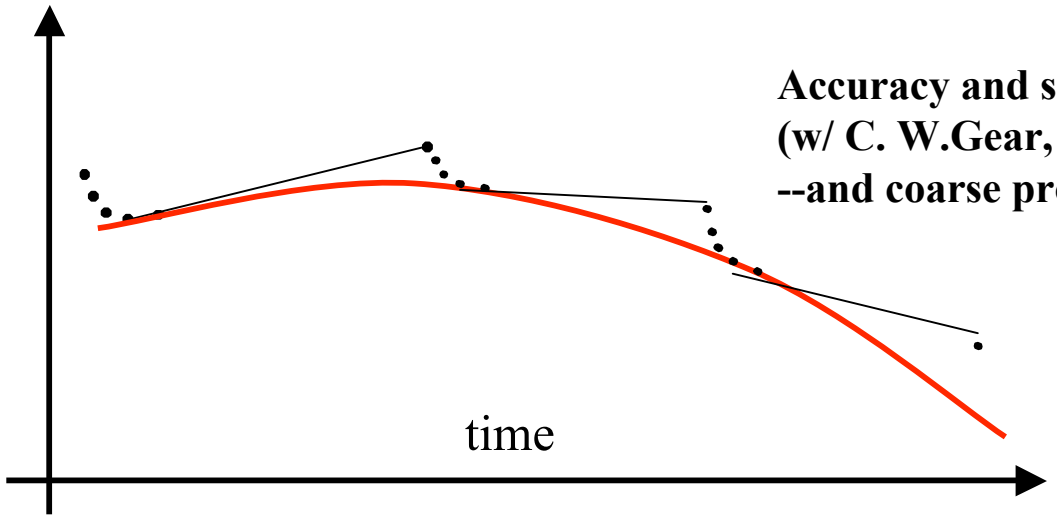


IN EFFECT, Do Forward Euler
not with the EQUATION, but with the (computational) EXPERIMENT

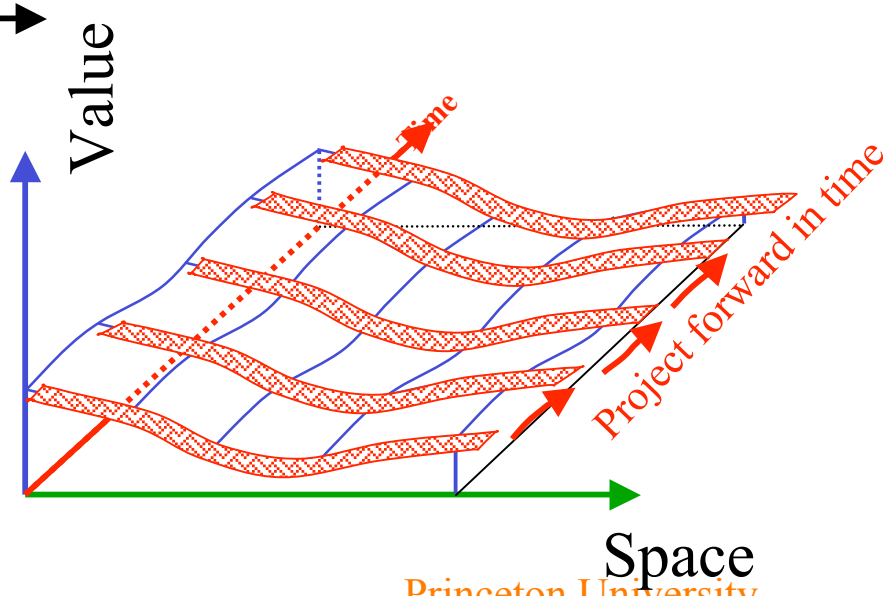
Look at the experiment & RESTART IT

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:

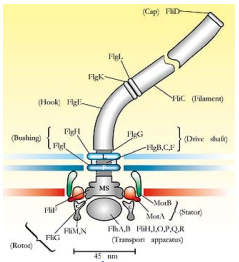
-perform detailed simulation for short periods
or use existing/legacy codes
- and then *extrapolate* forward over large steps

Chemotaxis: A biased random walk problem

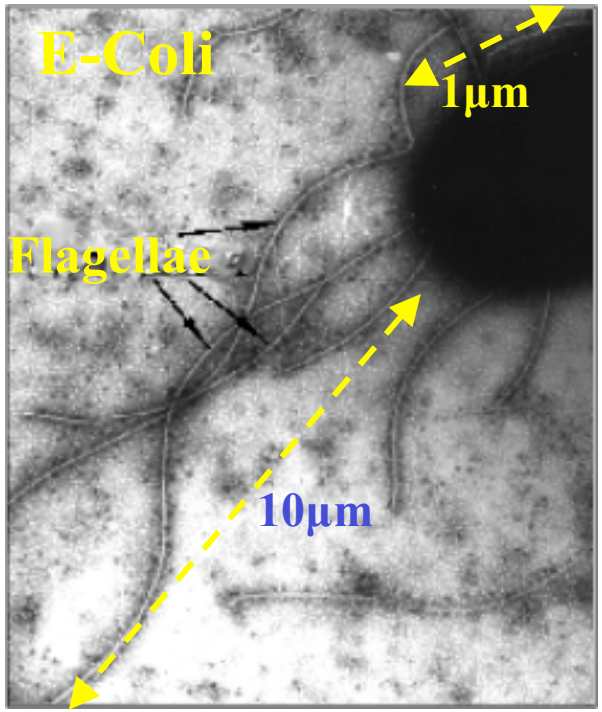
E-Coli bacteria move in space by rotating their flagellae

Chemotactic signal (attractant/repellant)

Signal transduction

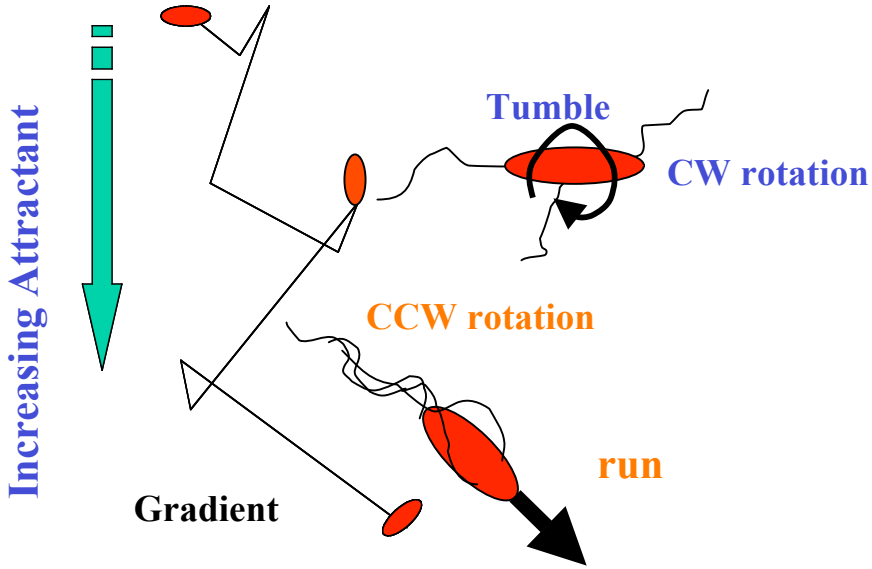


Response (Che-Yp protein)



Berg HC Motile behavior of bacteria

PHYS TODAY 53 (1): 24-29 JAN 2000



Physical Constants (no stimulus):

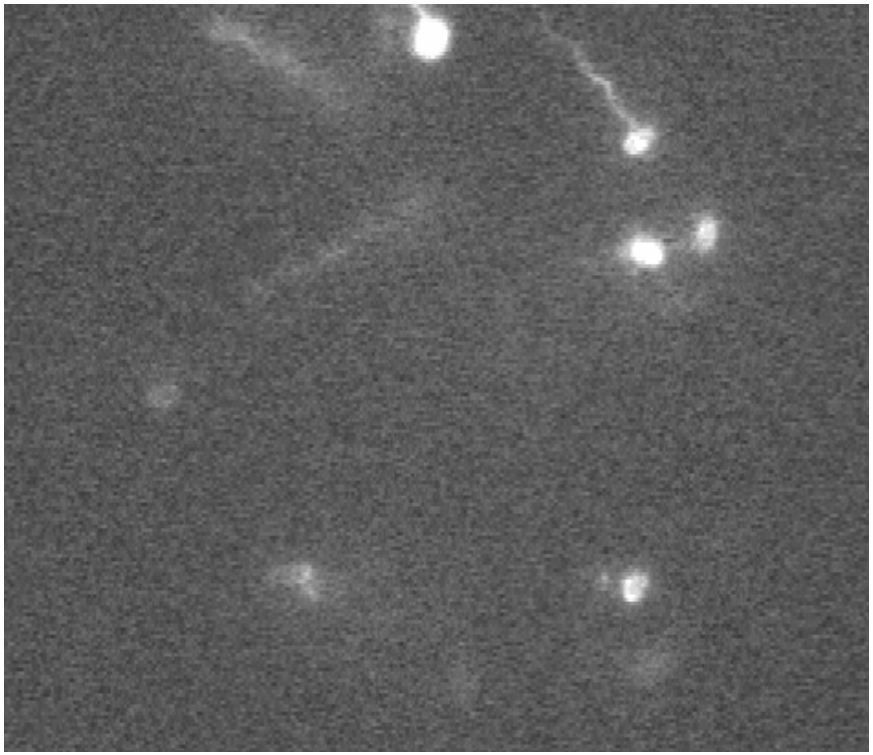
velocity: u~20-60 µm s^{-1}

Mean run time ~1s

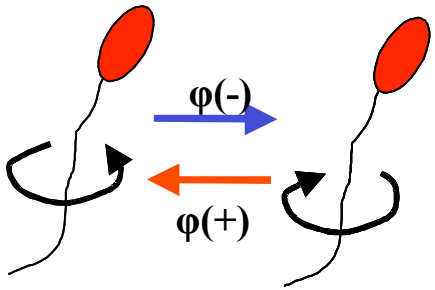
Mean tumble time~0.1s

Chemotaxis: A biased random walk problem

The stochastic model: Markov-Monte Carlo



(Movie: http://www.rowland.org/bacteria/movies.html)



Draw URN ζ [0, 1]

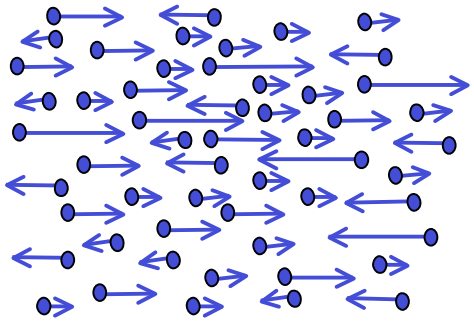
Compare ζ with probabilty of switching direction of rotation:
p = 1-φ(+/-)

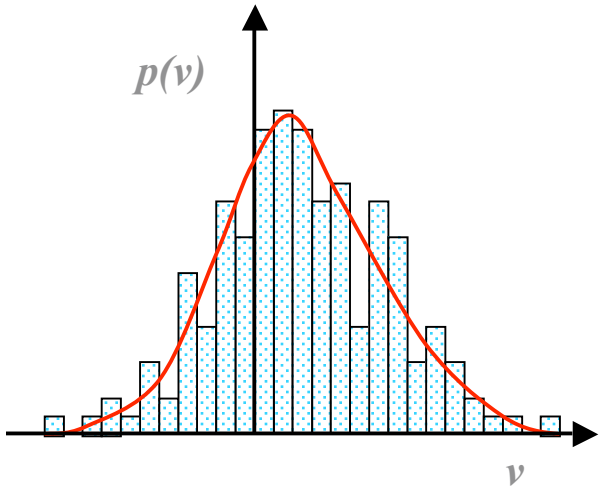
If CW and ζ>p, keep rotating CW;
Else, switch to CCW

If CCW and ζ>p, keep rotating CCW;
Else, switch to CW

If <3 CW, then TUMBLE. Else, RUN.
If previously running, direction unchanged.
Else, direction = +/- 1, with equal probability

DISTRIBUTIONS & MOMENTS





Descriptions

1. **Detailed: v_i for each variables**
2. **Moments**

m_0 **Zeroth moment**: density ρ $\int p(v)dv$

m_1 **First moment**: momentum ρv $\int p(v)v\,dv$

m_2 Second moment

m_3 Third Moment

.....

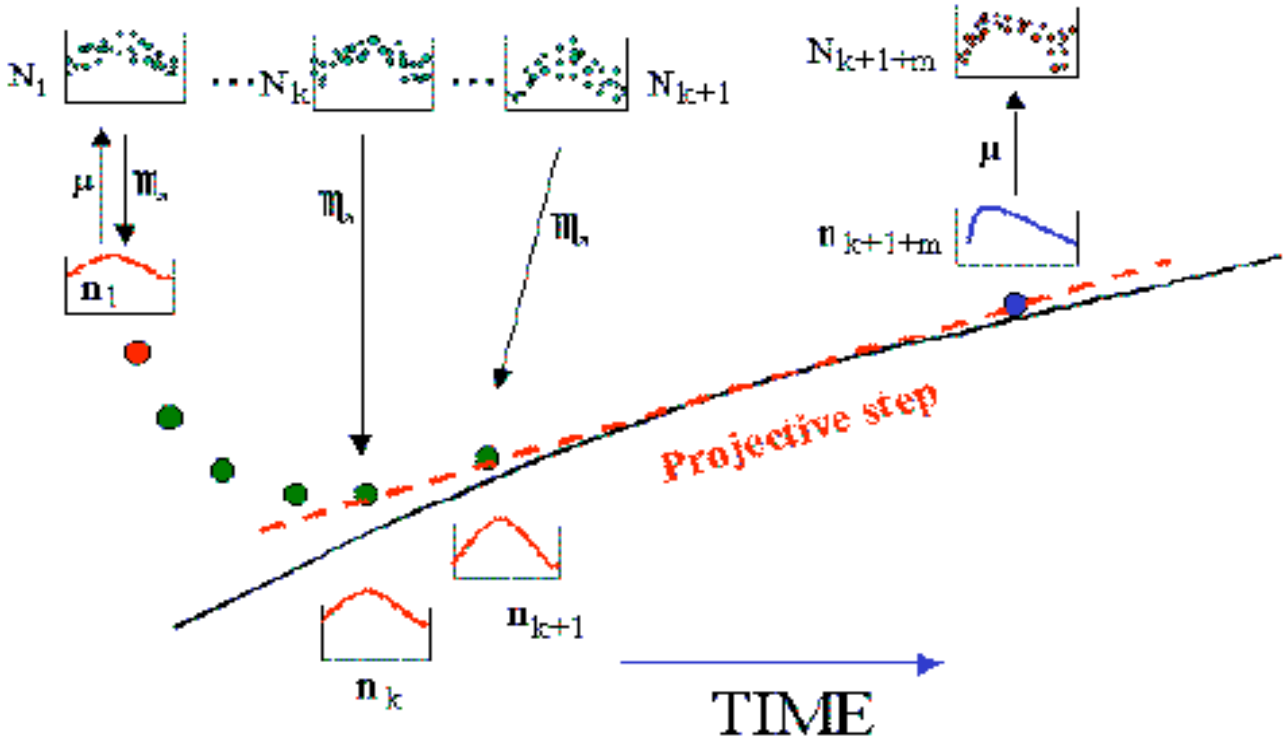
$$\frac{\partial p}{\partial t} = D\frac{\partial^2 p}{\partial x^2} + X\frac{\partial}{\partial x}\left(p\frac{\partial U}{\partial x}\right)$$

p: density

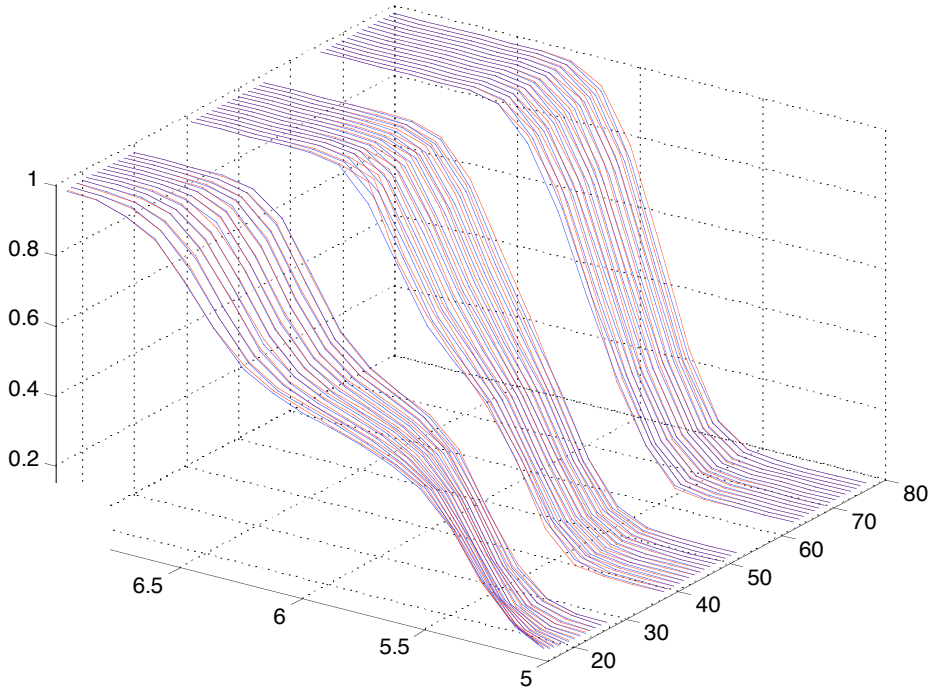
U: Potential

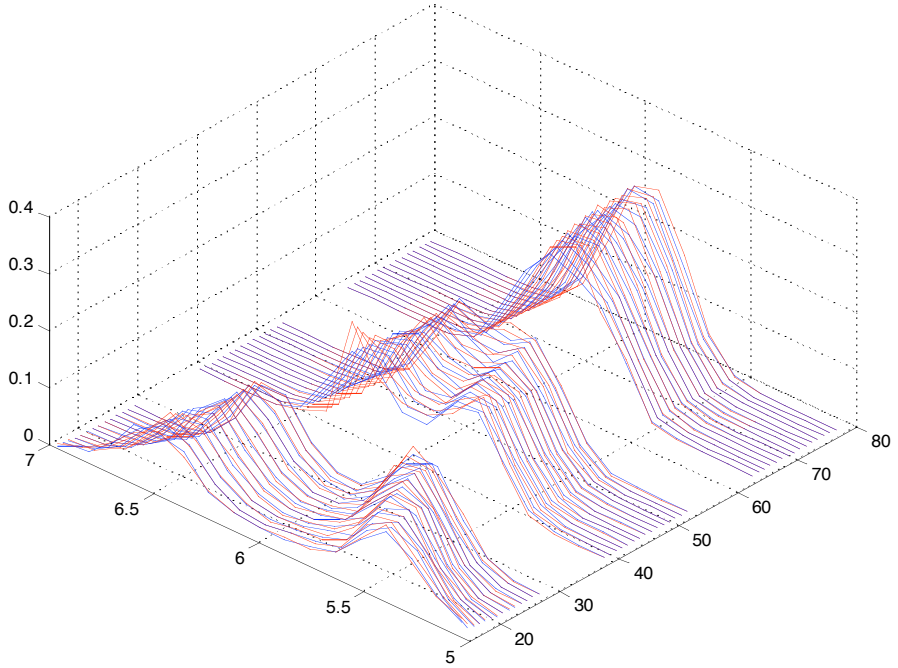
X: chemotactic coefficient

(Keller and Segel, 1971)



Chemotaxis: Coarse Projective Integration



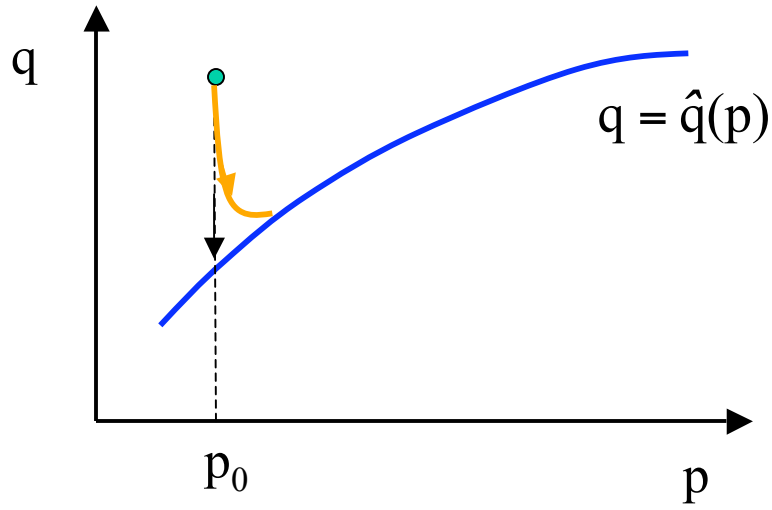


RESTRICTION - a *many-one* mapping from a high-dimensional description (such as a collection of particles in Monte Carlo simulations) to a low-dimensional description - such as a finite element approximation to a *distribution* of the particles.

LIFTING - a *one-many* mapping from low- to high-dimensional descriptions.

We do the step-by-step simulation in the high-dimensional description.

We do the macroscopic tasks in the low-dimensional description.



$$\frac{dp}{dt} = f(p,q)$$

$$\frac{dq}{dt} = \frac{1}{\varepsilon} g(p,q)$$

ε small → $g(p,q) \approx 0$, $q \approx \hat{q}(p)$ } very fast

Quickly: $\dot{p} = f(p,q) \approx f(p,\hat{q}(p)) \equiv \tilde{f}(p)$

Inertial Manifolds : p, q Low/ high MODES --- EOF/POD

Here : p, q Low/ high MOMENTS --- Order Parameters
Phase Fields

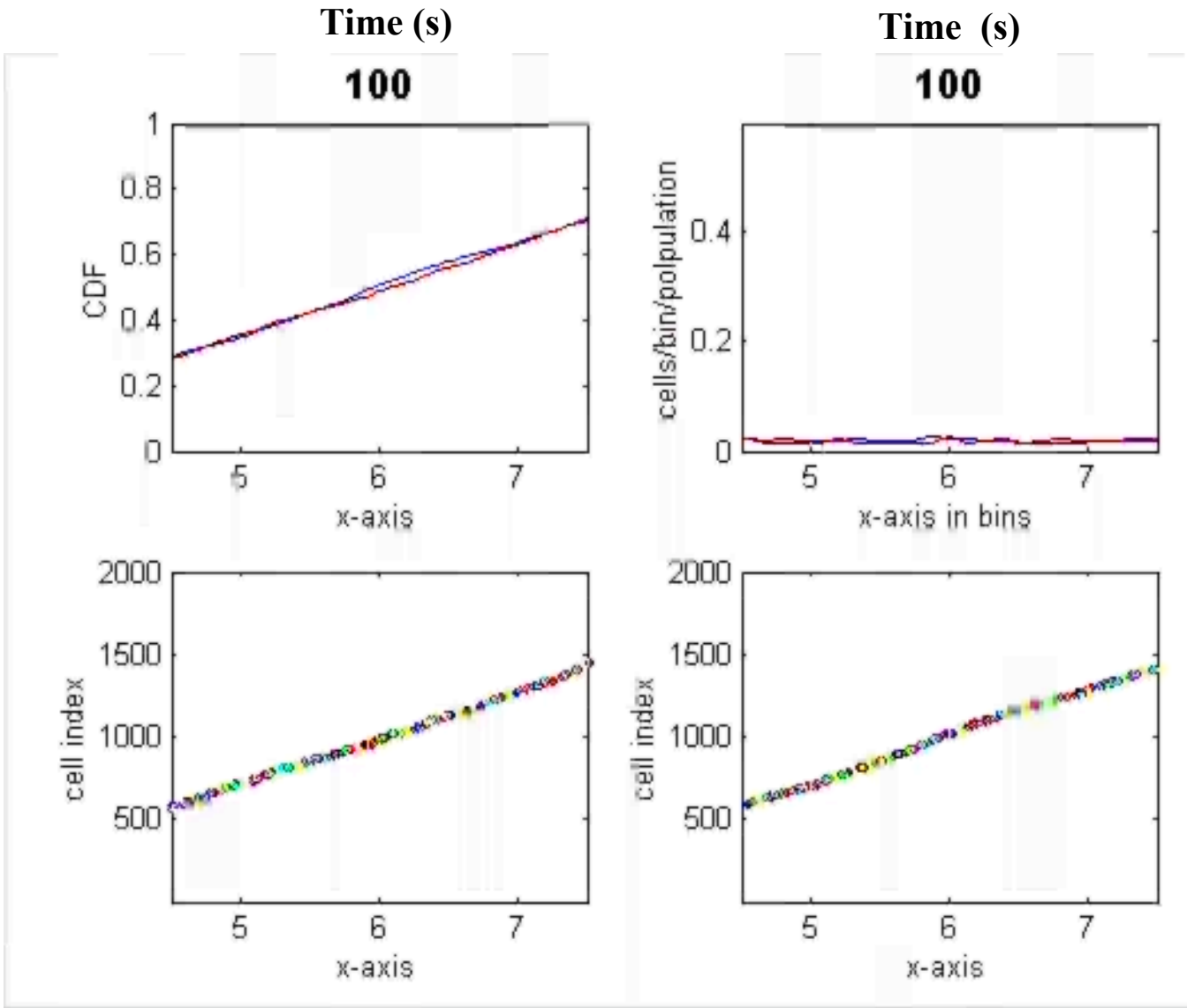
Attacting, slow, invariant manifold ——

AIM/ AIF --- Closure ---- "Free energy surface"

Initialize p_0 ——— Natural dynamics
——— Constrained dynamics (SHAKE)

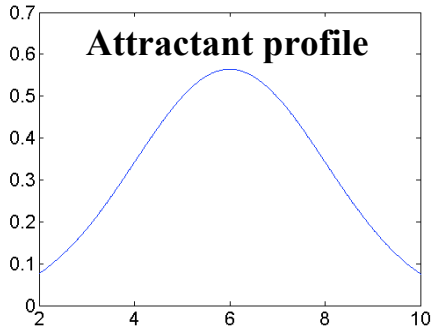
Chemotaxis: Coarse Projective Integration

(5 healing, m=10 acquisition and k=10 projection till time=6000 and then 5 healing, 10 acquisition and k=20 projection till time=20000)

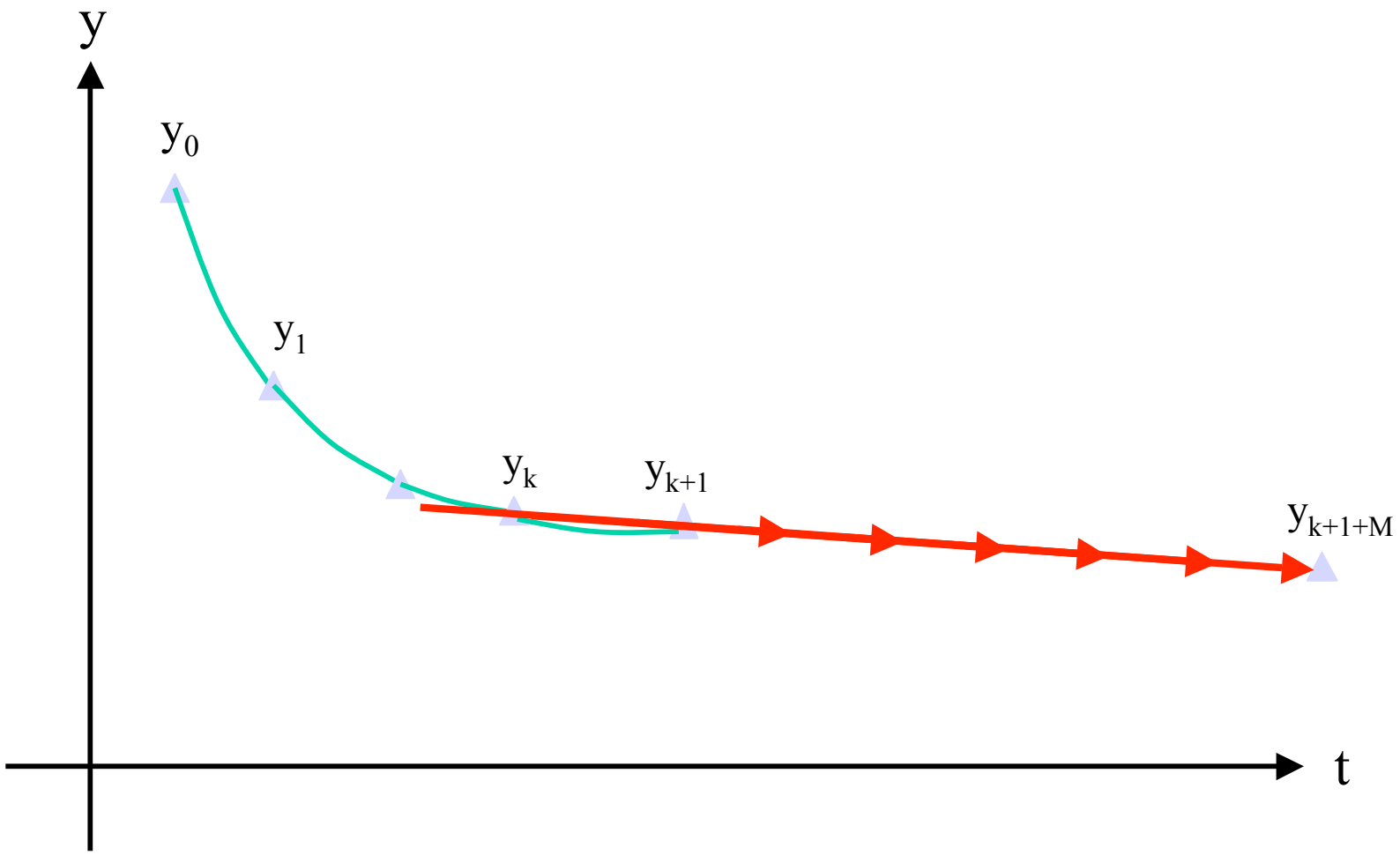


Normal Evolution

Coarse Projection



Direction



… but everybody knows that simple explicit methods can't be stable for stiff problems ...

The underlying(extremely simple) idea is to use the chord connecting successive output points from the integrator to approximate the derivative for use by another integrator (or other analytical tools).

We will discuss:

1. What sort of stability can be expected in the integrator

2. How it can be applied to a *restriction* of a microscopic description

3. How it might be applied to stochastic systems

4. Some interesting extensions of the integrator

Projective Forward Euler (PFE)

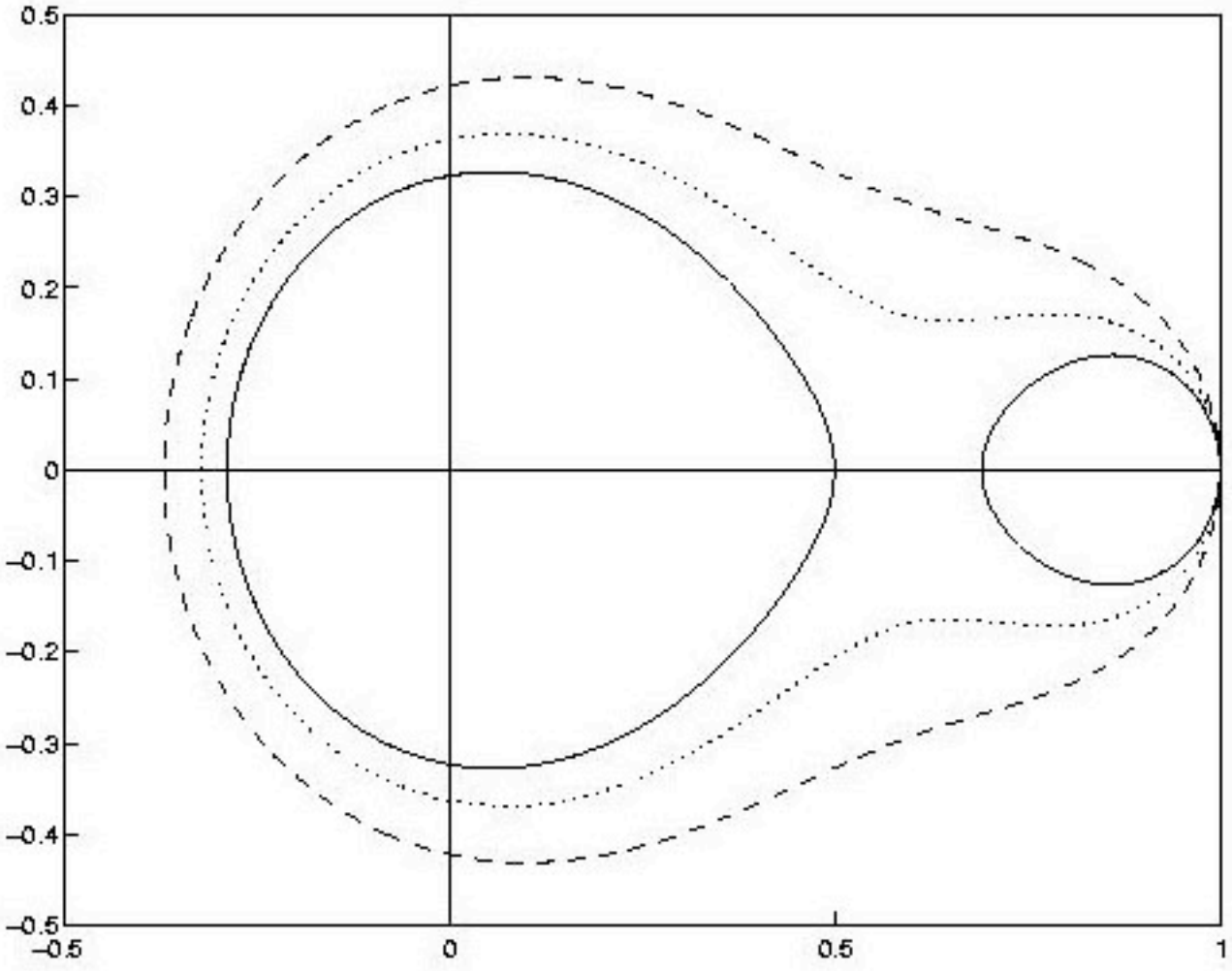
Usual linear analysis: for $y' = \lambda y$

Assume that one step of the supplied integrator (the *inner integrator*) has an amplification of $\rho(h\lambda)$
-for Forward Euler this is $1+h\lambda$
- for an "exact" integrator this is $\exp(h\lambda)$

Amplification from t_0 to t_{k+1+M} is

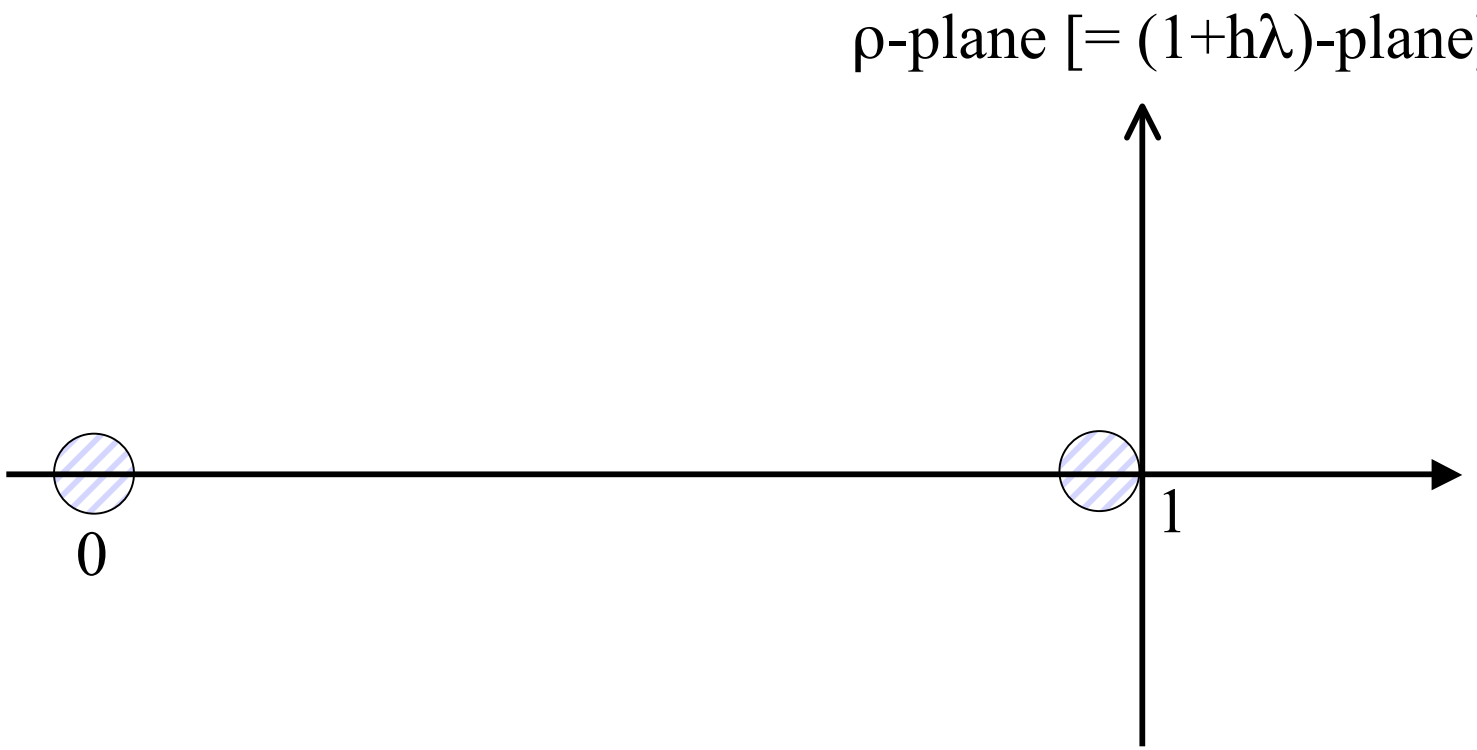
$$\sigma = \rho^k[(M+1)\rho - M]$$

Region of absolute stability: Set of ρ such that $|\sigma| \leq 1$.
Computed by finding boundary: ρ such that $\sigma = e^{i\theta}$
The region of absolute stability of takes one of two forms:



k = 2, M = 5, 7, and 9

If M is large:



Such a small stability region - what's the point?

"Coarse" Integration of Microscopically-defined systems

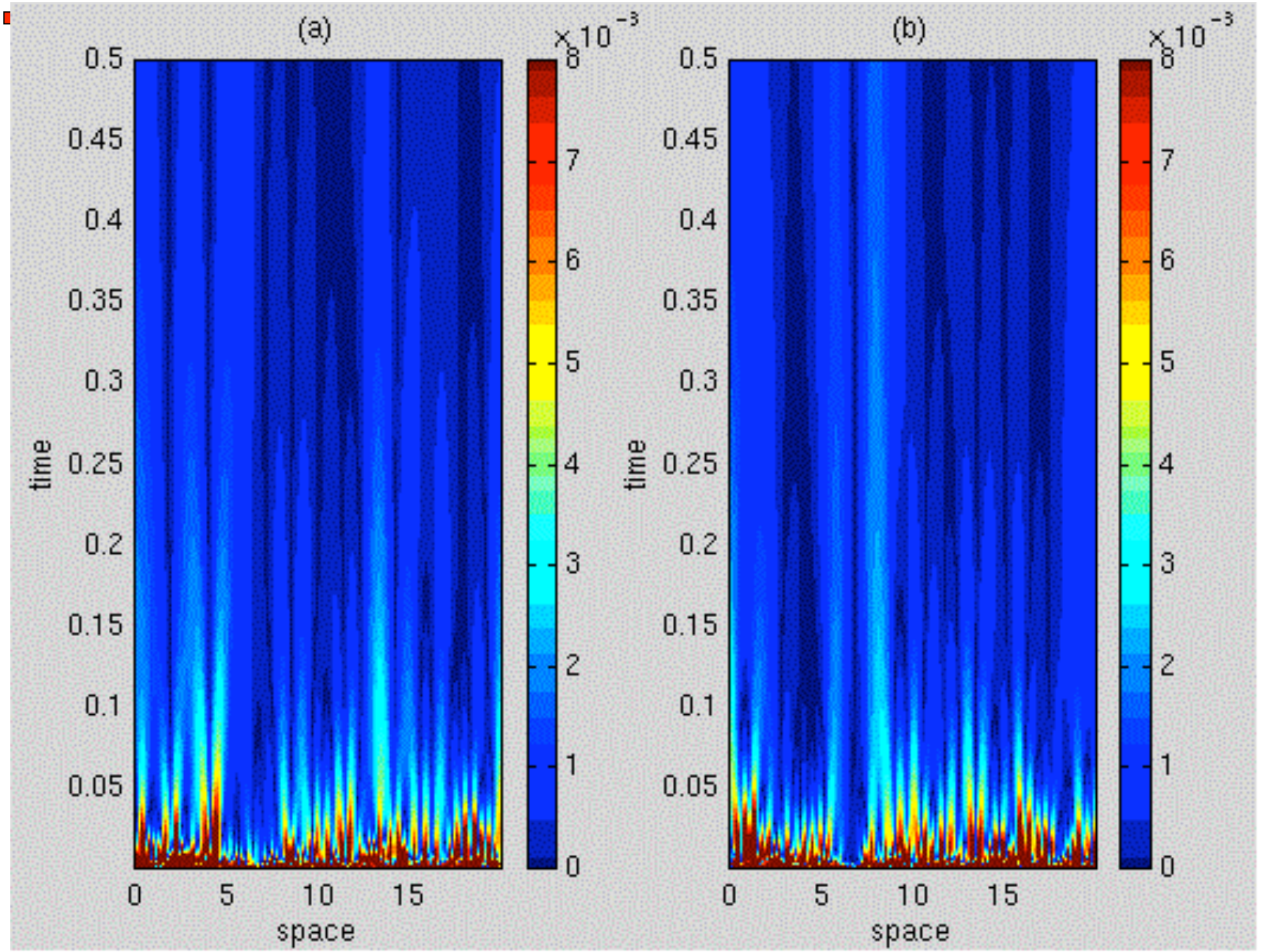
Microscopic systems usually do not have decaying fast components. In many cases we can find a slower system built on new variables that are averages (moments) of the microscopic system. In classical cases (e.g Navier Stokes) the differential equation has been found.

We suppose that we have a microscopically-defined system, and we believe that there is a PDE that describes the behavior of a lower-dimensional description of the system - the ***restriction*** of the system. We hope to be able to integrate it without explicit knowledge of the PDE using the techniques above.

We have at each time step t_j: a microscopic description $\mathbf{N}_j$
a macroscopic description $\mathbf{n}_j$

and mappings: ***Restriction*** $M\mathbf{N}_i = \mathbf{n}_i$ (this is like a projection)
Lifting $\mu\mathbf{n}_i = \mathbf{N}_i$

We expect $M\mu = \mathrm{I}$ (the identity)


(a)
(b)
time
space
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
5
10
15
×10^-3
8
7
6
5
4
3
2
1
0

Telescoping Projective Methods

We have created an "outer integrator" over a step size of h(k+q+M) using an inner integrator Φ over step size h.

Why not use recursion? (really iteration)

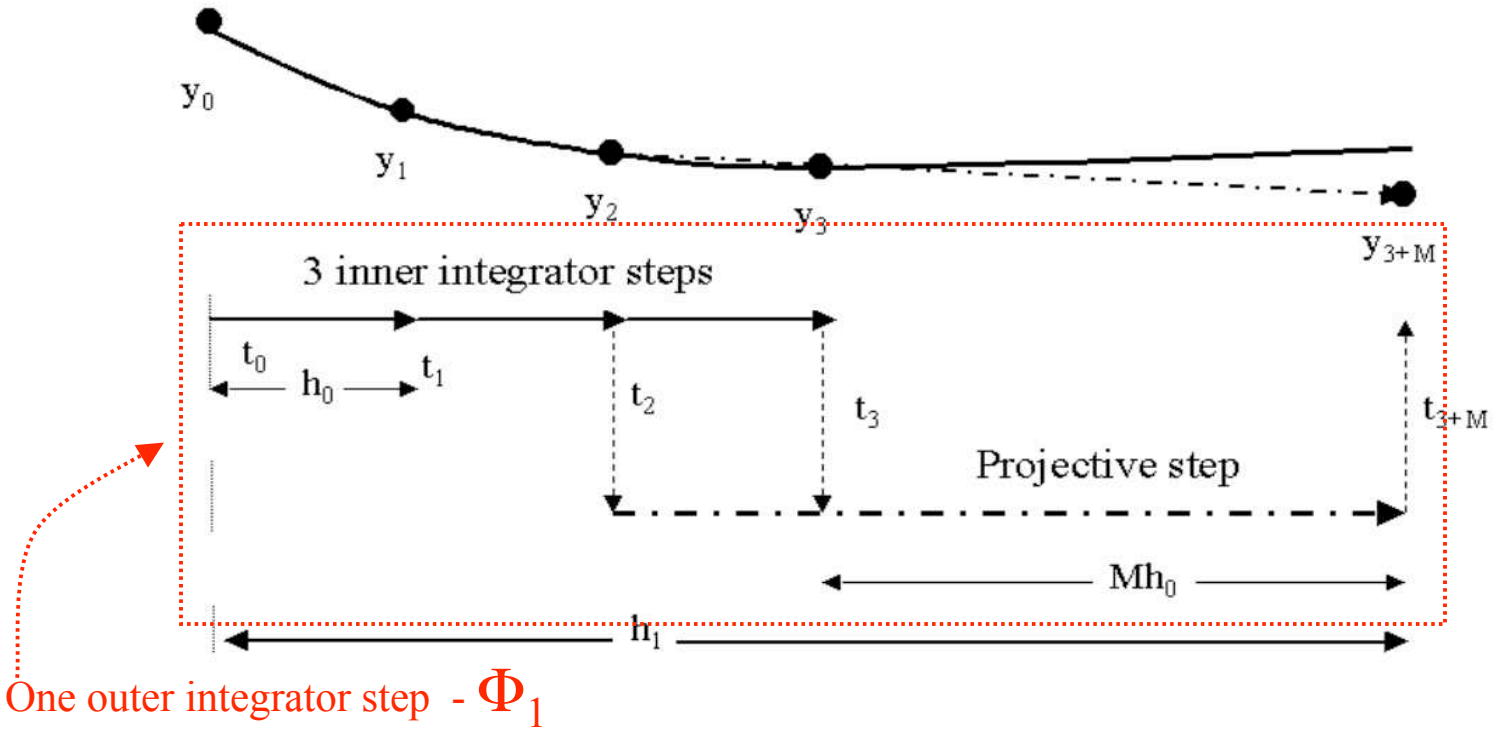
Inner integrator: Φ_0
Projective integrator based on Φ_{i-1} is Φ_i
Uses step size $h(k+q+M)^i$

What is stability? Consider Projective Forward Euler. Let amplification of level i integrator be σ_i and we have

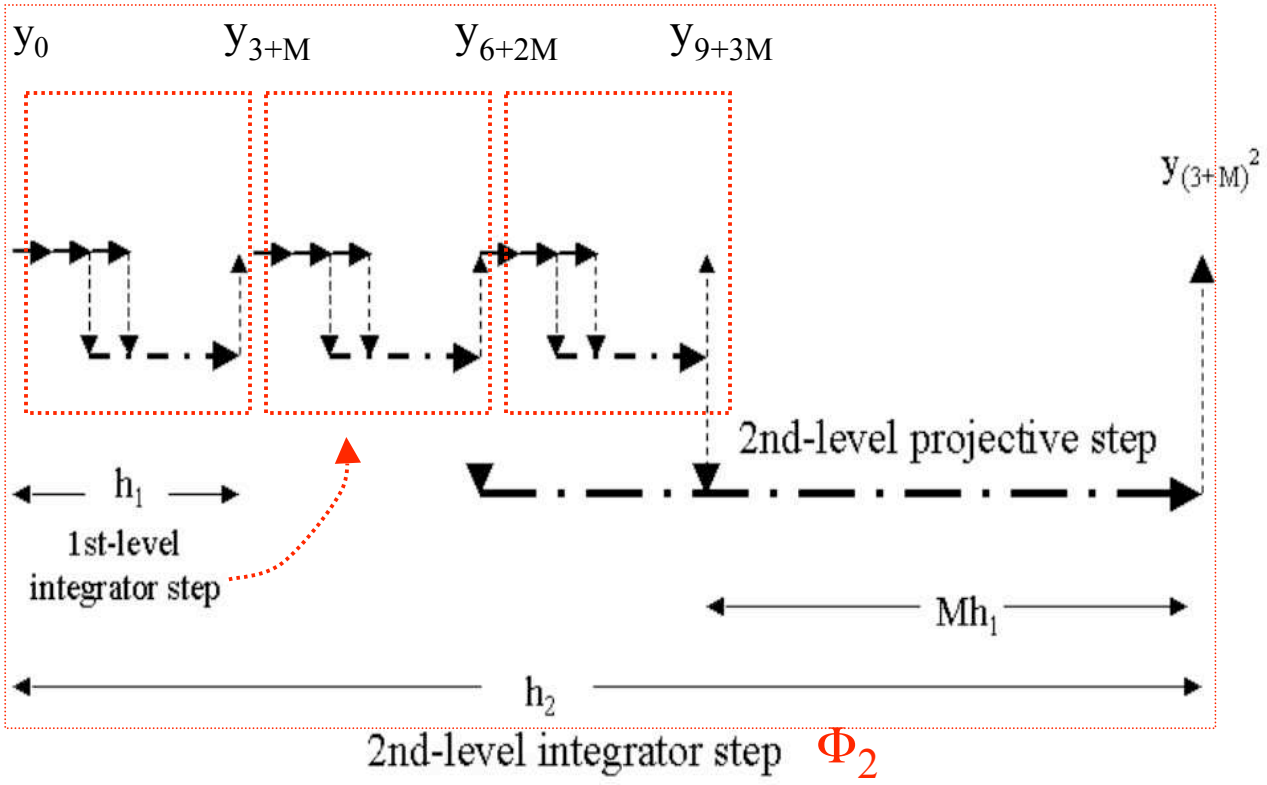
$$\sigma_{i+1} = \sigma_i^k[(M + 1)\sigma_i - M]$$

Stability region is set of σ in unit disk (same as bounded) in iteration

$$\sigma \leftarrow \sigma^k[(M + 1)\sigma - M]$$



Projective Forward Euler Method - linear fit to last two points



Two-level Projection method (k = 2 at both levels)

Telescoping Projective Methods:

What is stability? Consider Projective Forward Euler.

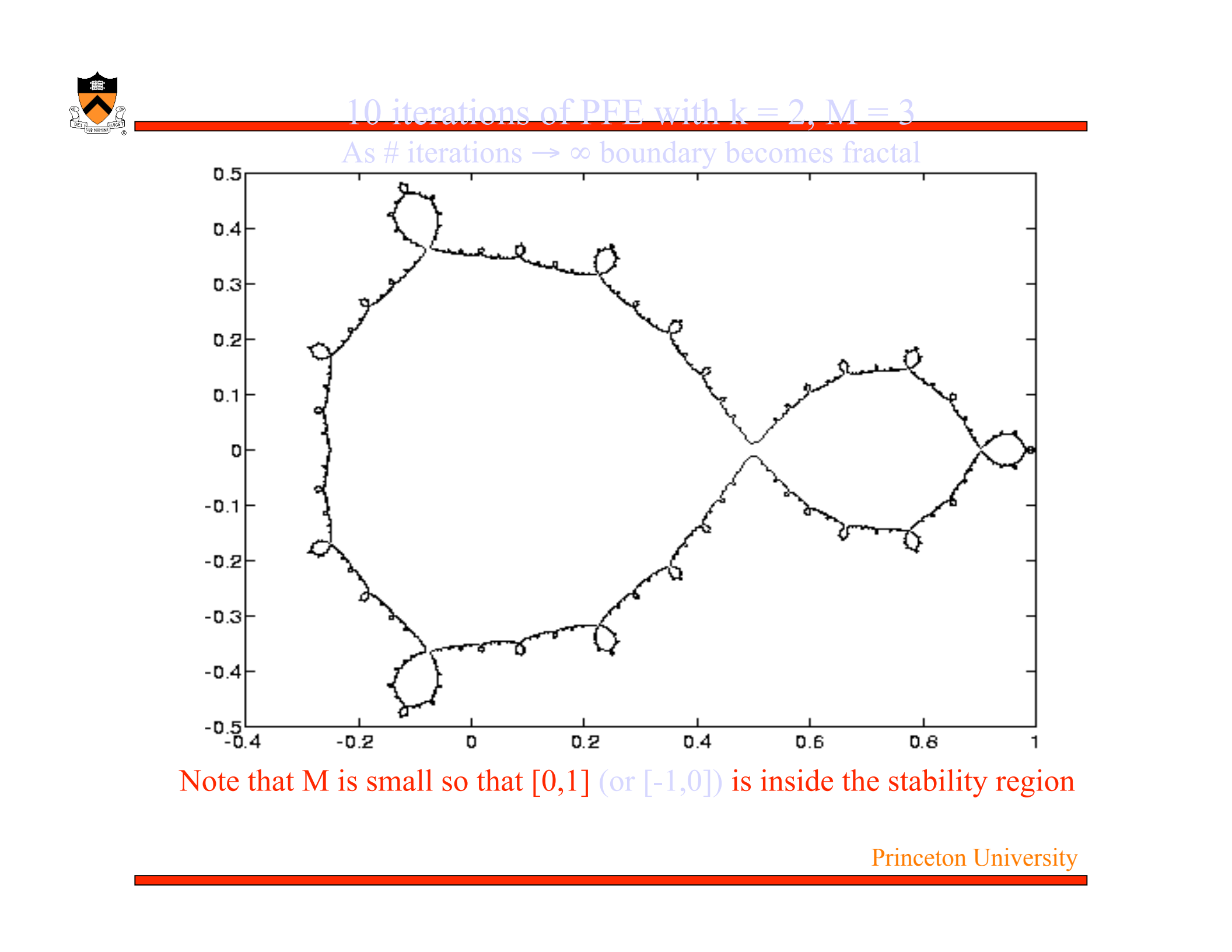
Let amplification of level i integrator be σ_i We have

$$\sigma_{i+1} = \sigma_i^k[(M + 1)\sigma_i - M]$$

Stability region is set of σ that remain in unit disk (or, remain bounded) in iteration

$$\sigma \leftarrow \sigma^k[(M + 1)\sigma - M]$$

This stability region will contain the stability region of the method with any finite number of iterations.



10 iterations of PFE with k = 2, M = 3

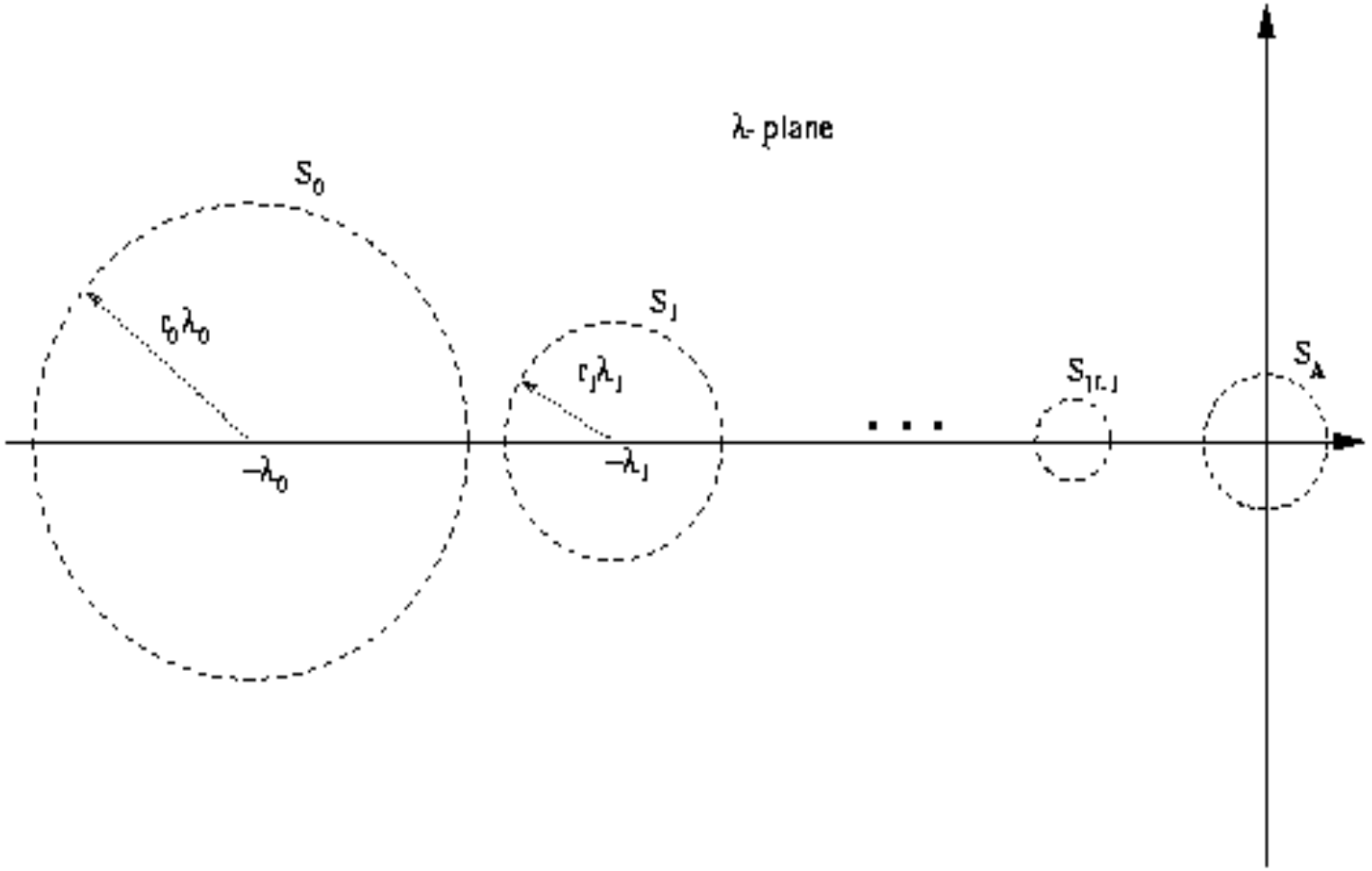
As # iterations → ∞ boundary becomes fractal

Note that M is small so that [0,1] (or [-1,0]) is inside the stability region

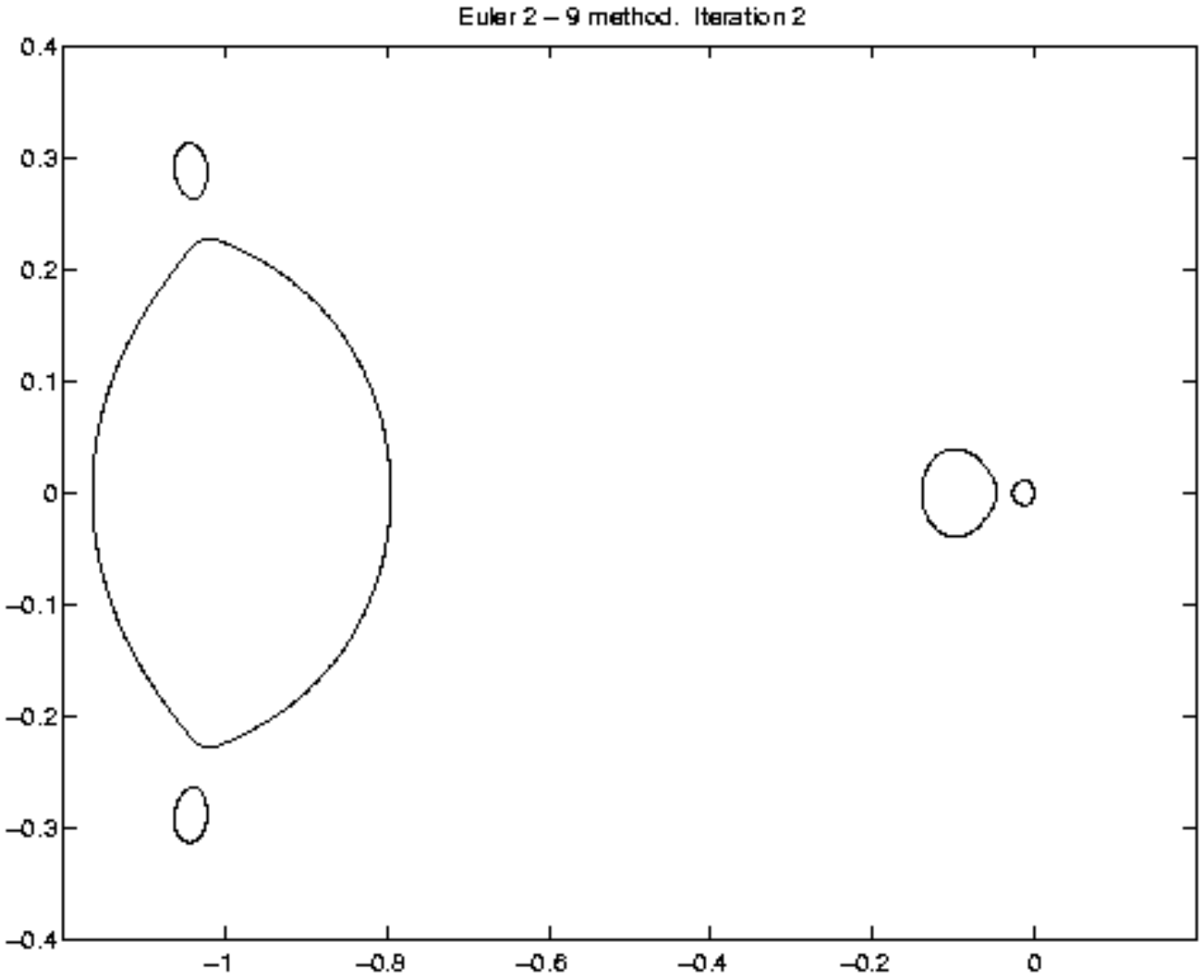
In some ways, these are like high-stage number Runge Kutta Methods which have been used to extend the region of stability.

A more interesting application may be to problems with multiple clusters of eigenvalues as shown on the next slide:

Suppose all eigenvalues lie in a union of disjoint disks:



A two-level PFE2-9 method

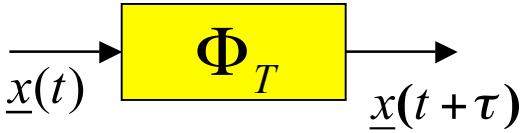


THE CONCEPT: What else can I do with an integration code ?

Have equation $\dot{\underline{x}} = \underline{f}(\underline{x})$

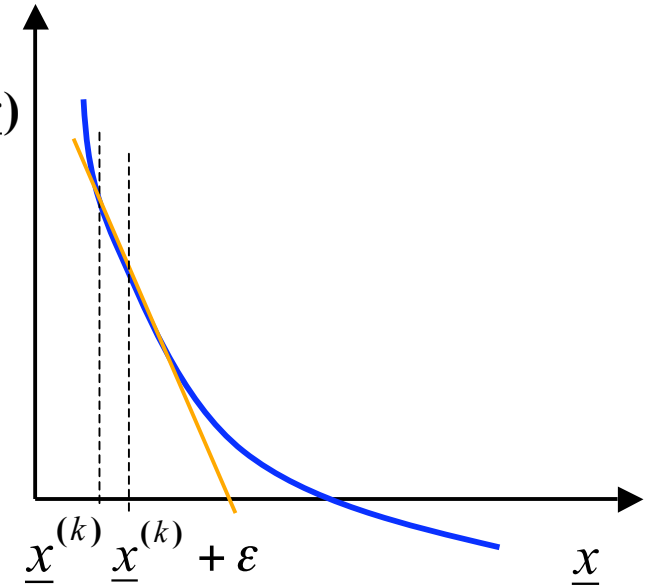
Write Simulation

Compile

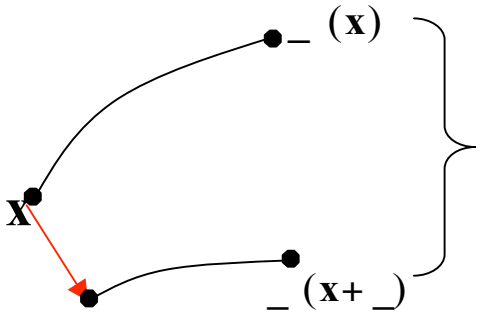


Do Newton on $\underline{x} - \Phi(\underline{x}) = 0$

Do Newton



Also



Estimate

matrix-vector product

⇩

Matrix free iterative linear algebra

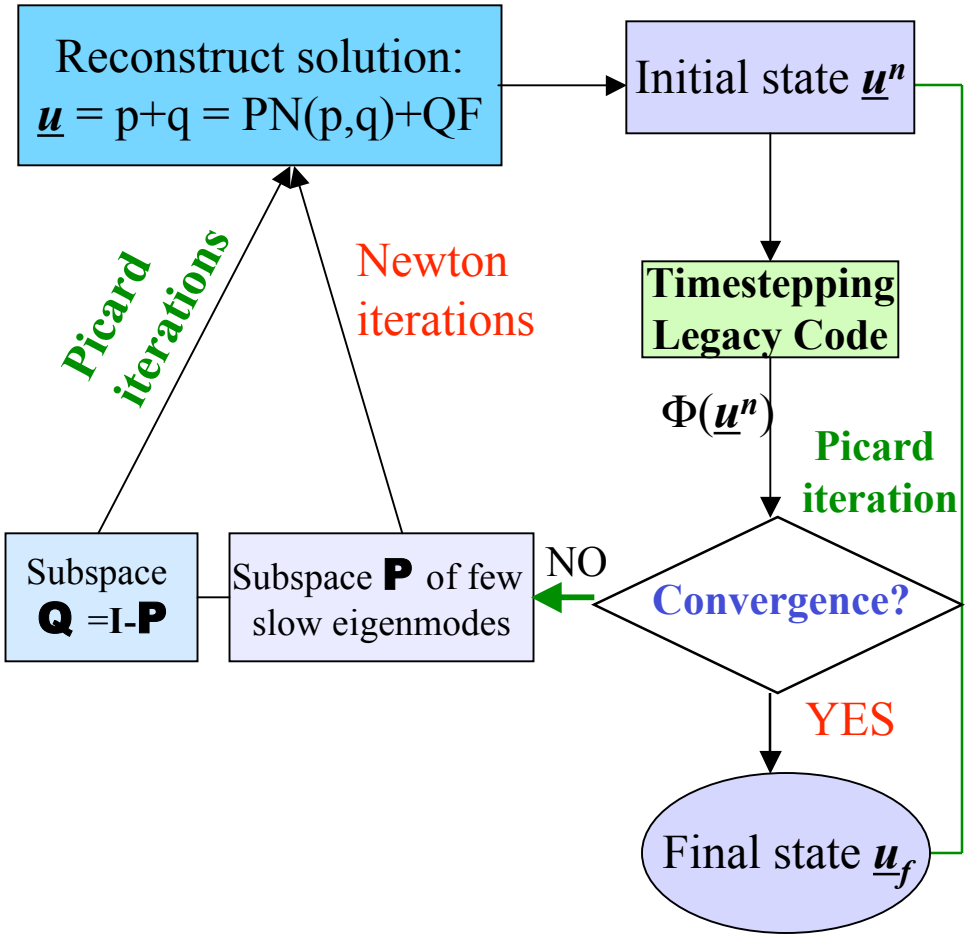
⇩

The World

⟺

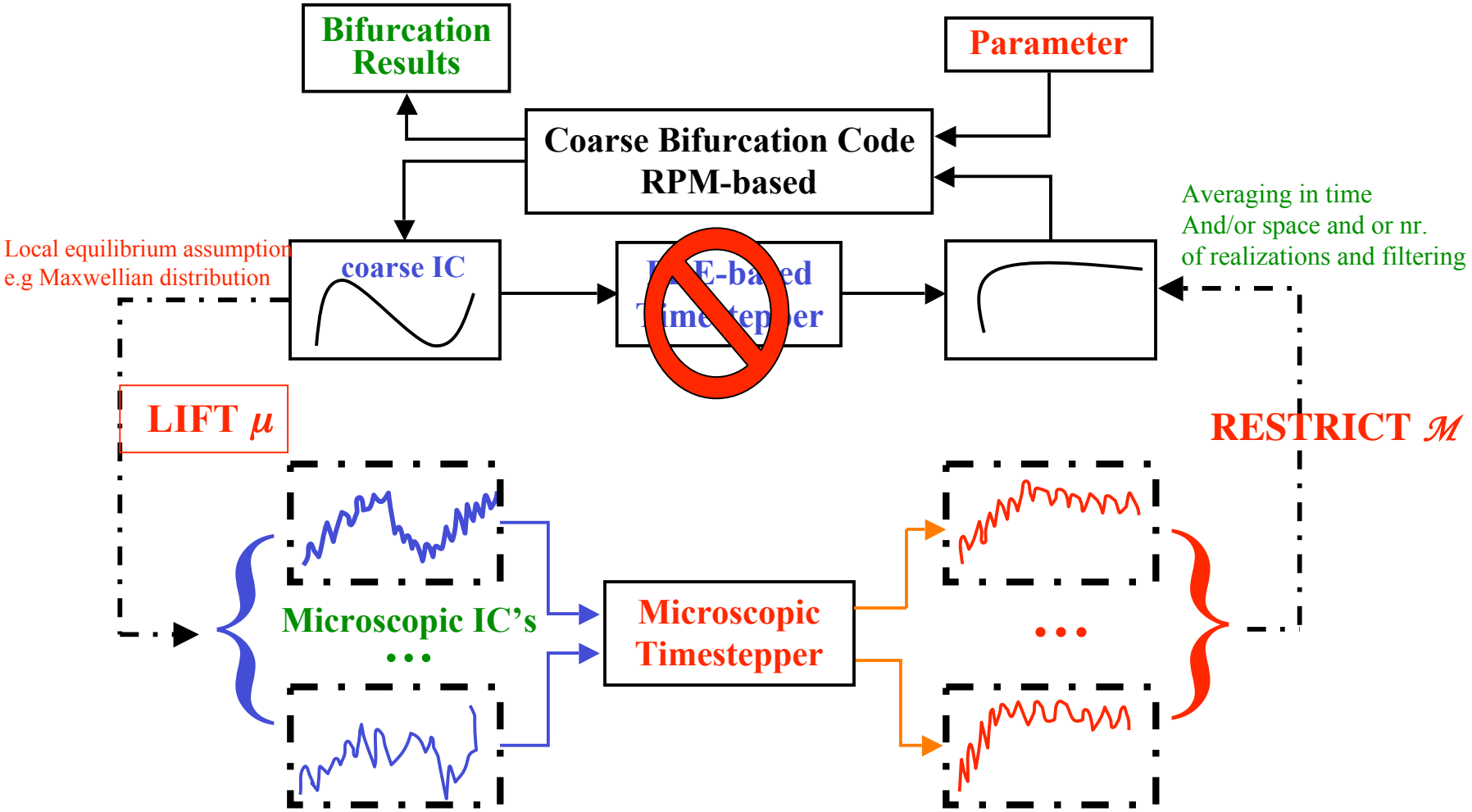
CG, GMRES Newton-Krylov

Recursive Projection Method (RPM) for $x-\Phi(x)=0$



- Treats timestepping routine, as a "black-box"
  - Timestepper evaluates $\underline{u}^{n+1}= \Phi(\underline{u}^n)$
- Recursively identifies subspace of slow eigenmodes, **P**
- Substitutes pure Picard iteration with
  - Newton method in **P**
  - Picard iteration in **Q** = I-**P**
- Reconstructs solution $\underline{u}$ from sum of the projections P and Q onto subspace **P** and its orthogonal complement **Q**, respectively

RPM for "Coarse" Bifurcations

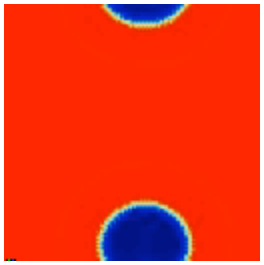


LB Simulations

g = 5e-5
Mo=1.64 e-4
Eo= 2.133

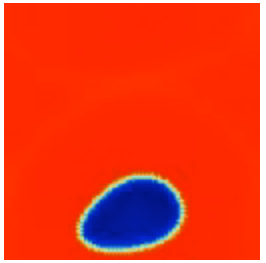
128 x 128 LB lattice points
9 LB unknowns per point
BUT 4 "coarse" unknowns
(2 densities, x-,y-momentum)

g = 1.25e-4
Mo=4.1e-4
Eo =5.334



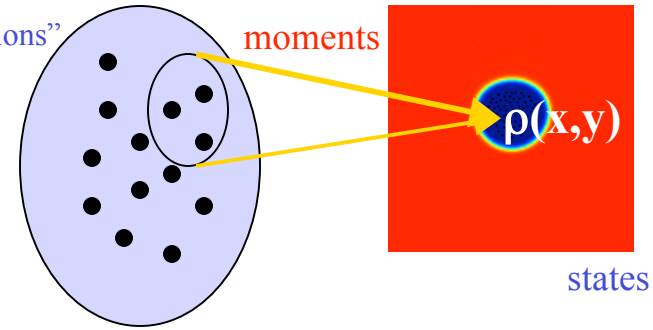
Parallel LB integration

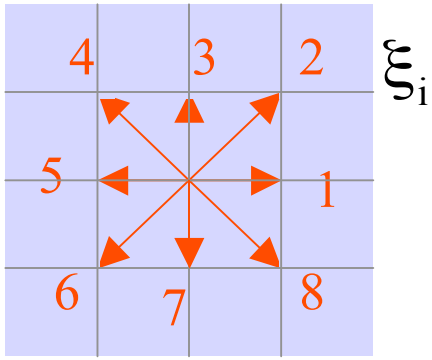
Using 8 PentiumIII processors



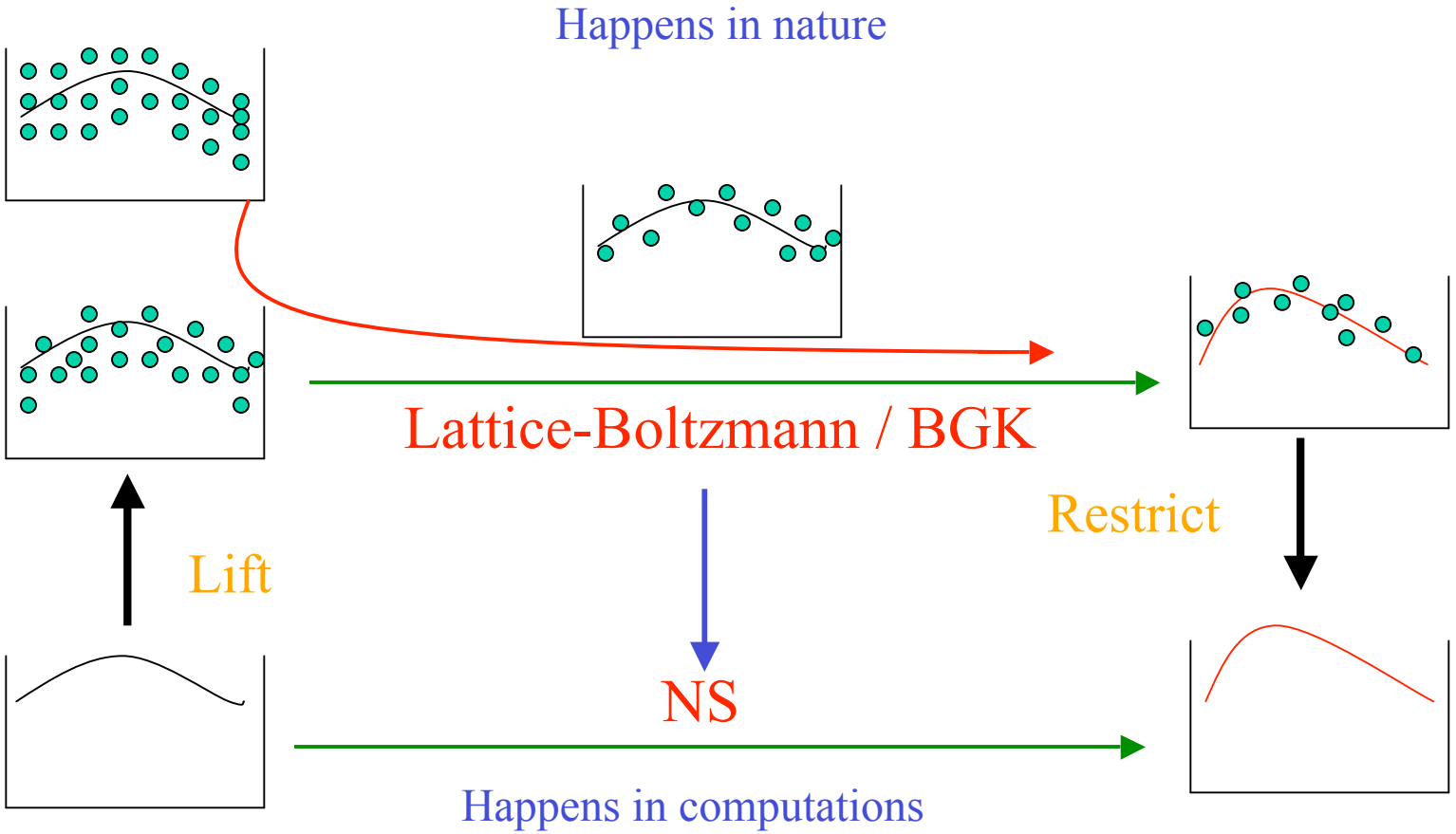
$$\sum f_i = \rho(x,y)$$

$$\sum f_i \xi_i = \rho \underline{u}(x,y)$$

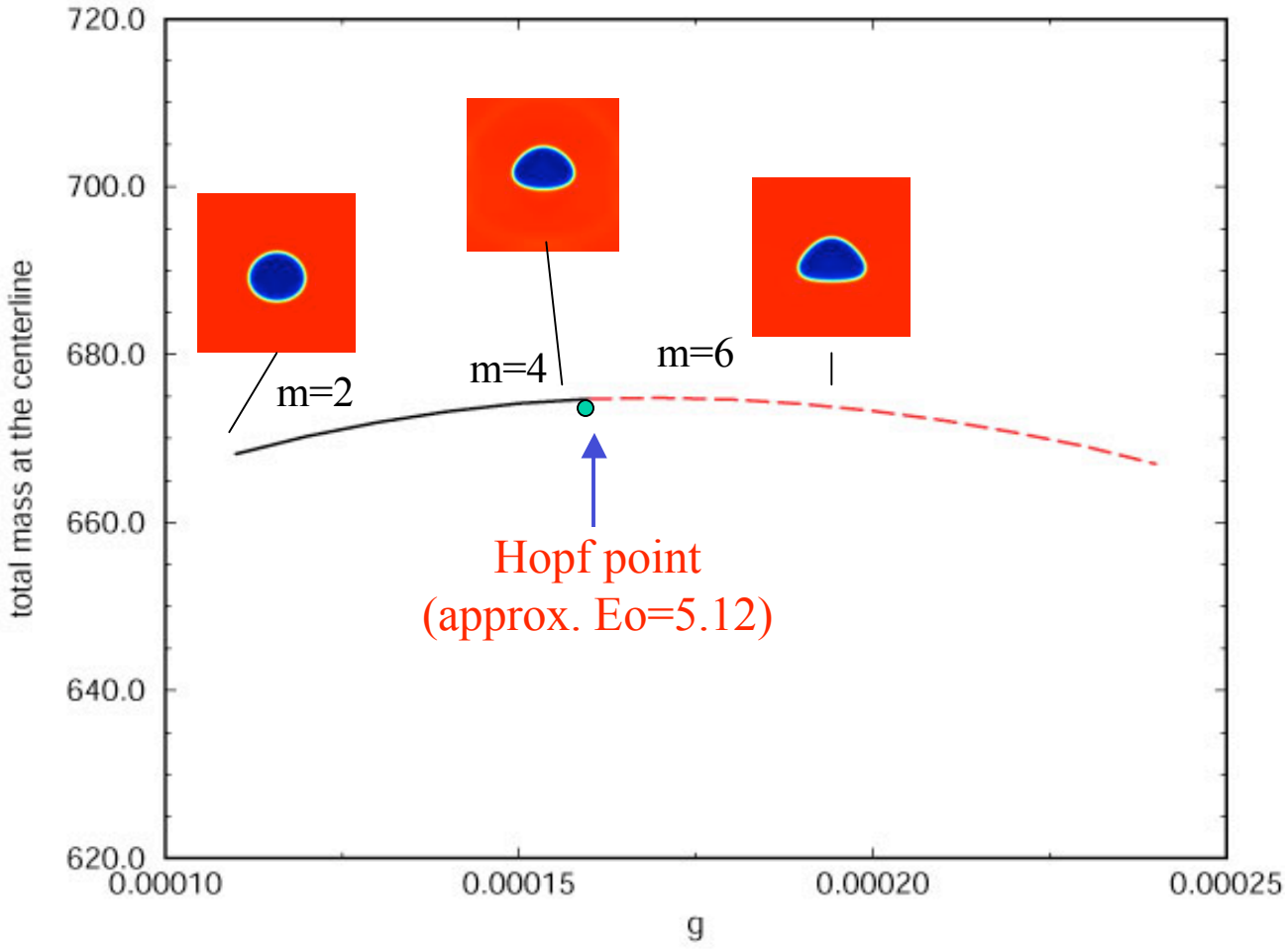




Coarse Behavior

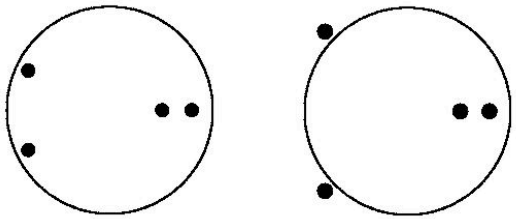


Bifurcation Diagram



Eigenspectrum Around Hopf Point

Eo=4.69 Eo=5.334



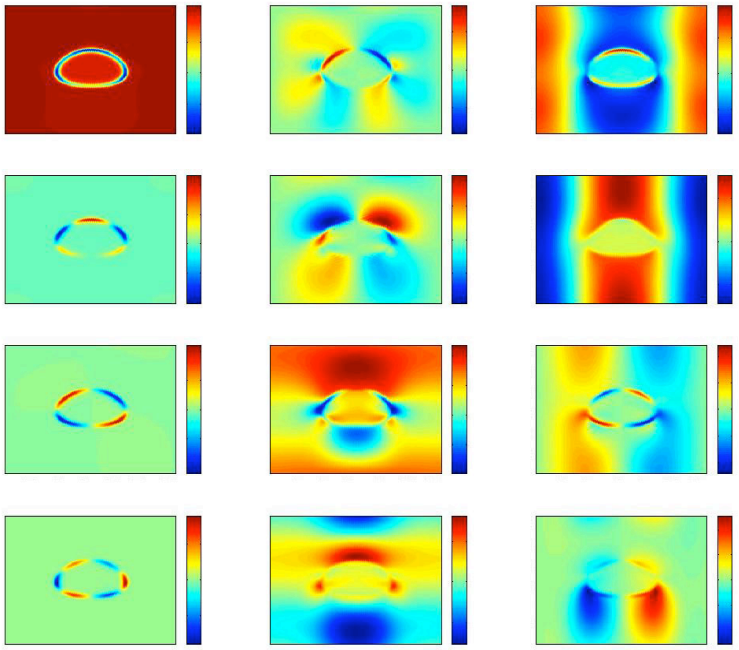
density x-momentum y-momentum

λ=0.7934

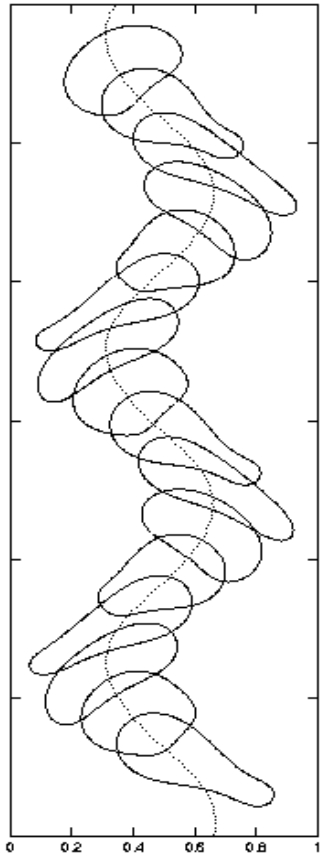
λ=0.5107

Re(λ)
=0.7934

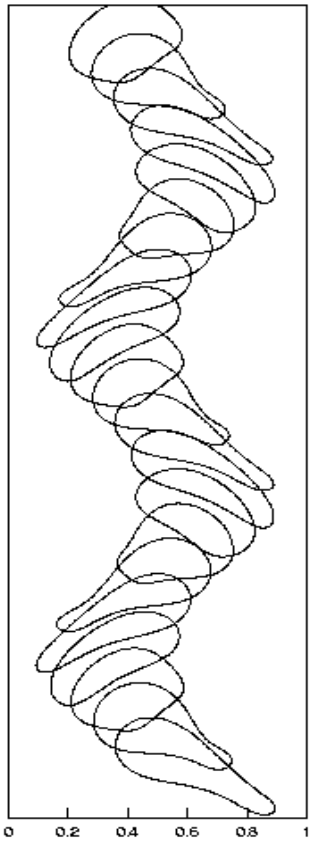
Im(λ)=0.77



FT/NS

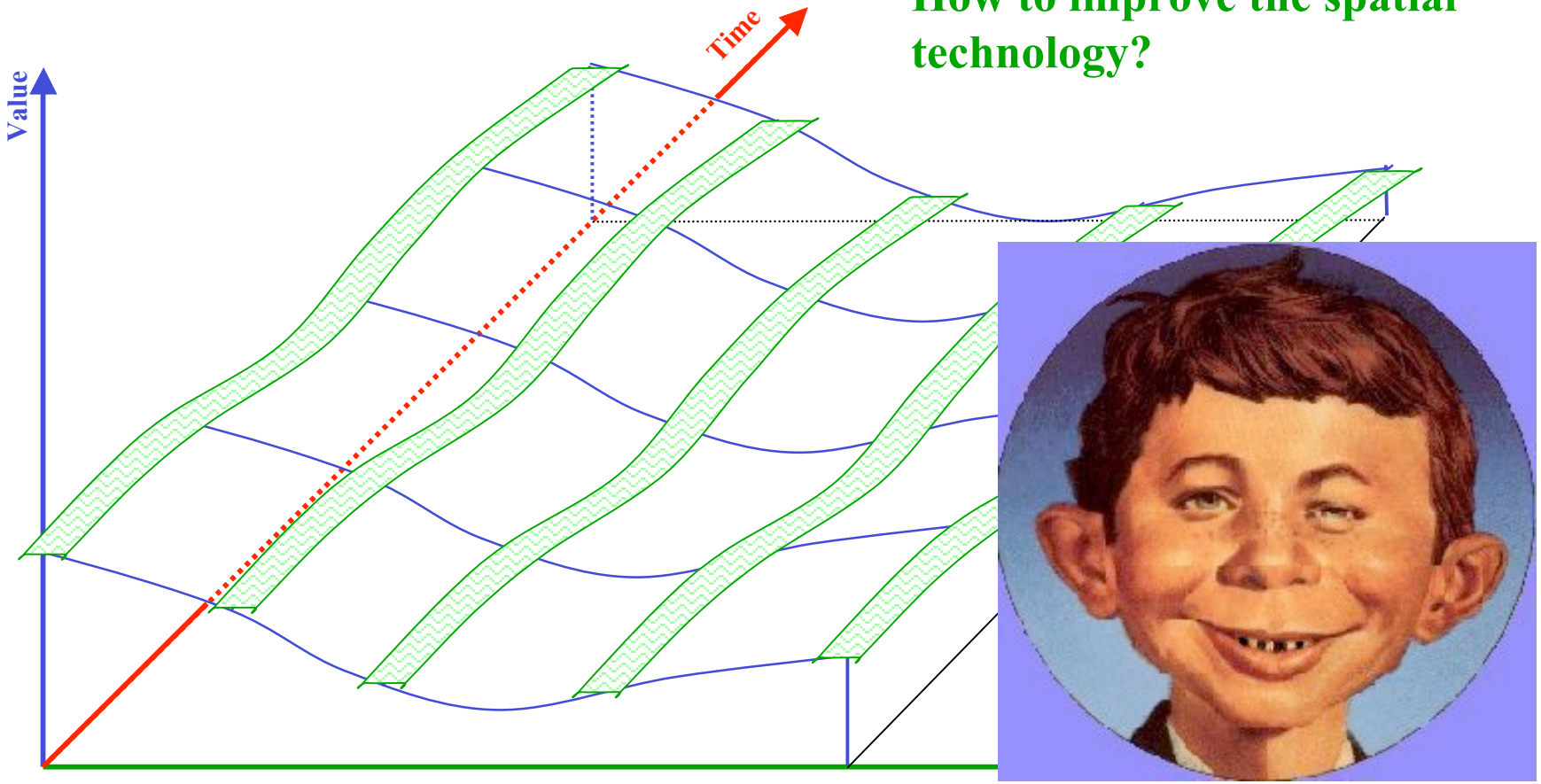


LB-BGK



Multiscale Modeling Challenges:

How to improve the spatial technology?



Proposal: detailed modeling in small spatial boxes with interpolation between boxes - the "gap-tooth scheme"

Gap-Tooth Scheme

Microscopic description in each tooth

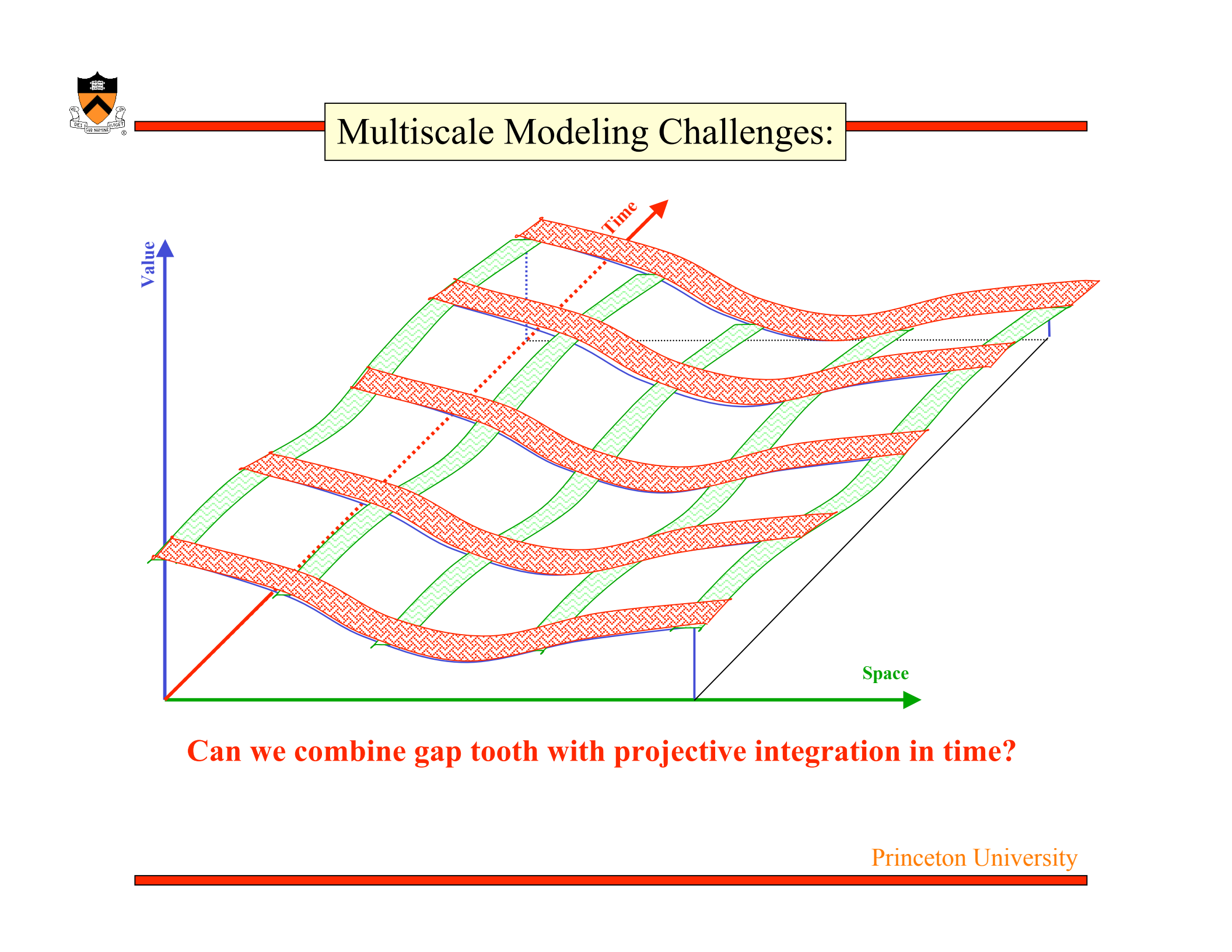
Boundary conditions on teeth edges
via interpolation of coarse quantities from neighboring teeth

Space

Ways to impose "coarsely inspired" boundary conditions

Motivated from Li & Yip, 1998: *Kevrekidis et al.,* **nlin.CD/0302055 at arXiv.org**

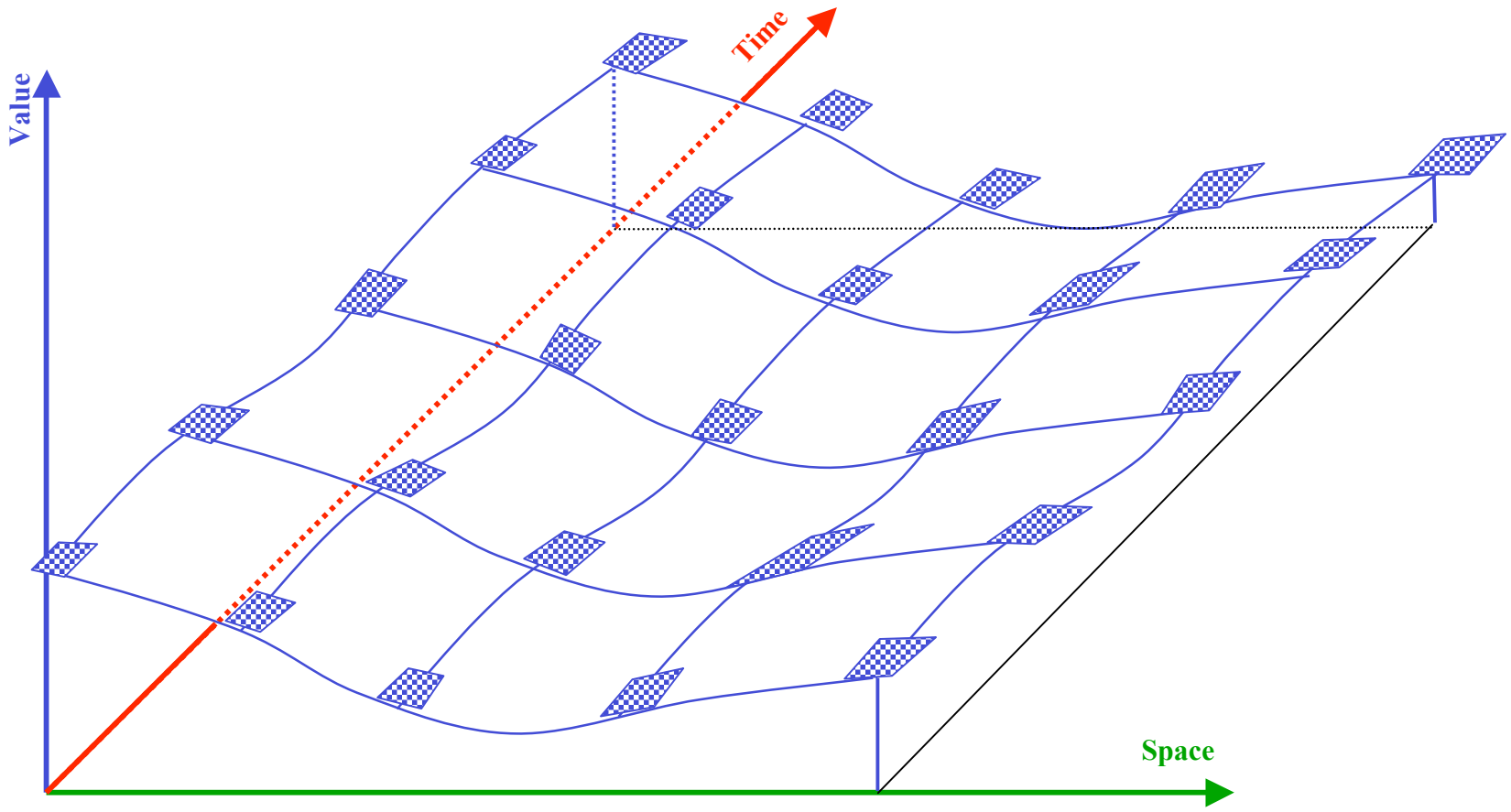
Gear, Li and Kevrekidis, **physics/0303010 at arXiv.org / PLA**



Multiscale Modeling Challenges:

Can we combine gap tooth with projective integration in time?

Multiscale Modeling Challenges:

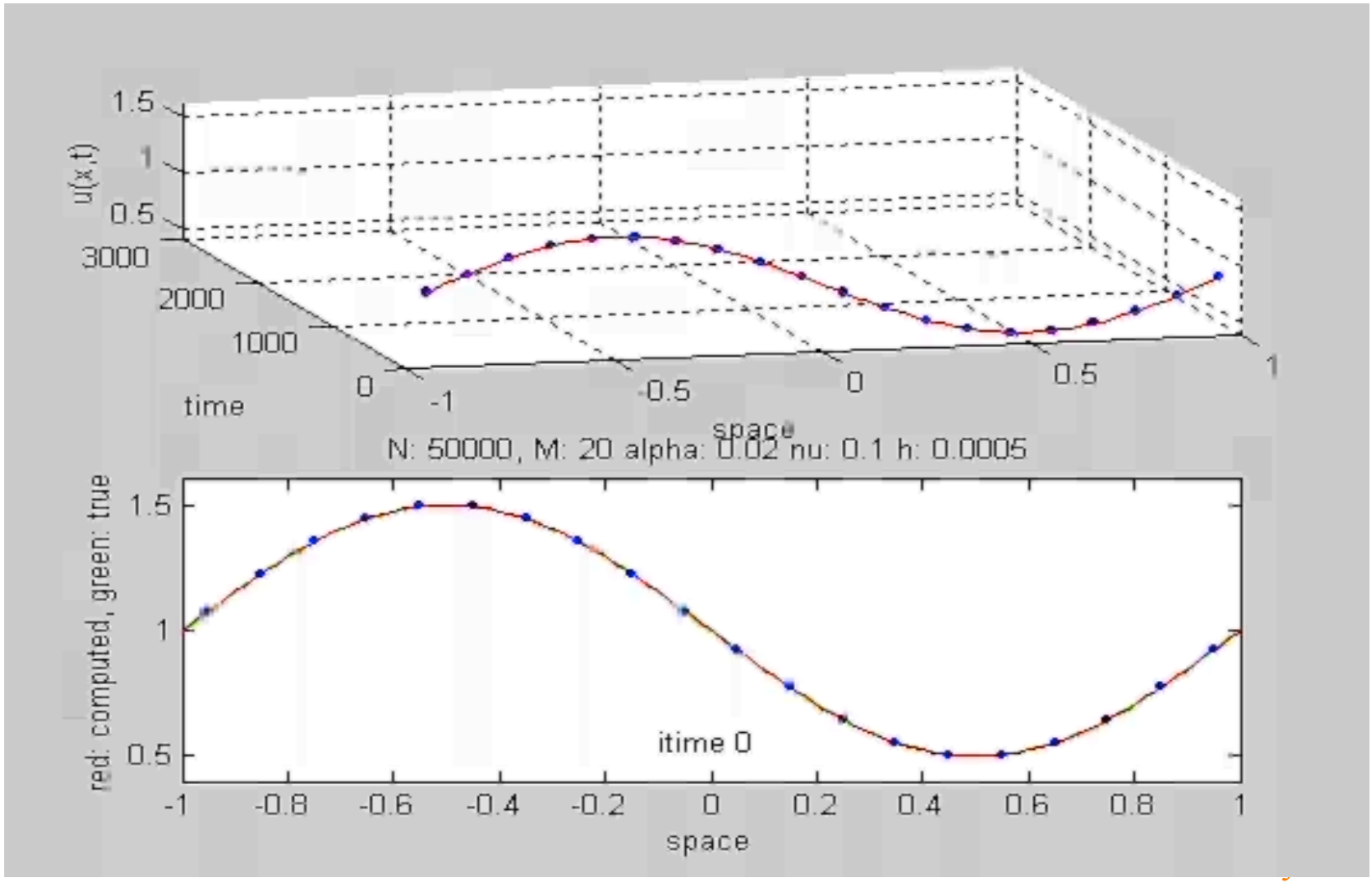


The "action" is going on at the intersection of the strips

- these are "microscopic" elements and, by interpolation and extrapolation, they are patched together over the full region

Viscous Burgers equation: kMC Realization

$$u_t + uu_x = vu_{xx}$$



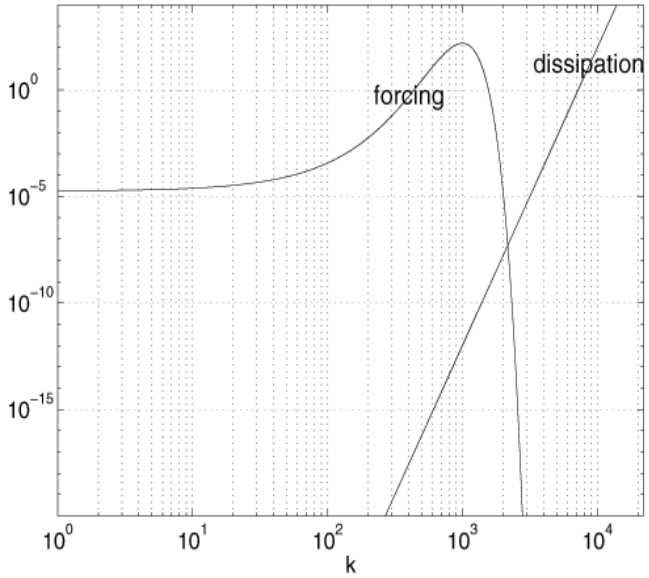
Randomly forced Burgers equation in 1-D – V.Yakhot

$$u_t + uu_x = (-1)^{n+1}\nu_{hyp}\nabla^{2n}u + f(x,t)$$

$$n = 7, \nu_{hyp} = 10^{-54}$$

- High Reynolds number regime modeled by a hyperviscosity term acting essentially at the smallest scales
- White-in-time forcing acting at scales much smaller than the size of the system

$$f(k,\omega)f(k',\omega') = D_0 e^{-\frac{(k-k_f)^2}{\sigma_f^2}}\delta(k-k')\delta(\omega-\omega')$$



Wavenumber, k

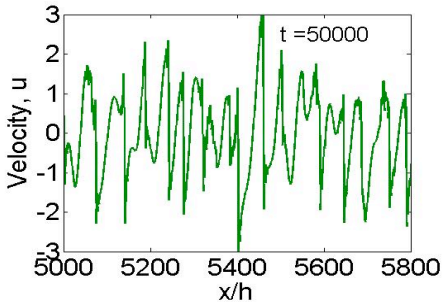
Observables are:

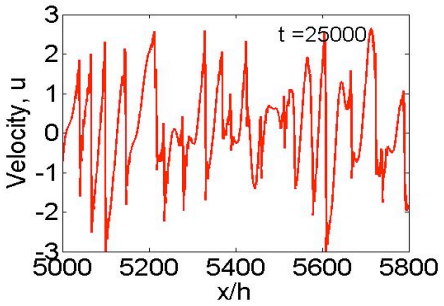
$$E(k,t) = \langle u(k,t)u^*(k,t)\rangle$$

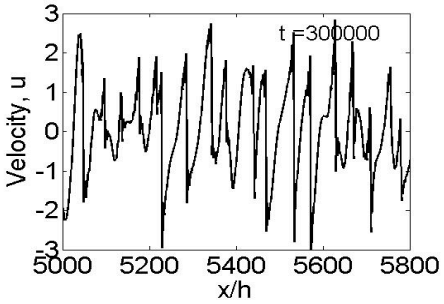
$$S_3(r,t) = \langle (u(x,t) - u(x+r,t))^3 \rangle$$

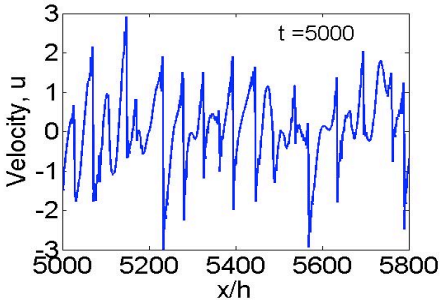
Why *E(k)*?

- Velocity field *u(x)* is stochastic and consists of tiny shocks
- The fields at different times "look" the same



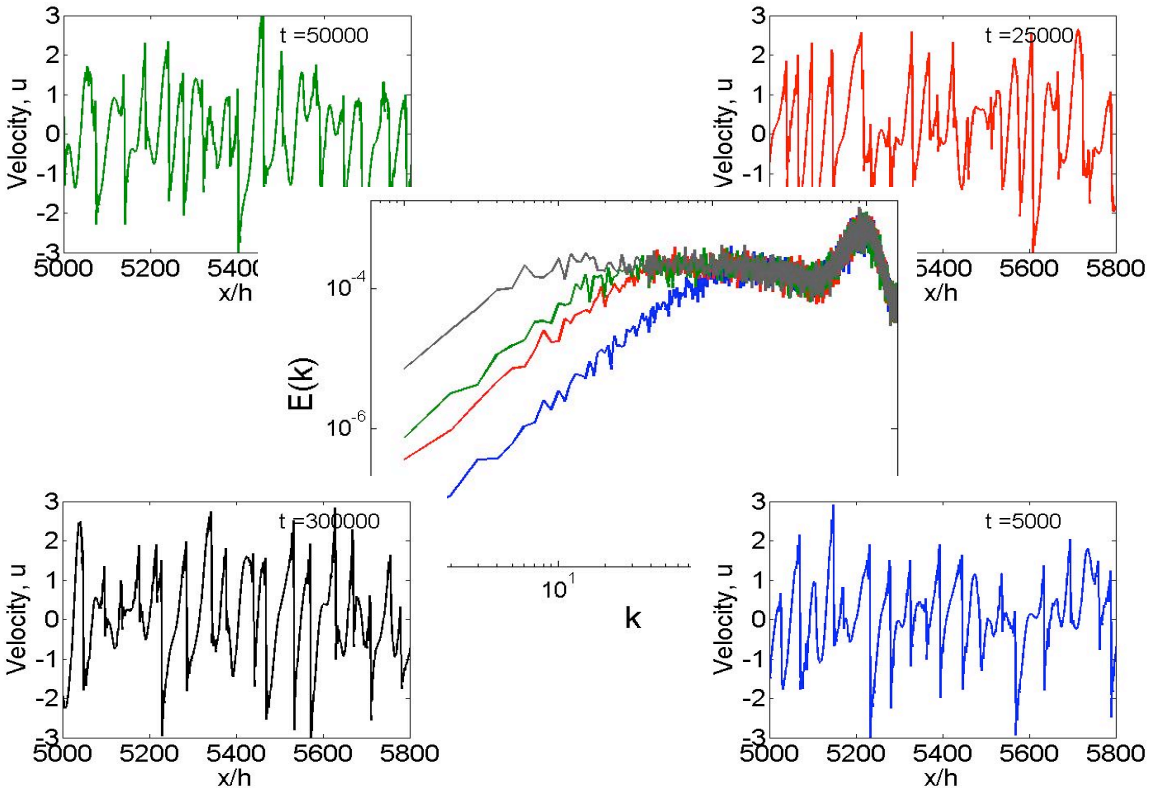






Why *E(k)*?

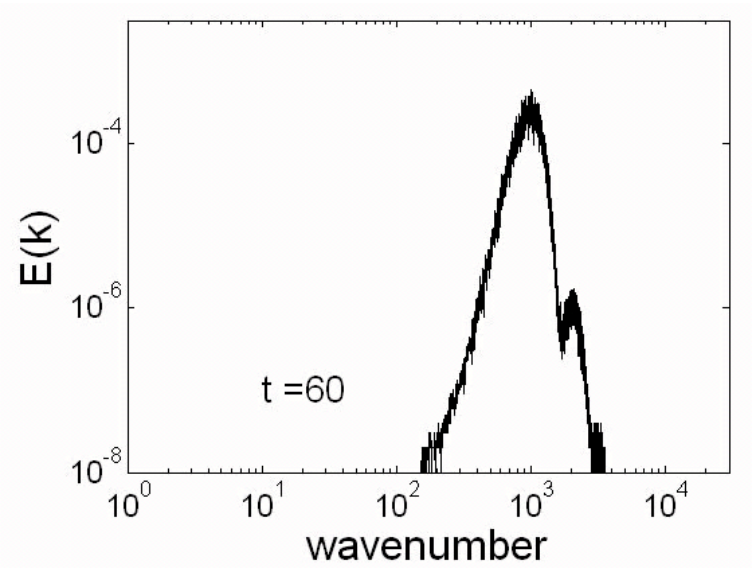
- Velocity field *u(x)* is stochastic and consists of tiny shocks
- The fields at different times "look" the same
- Energy spectrum enables us to distinguish between the fields
- "Coarse Evolution" of *E(k)* appears deterministic

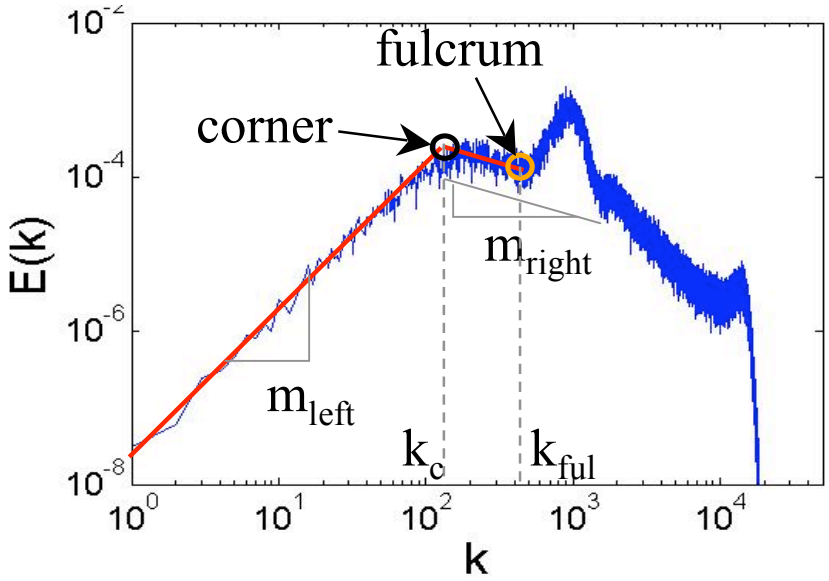


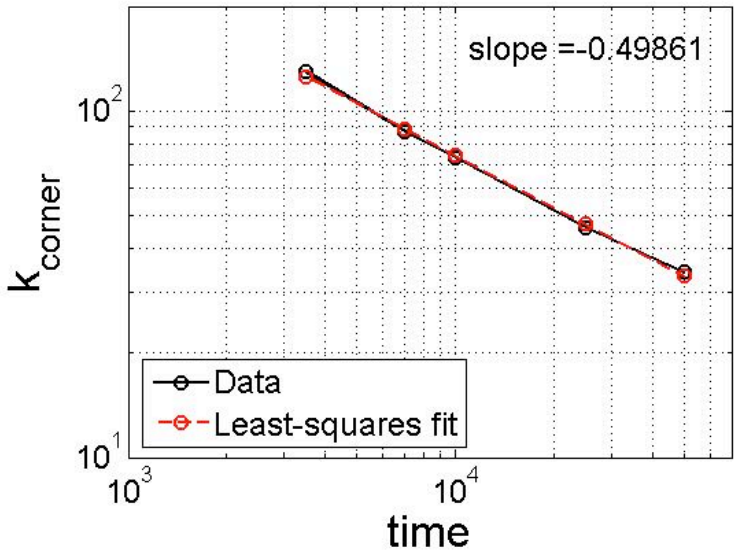
Behavior of *E(k)*

- Fast evolution for the large wavenumbers to stationarity - THEN slow –and slower- evolution for small wavenumbers
- Steady state for small wavenumbers:

  E(k)=const
- Transient evolution in the small wavenumber region can be described by two straight lines

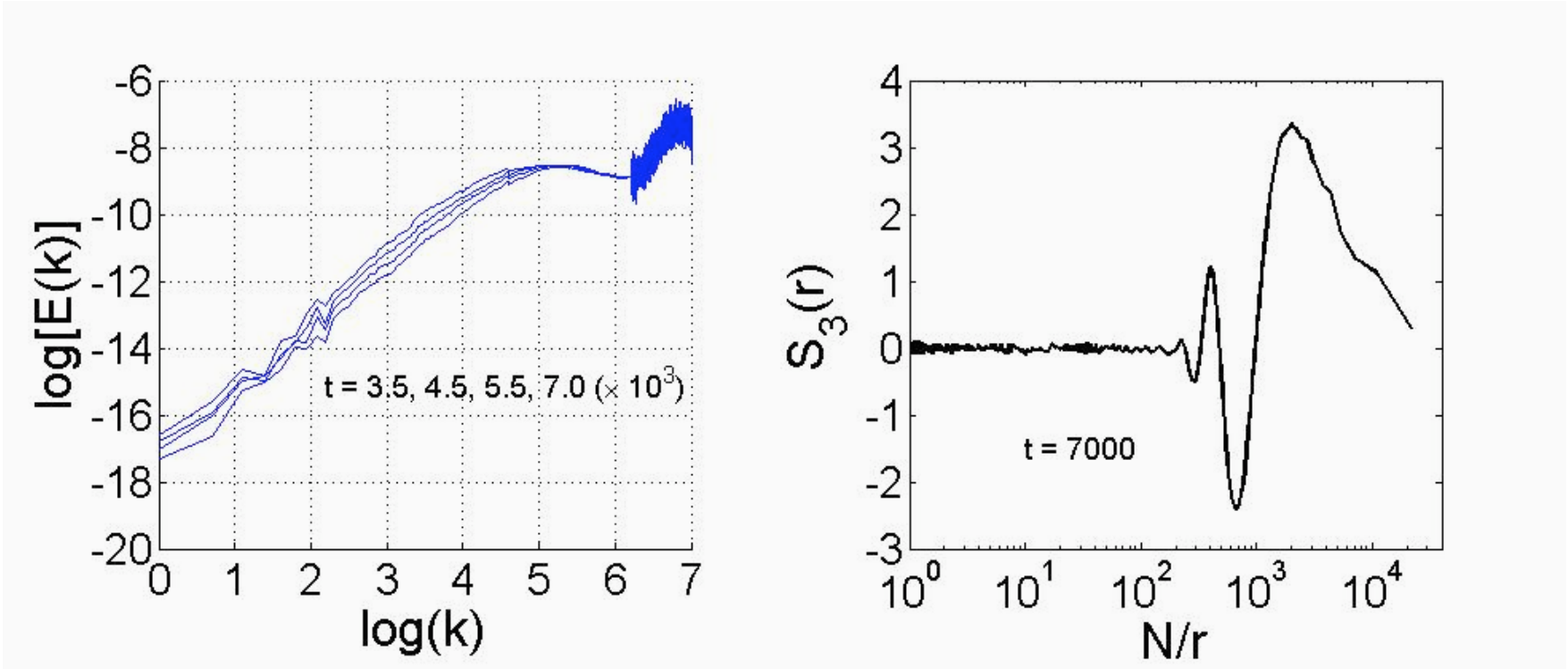






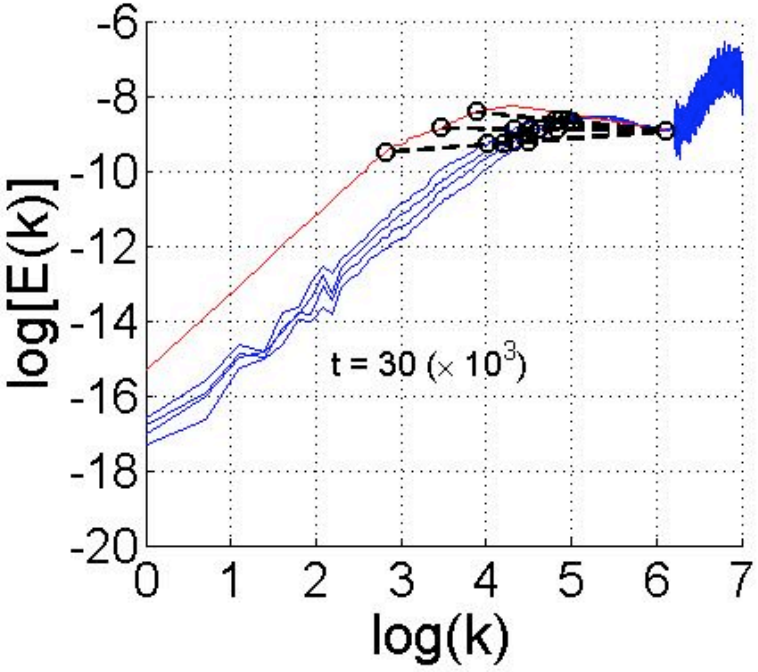
Projective integration

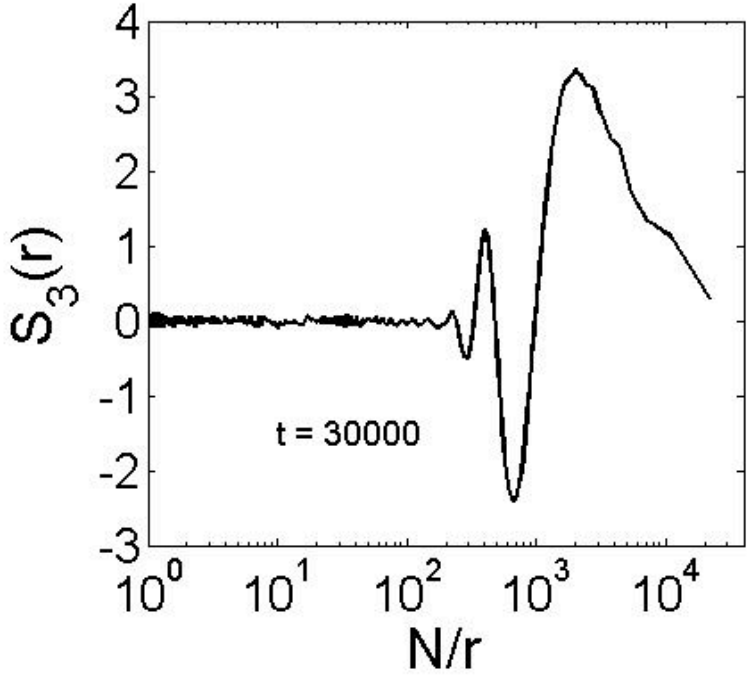
1. Run the simulation for short time and get the averaged stretching factor



Projective integration

1. Run the simulation for short time and get the averaged stretching factor
2. Project *E(k)* using the predicted value of the stretching factor (LIFT)

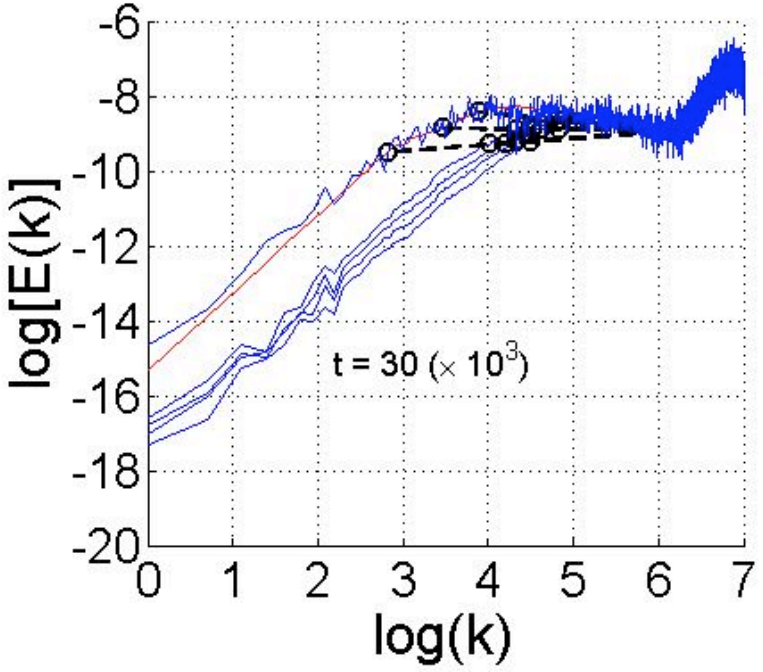


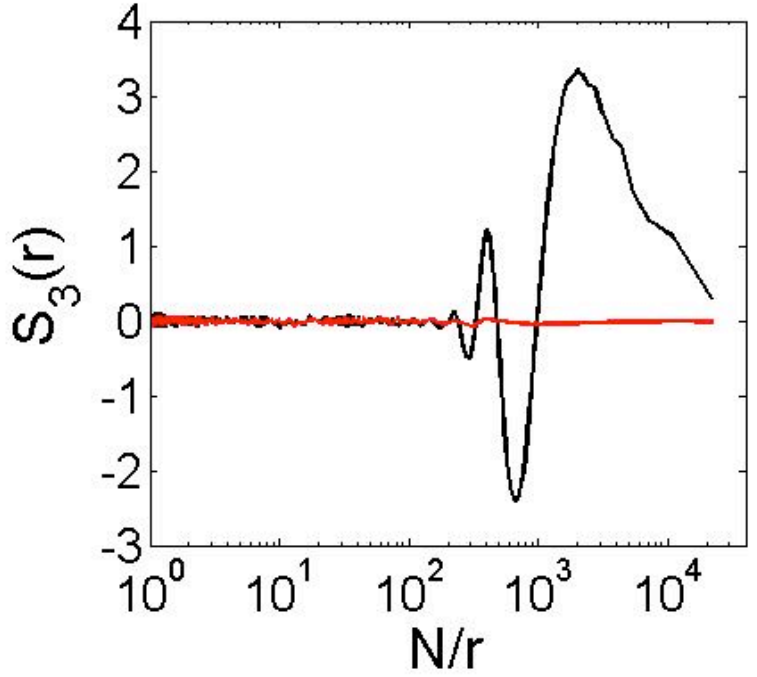


Projective integration

1. Run the simulation for short time and get the averaged stretching factor
2. Project *E(k)* using the predicted value of the stretching factor (LIFT)
3. Randomize the phases to generate new initial conditions

   $S_3(r)$ goes to zero

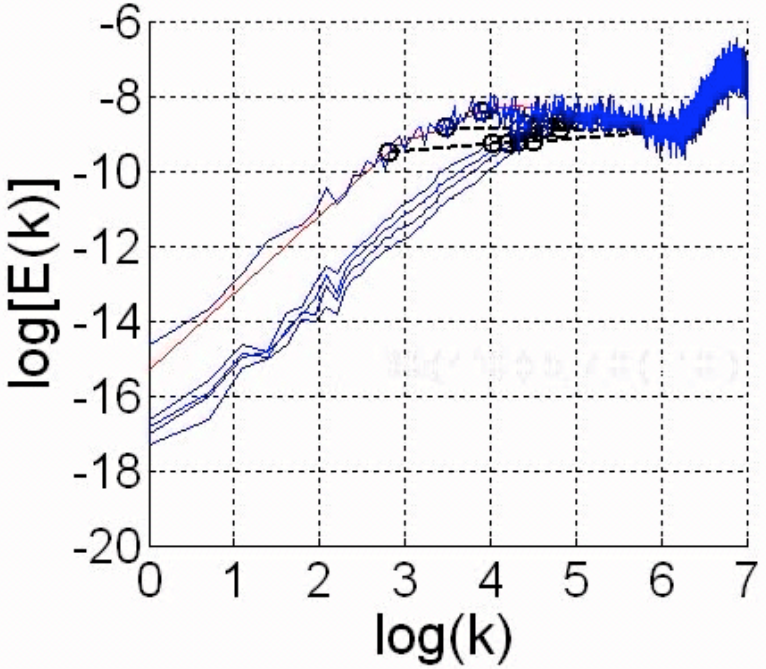


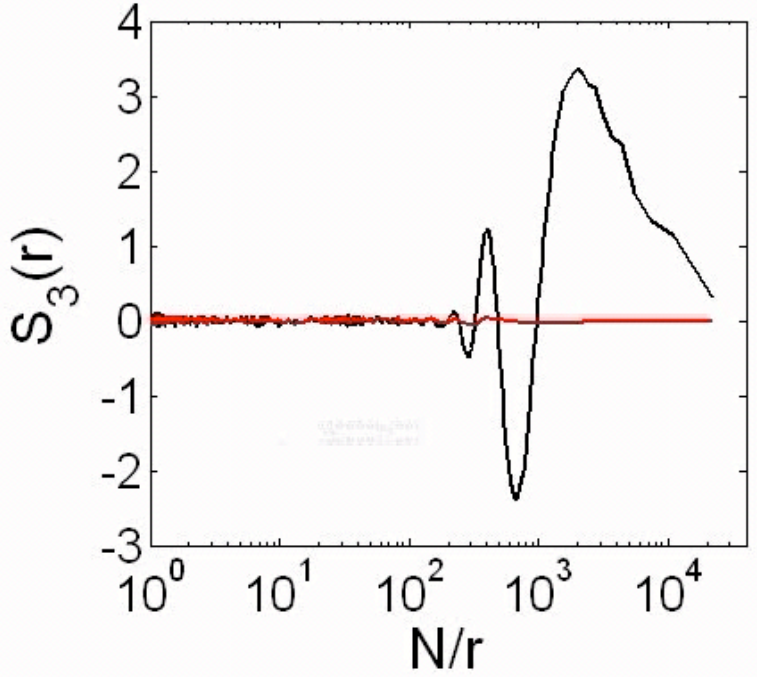


Coarse Projective integration

1. Run the simulation for short time and get the averaged stretching factor
2. Project *E(k)* using the predicted value of the stretching factor (LIFT)
3. Randomize the phases to generate new initial conditions

   $S_3(r)$ goes to zero
4. Continue the simulation with new ic's
5. $S_3(r)$ gets enslaved to the steady state value in less than 100 time steps

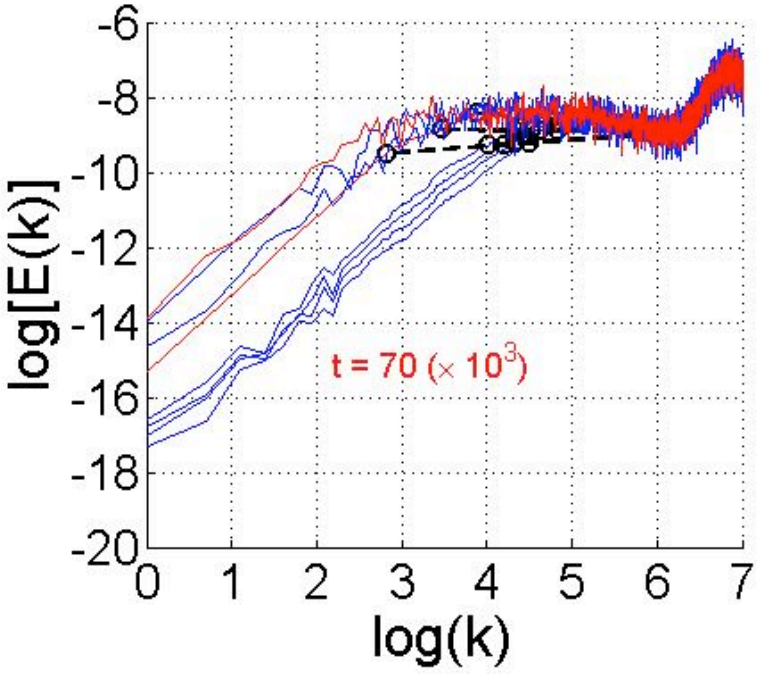


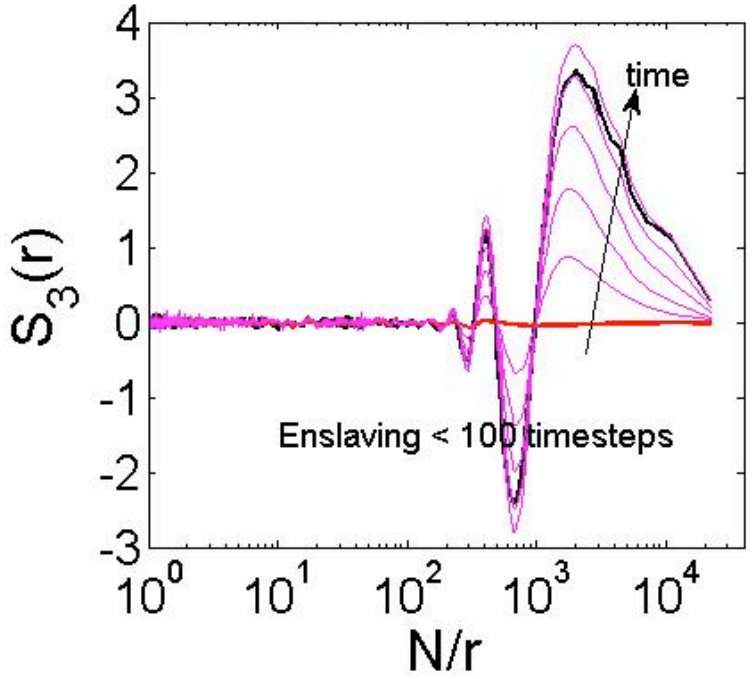


Projective integration

1. Run the simulation for short time and get the averaged stretching factor
2. Project *E(k)* using the predicted value of the stretching factor (LIFT)
3. Randomize the phases to generate new initial conditions

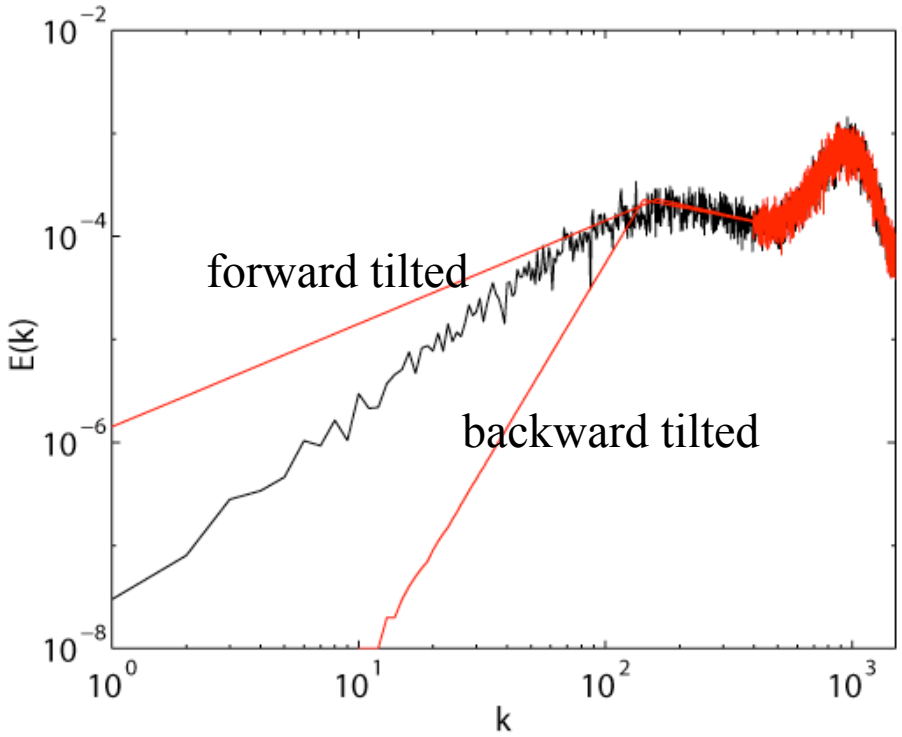
   $S_3(r)$ goes to zero
4. Continue the simulation with new ic's
5. $S_3(r)$ gets enslaved to the steady state value in less than 100 time steps
6. *E(k)* thus obtained evolves with the original simulation





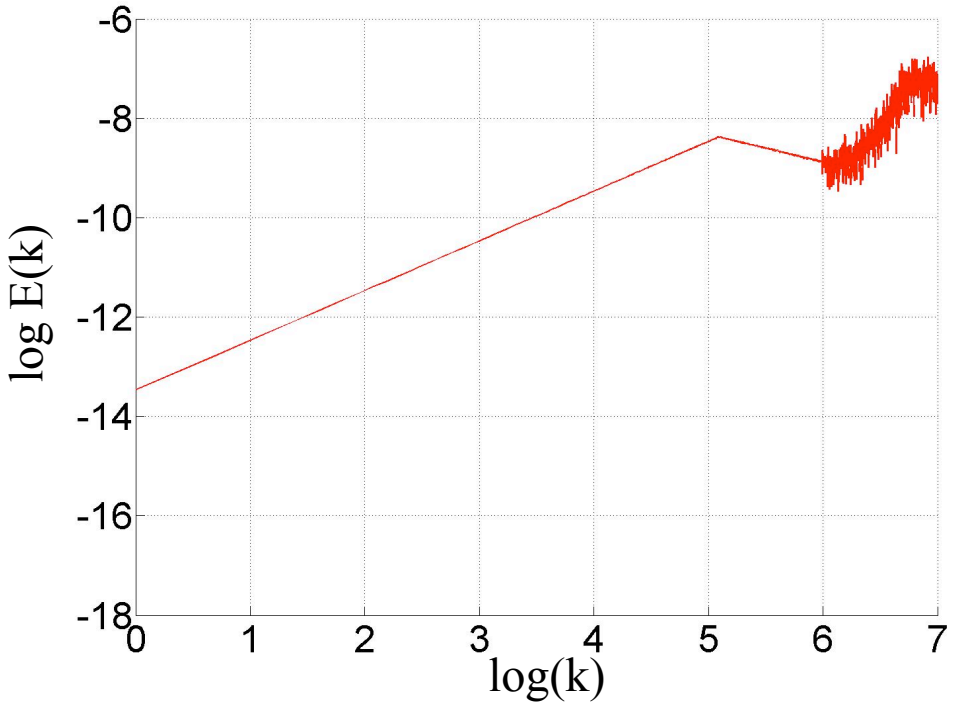
"Non self-similar" initial condition

- Projective integration designed for initial conditions in the "self-similar" regime
- Other initial conditions *renormalized* using *run and restrict* algorithm
- *Forward-tilted* and *backward-tilted* initial conditions considered



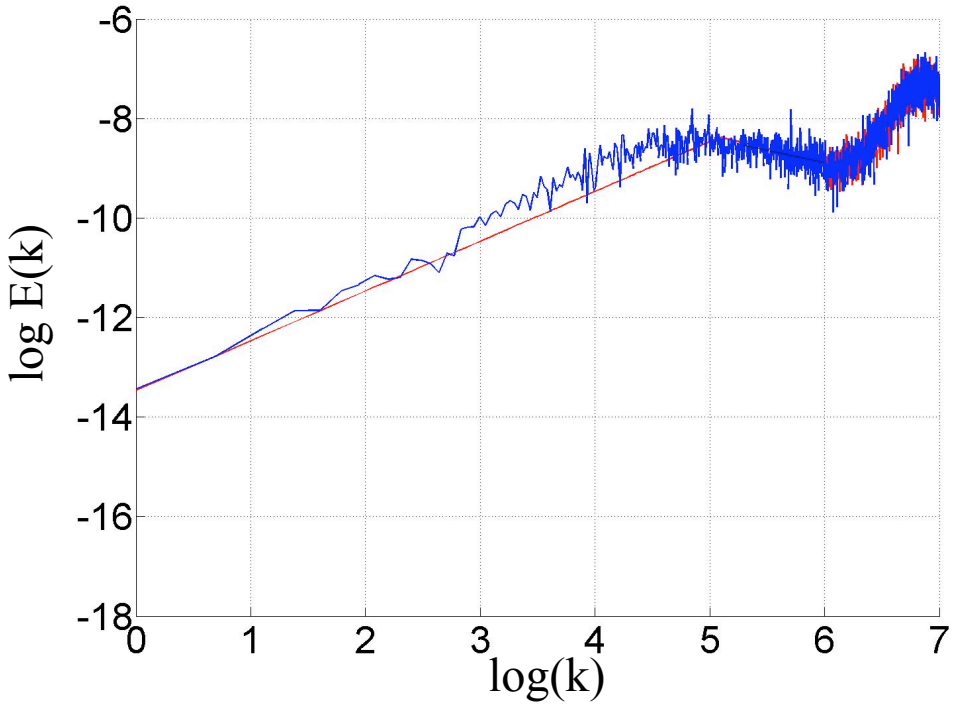
Renormalization algorithm

- A simplified initial condition



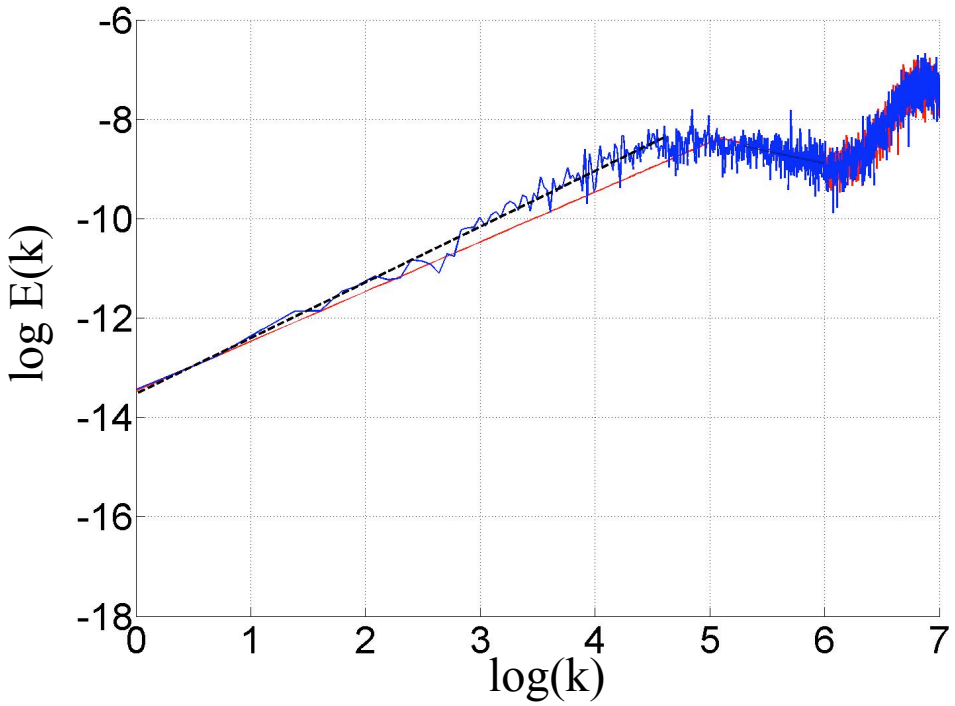
Renormalization algorithm

- A simplified initial condition
- Run for short time (5000 time steps)



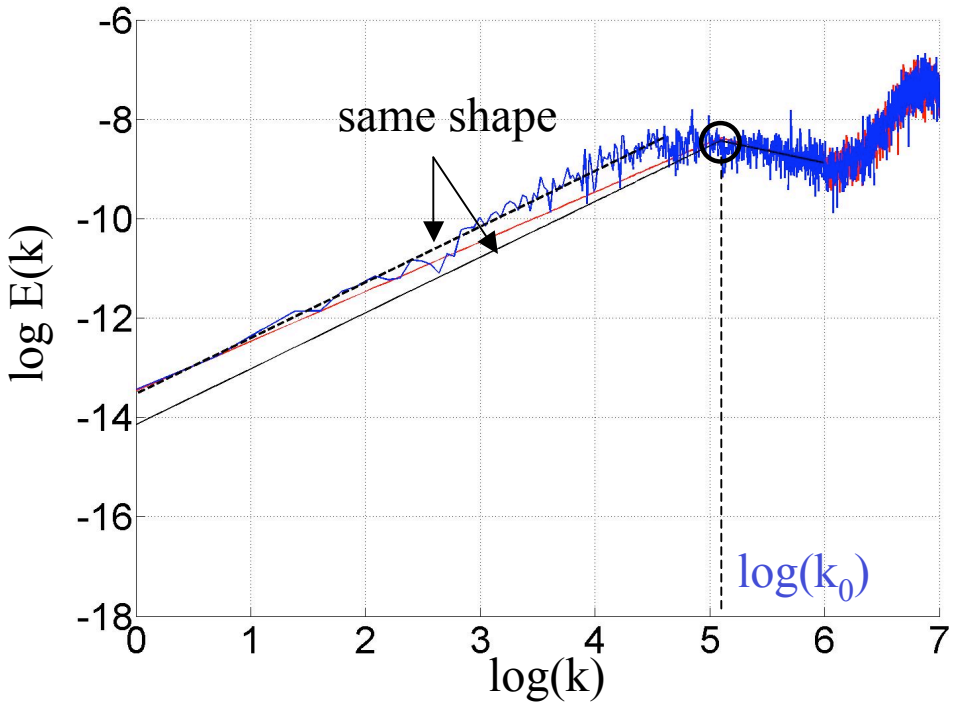
Renormalization algorithm

- A simplified initial condition
- Run for short time (5000 time steps)
- Approximate the spectrum by straight lines



Renormalization algorithm

- A simplified initial condition
- Run for short time (5000 time steps)
- Approximate the spectrum by straight lines
- Pull back the shape keeping the initial ordinate fixed

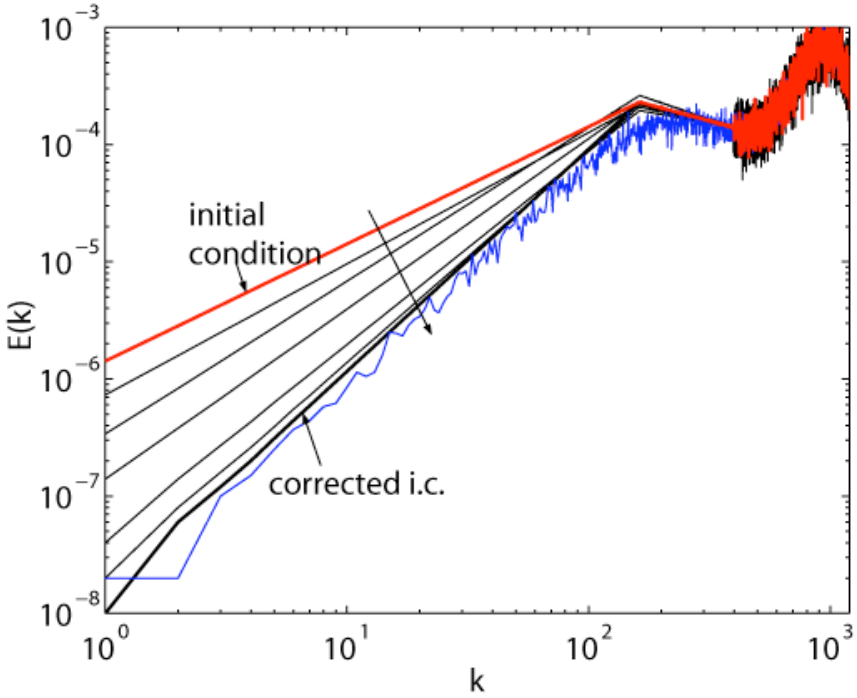


Renormalization algorithm

- Sequence of iterations leading to the right shape
- Comparison with a representative spectrum

Forward tilted i.c.

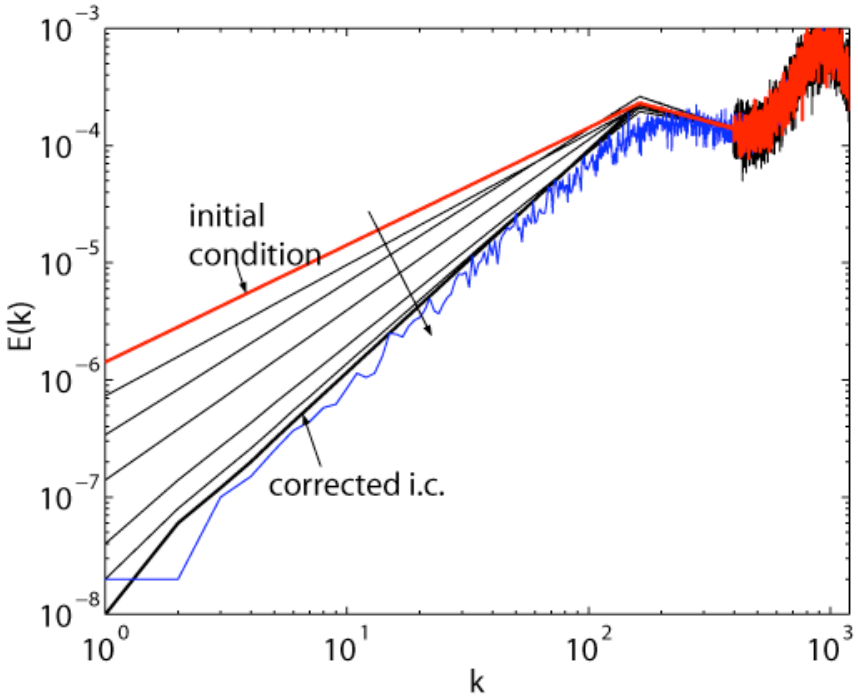
- convergence in 6-8 iterations



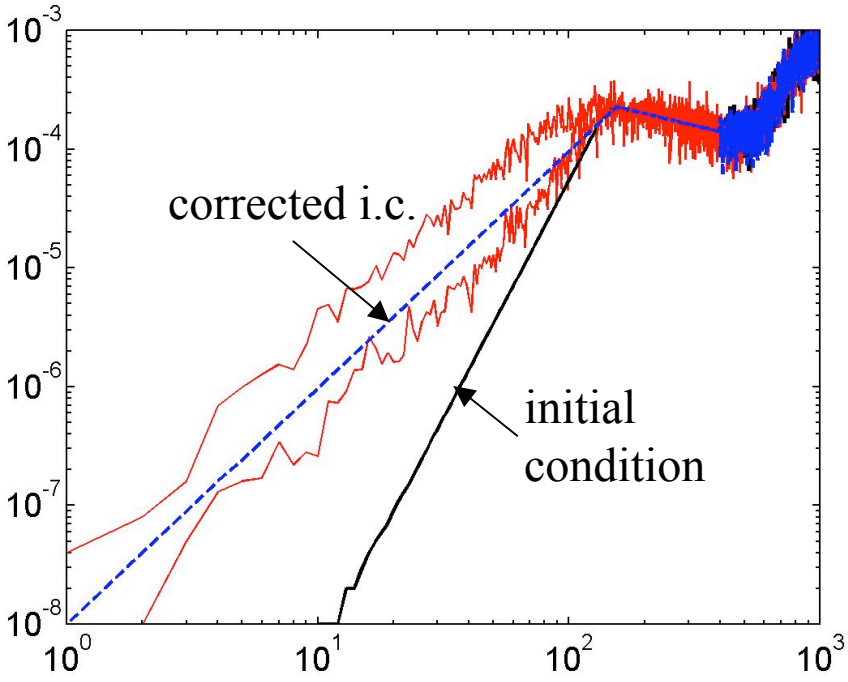
Renormalization algorithm

- Sequence of iterations leading to the right shape (cf. NLS work, LPMSS)
- Comparison with a representative spectrum

Forward tilted i.c.
- convergence in 6-8 iterations



Backward tilted i.c.
- convergence in 1 iteration



Coming full circle

No equations ?

Isn't that a little medieval ? Equations = "Understanding"

AGAIN matrix free iterative linear algebra

$$A\,x = b$$

PRECONDITIONING, $B\,A\,x = B\,b$

B approximate inverse of A

Use "the best equation you have"

to *precondition* equation-free computations.

With enough initialization authority:

equation free *laboratory experiments*

Computer-Aided Analysis of Nonlinear Problems in Transport Phenomena

Robert A. Brown, L. E. Scriven and William J. Silliman

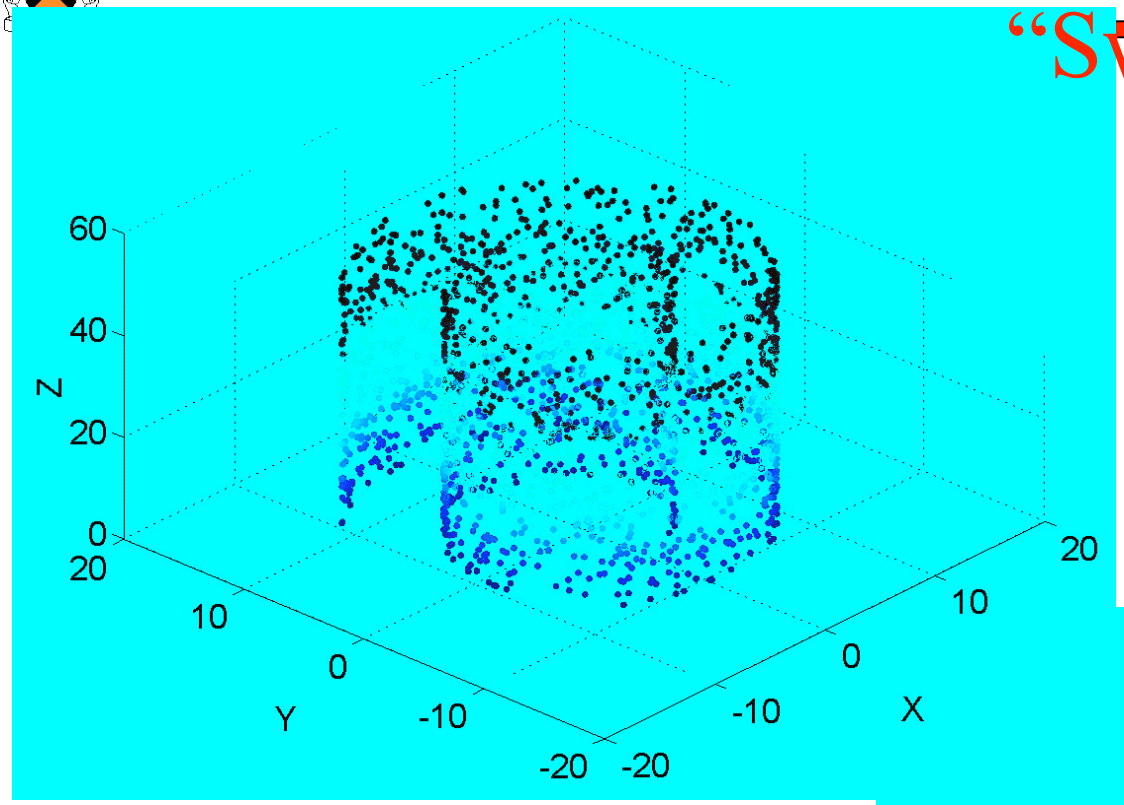
in HOLMES, P.J., New Approaches to Nonlinear Problems in Dynamics, 1980

ABSTRACT The nonlinear partial differential equations of mass, momentum, energy, Species and charge transport…. can be solved in terms of functions of limited differentiability, no more than the physics warrants, rather than the analytic functions of classical analysis… ….. basis sets consisting of low-order polynomials. …. systematically generating and analyzing solutions by fast computers employing modern matrix techniques.

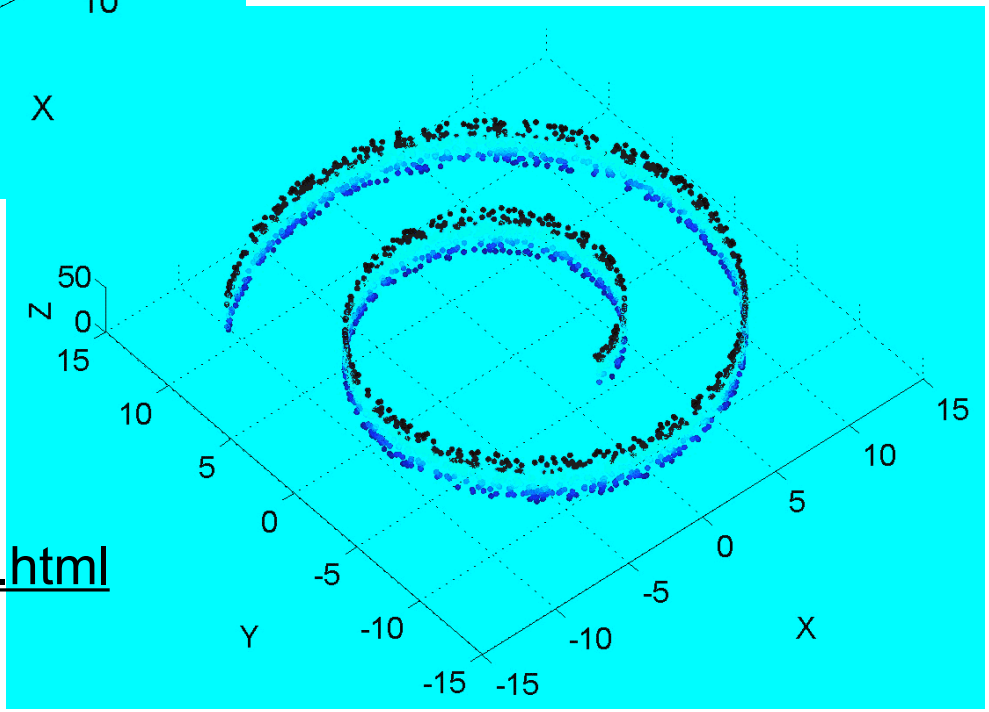
….. nonlinear algebraic equations by the Newton-Raphson method. … The Newton-Raphson technique is greatly preferred because the Jacobian of the solution is a treasure trove, not only for continuation, but also for analysing stability of solutions, for detecting bifurcations of solution families, and for computing asymptotic estimates of the effects, on any solution, of small changes in parameters, boundary conditions, and boundary shape……

In what we do, not only the analysis, but *the equations themselves* are obtained on the computer, from short experiments with an alternative, microscopic description.

"Swiss Roll" Dataset

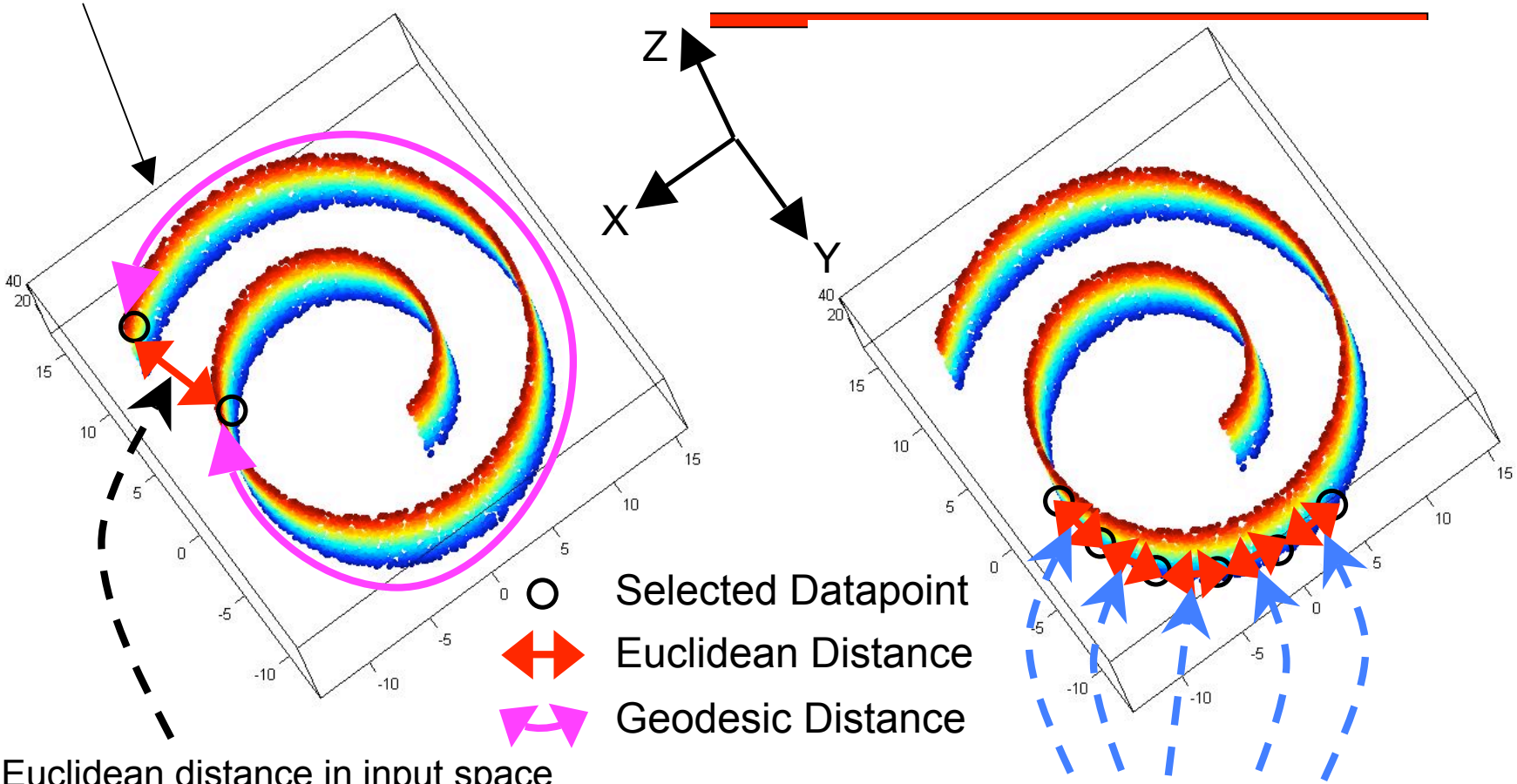


3D data sampled
from 2D manifold



Data from:
http://isomap.stanford.edu/datasets.html

D Dataset with 2D manifold



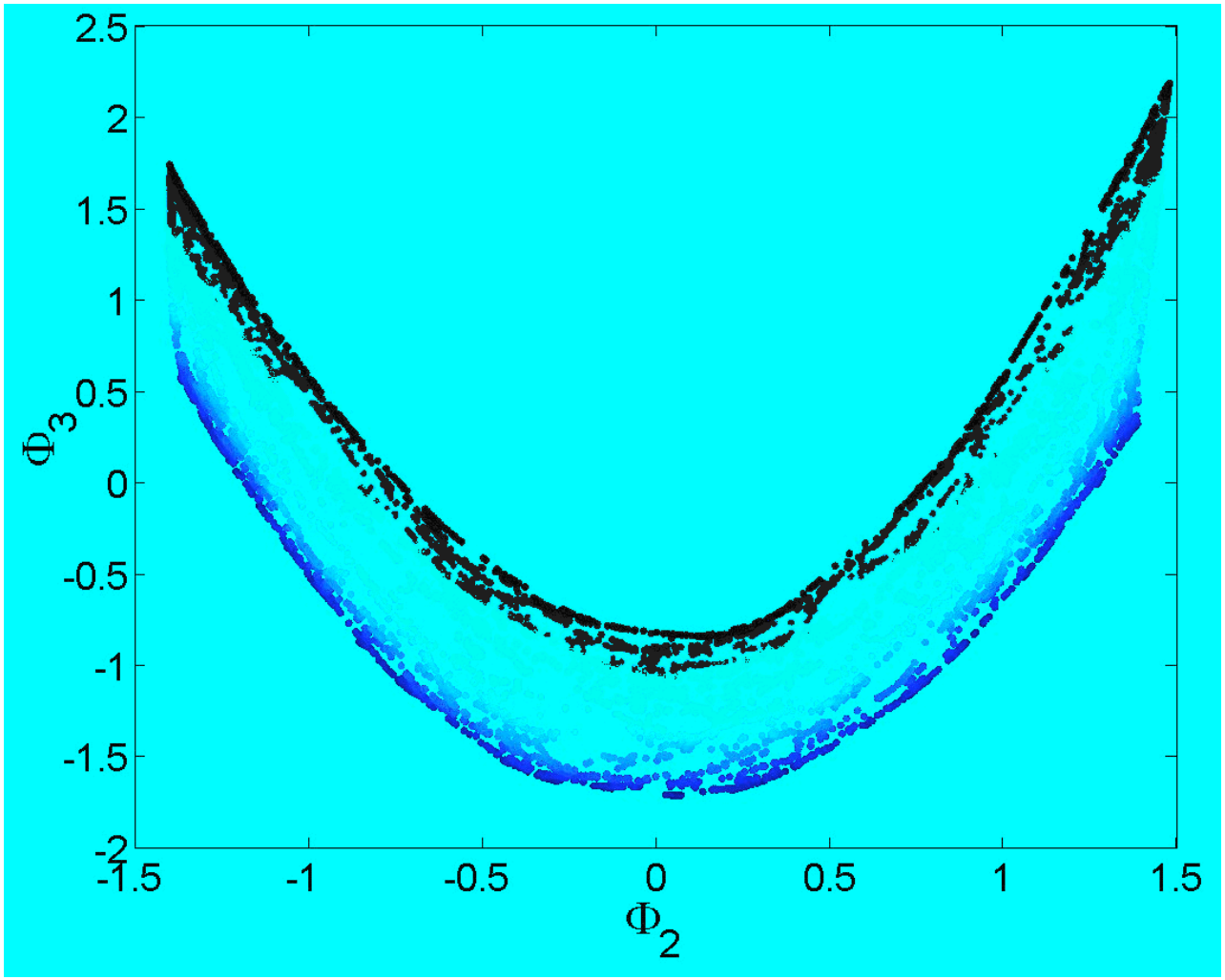
Euclidean distance in input space may be weak indicator of INTRINSIC similarity of datapoints

Euclidean distance reliably describes local neighborhood structure for similar datapoints

Geodesic distance is good for this dataset

COMBINE LOCAL DISTANCE INFORMATION TO INFER GLOBAL DATASET STRUCTURE

Diffusion Map (Φ_2, Φ_3)

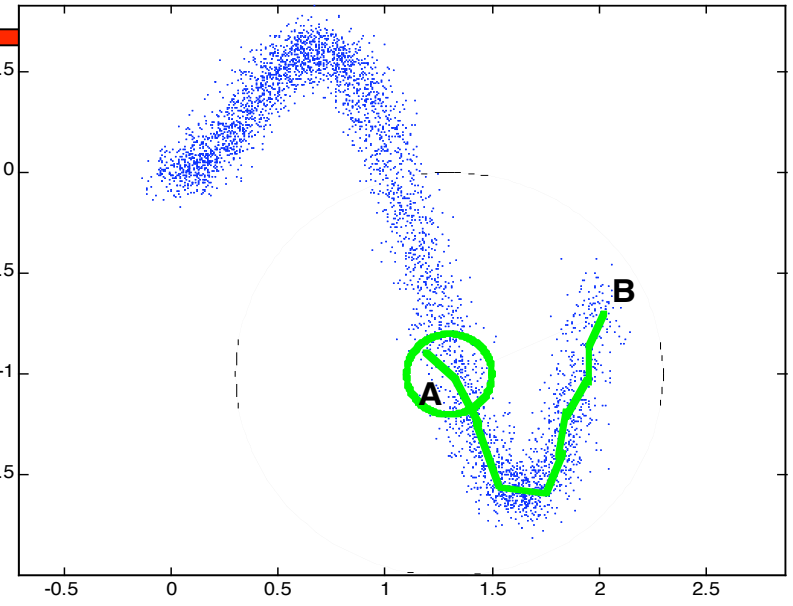


Same LOWER dimensional representation found for different “roll” rotations

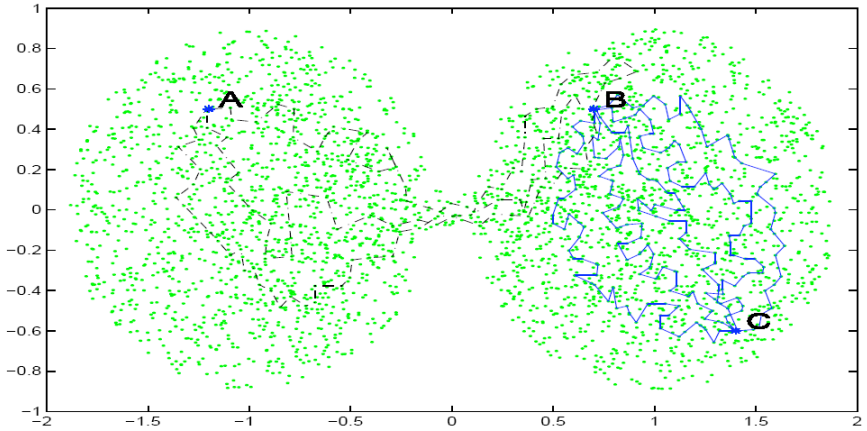
B. Nadler, S. Lafon, R.R. Coifman, & I.G. Kevrekidis, *Appl Comp Harm Anal* 2005

Parametrizing nonlinear manifolds

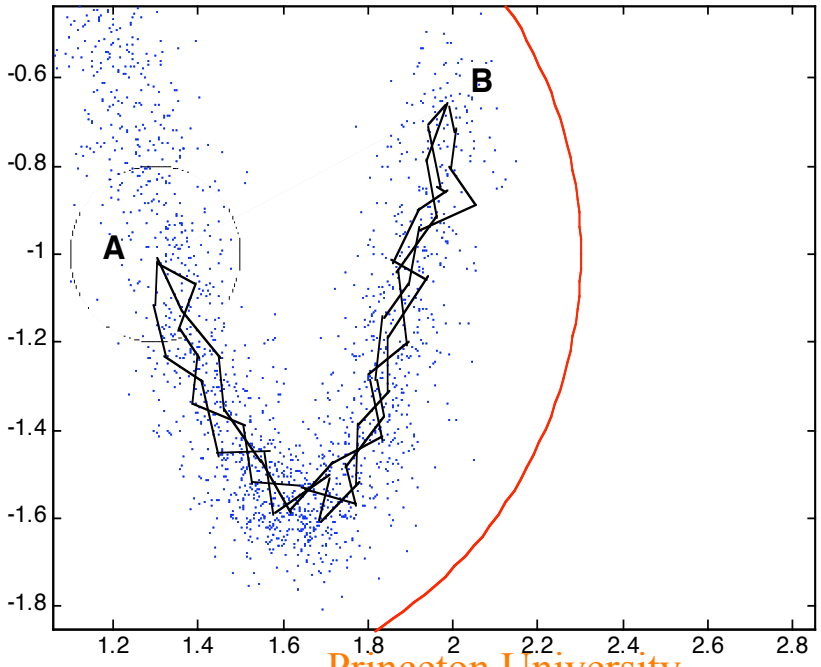
Given: (noisy) data in space, unordered, in high dimension. Need: discover meaningful parameters.
Euclidean distances between any two points usually not meaningful.
Euclidean distances between **very close** points **is** meaningful. Use local Euclidean distances and glue them to find paths between any two points.
Diffusion distance: Compute ALL paths between A and B and take a weighted average (long paths count less, but there are more of them). This is stable under noise and behaves well with bottlenecks. It is a diffusion process.



Slowly communicating states:



geod.dist(A,B)≈geod.dist(B,C), diff.dist(A,B)>>diff.dist(B,C)



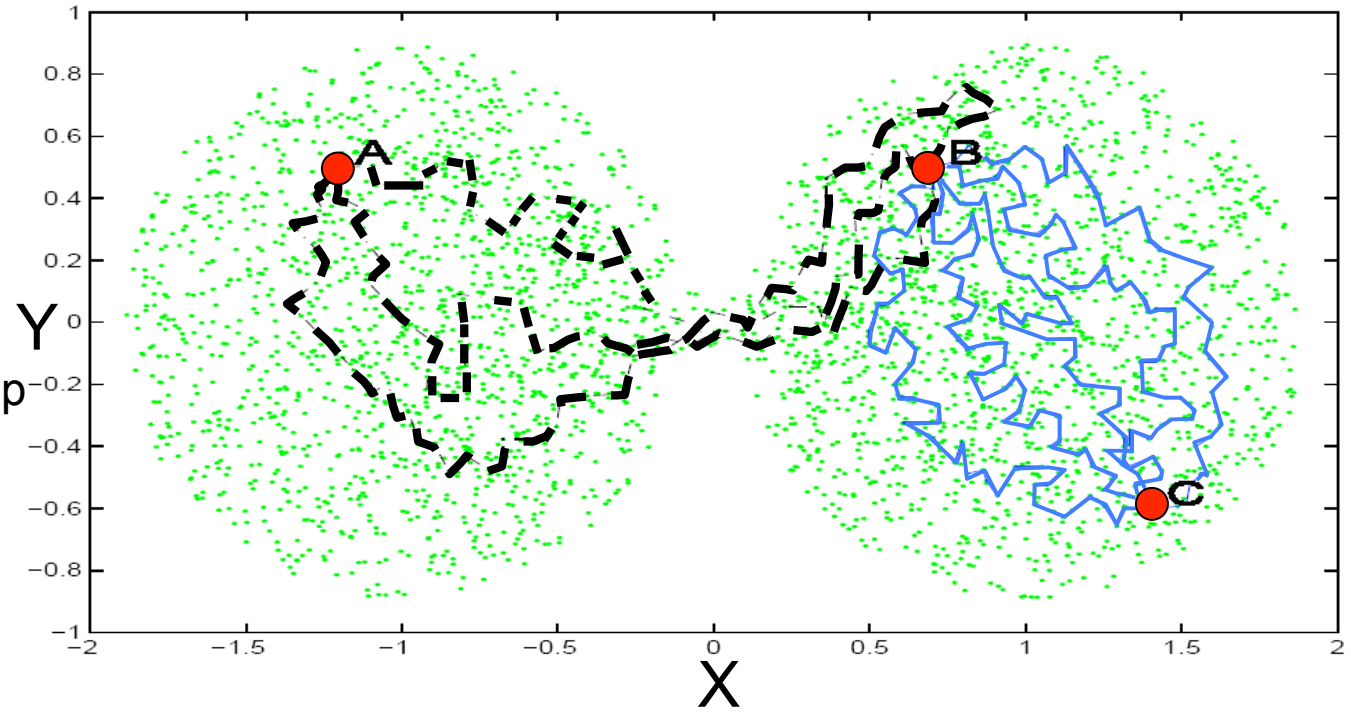
Diffusion Distance

datapoint

points A, B, C

Data set consisting of points on two disks connected by small strip

D(A,B)
Diffusion distance between states A and B



D(A,B) >> D(B,C)

Markov matrix defining diffusion given by kernel

The shortest path between points A and B is roughly the same as between B and C.

The diffusion distance however is much larger between A and B

since diffusion occurs through a bottleneck.

Much fewer paths through the data between A and B than between B and C

Graphs, Laplacian Eigefunctions, Embeddings, Heat Parametrization

We want to compute diffusion distances, and then deduce parametrization from those.

Data ⟶ Vertices of graph

Local distances ⟶ Conductivity of edges

Want to solve an heat/electrical network equation: look at Laplacian on graph, compute eigenfunctions

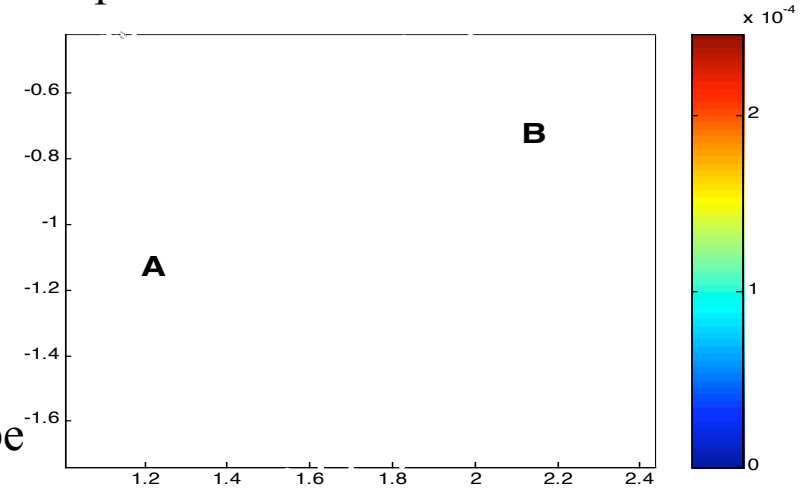
$$L\phi_i = \lambda_i \phi_i$$

The eigenfunctions (long time behaviour) can be used to compute diffusion distances:

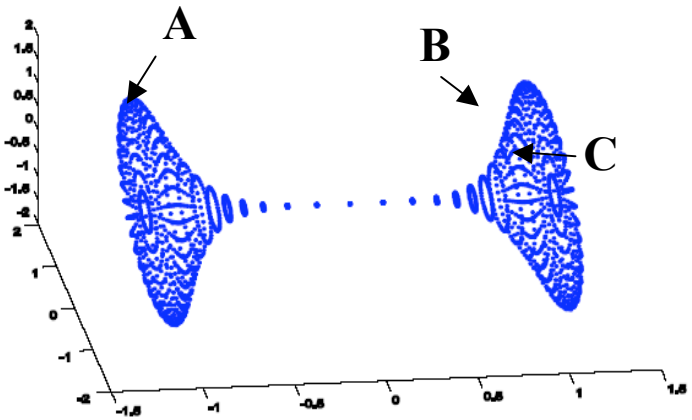
$$\sum_{j>0} \lambda^m{}_j (\phi_j(x) - \phi_j(y))^2$$

and to parametrize, mapping the set into R^n

$$E_k(x) = (\phi_2(x), \ldots, \phi_{k+1}(x))$$



Color proportional to diffusion distance, related to how much heat flows between A and B in a certain time



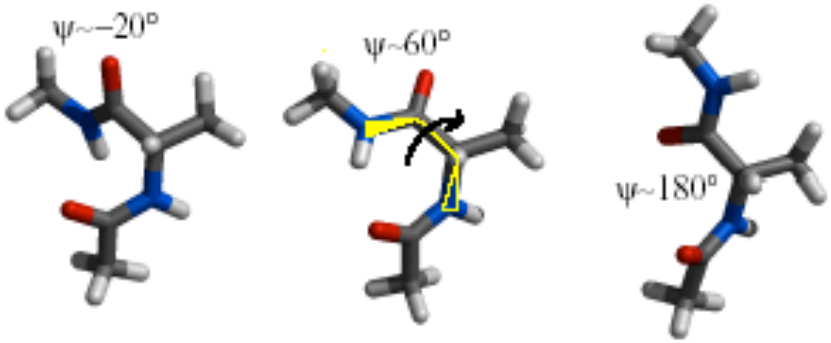
Slowly communicating states are mapped by the eigefunctions into far away points.

** **Belkin & Nyogi, Lafon, Coifman, Maggioni**

Computations:

- Order n, nlog(n) via eigenfunctions
- Order nlog(n) via diffusion wavelets (full multiscale organization)

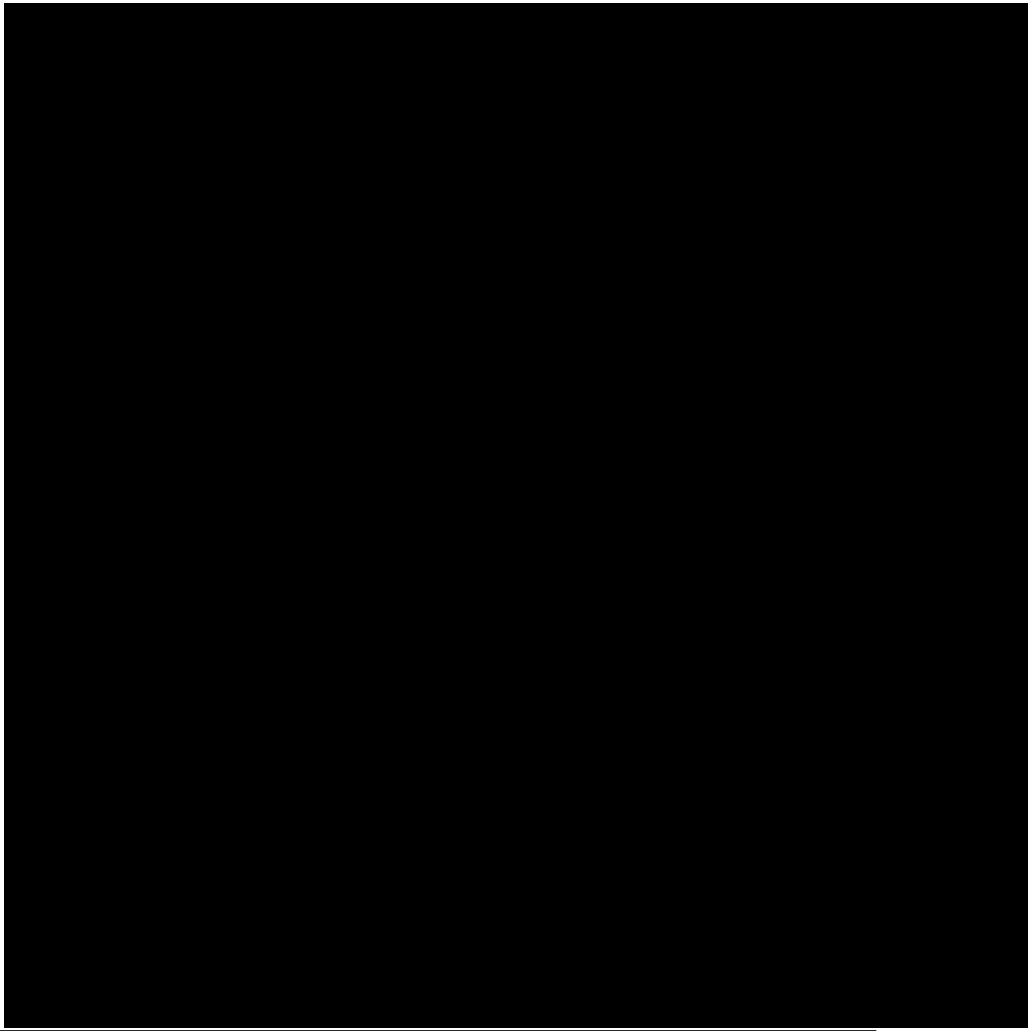
All depend on decay of eigenvalues, i.e. whether there is *separation of time scales*



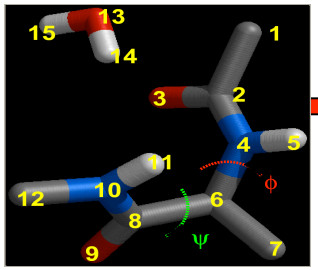
w/ Gerhard Hummer, NIDDK / *J.Chem.Phys. 03*

Alanine Dipeptide
In 700 tip3p waters

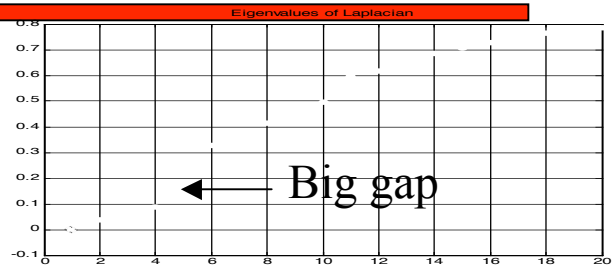
The waters
The dipeptide
and the Ramachandran plot



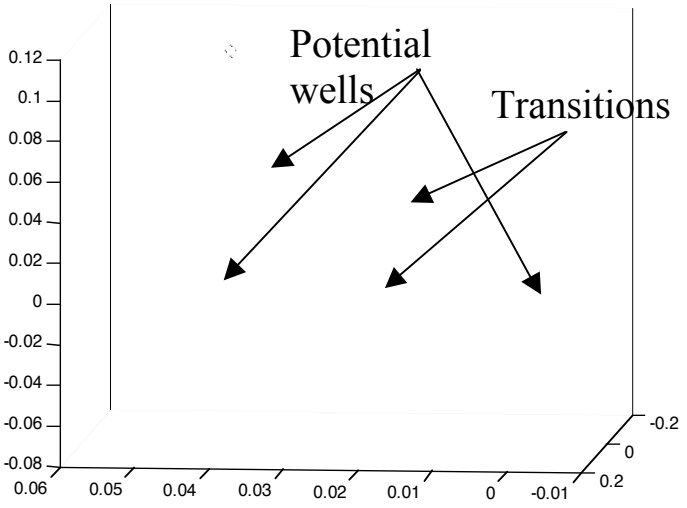
Application to Alanine Dipeptide data



12-atom dipeptide fragment in water. Coordinates of the atoms in the molecule and of the closest water molecule.

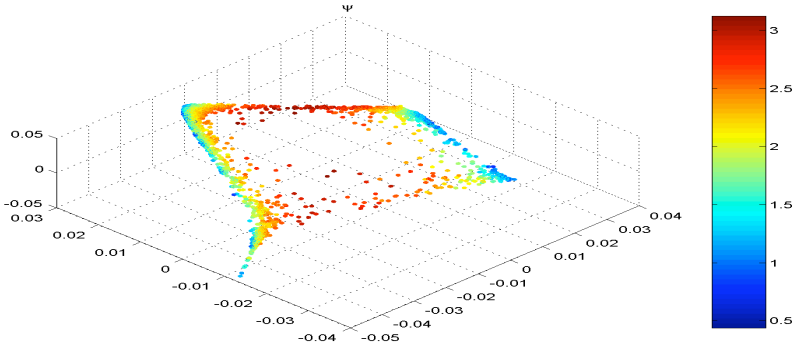


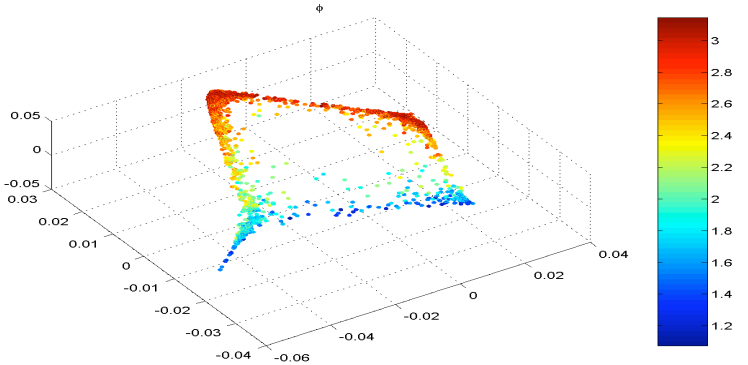
Chemists parametrize with two angles which are dynamically significant. We would like to **LEARN** these good parameters just by looking at the high-dimensional clouod of points resulting from the simulation.



Analysis of data (with no prior knowledge about the problem) reveals:

- configuration space is one dimensional,
- one parameter (arc length on states) is enough,
- the two commonly used angles are rather good parameters
- we find two good parameters, in part related to the angles, that do parametrize



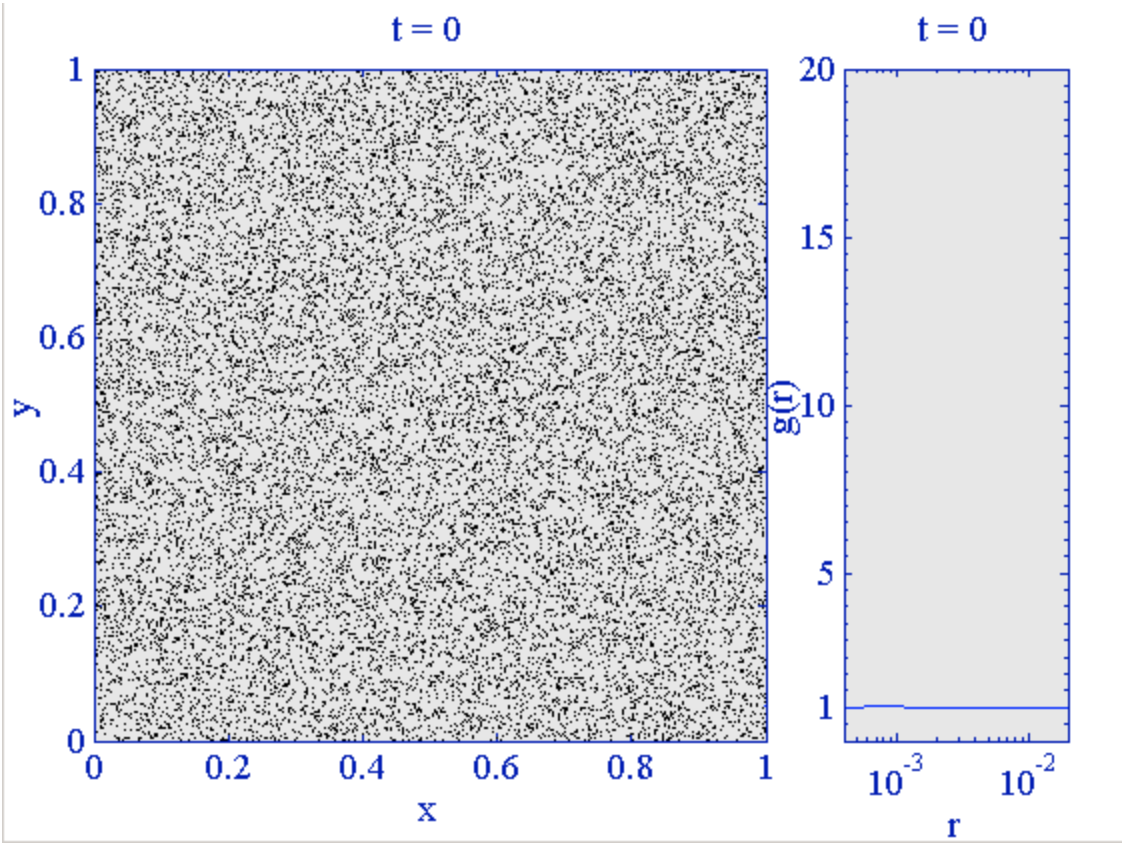


Color represents values of the chemist's angles on the set of states of the molecule, our parameters give the axes (2 are enough), as well as the natural "arc-length" parametrization.

Clustering and stirring in a plankton model

Young, Roberts and Stuhne, *Nature* 2001

Dynamics of System with convection



Simulation Method

- Random (equal) birth and death, probability: $\lambda = \mu$.
- Brownian motion. $x'_k = x_k + \delta x_k(t); \left\langle \delta^2 x_k \right\rangle = 2\tau D$
- Advective stirring. (φ, θ are random phases)

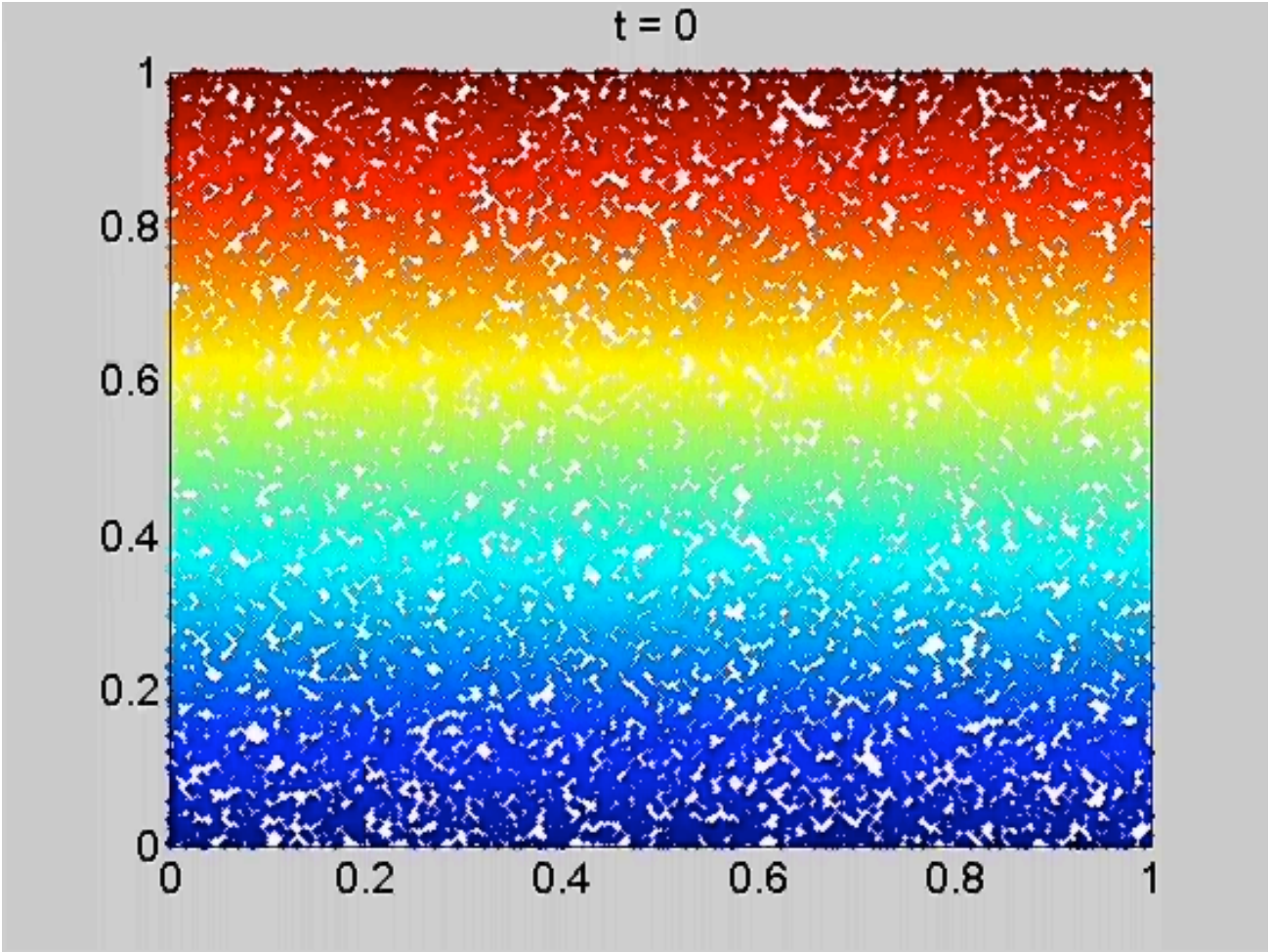
$$x_k(t+\tau) = x'_k(t) + U\tau/2 \cos[k y'_k(t) + \varphi(t)]$$

$$y_k(t+\tau) = y'_k(t) + U\tau/2 \cos[k x'_k(t) + \theta(t)]$$

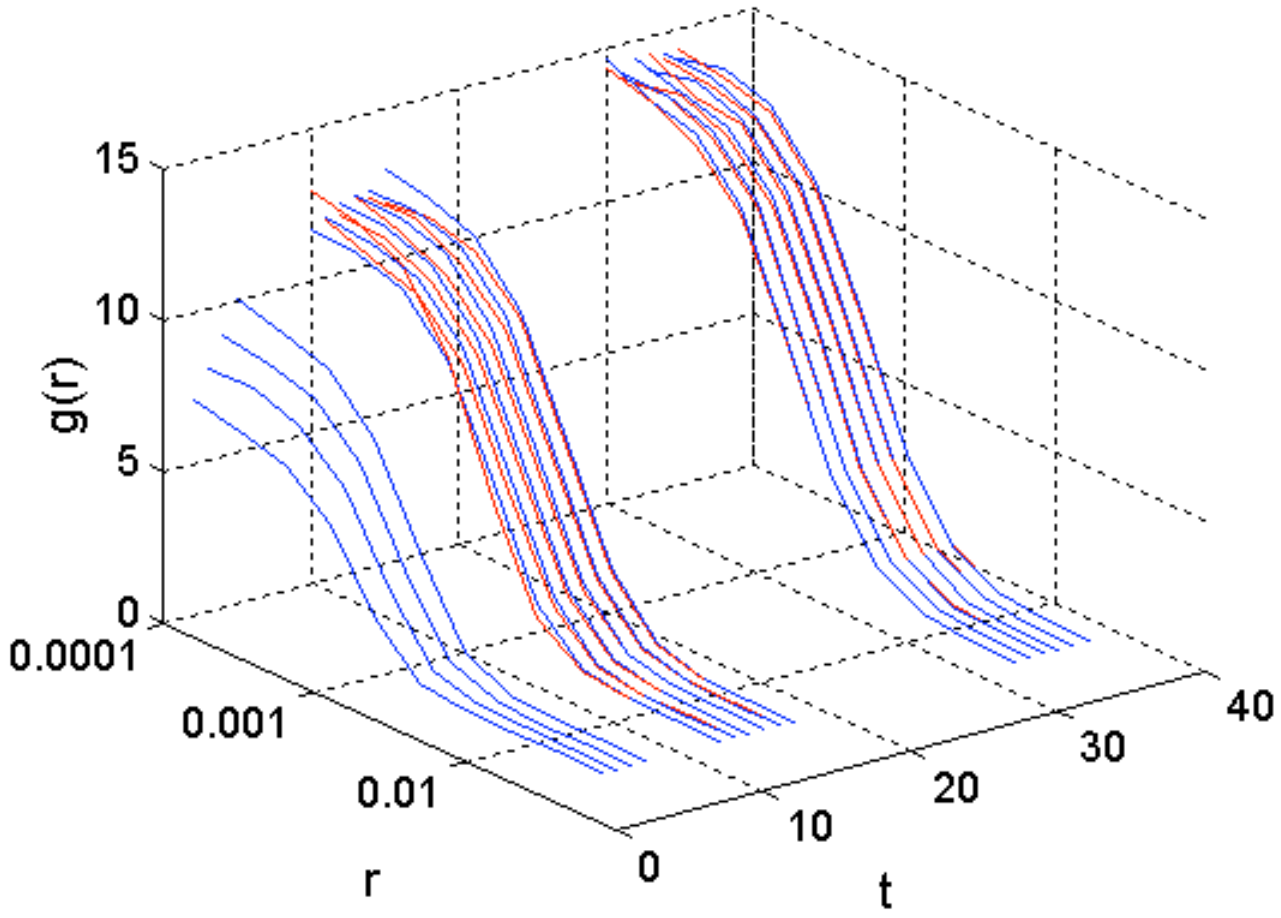
- IC: 20000 particles randomly placed in 1*1 box
- Analytical Equation for G(r):

$$G_t = 2D\frac{1}{r}(rG_r)_r + 2(\lambda - \mu)G + \gamma\frac{1}{r}(r^3 G_r)_r + 2\lambda C\delta(\mathbf{r})$$

Stirring by a random field (color = y)



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



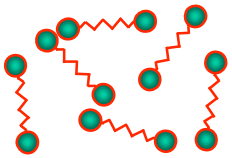
Coarse Brownian Dynamics for Nematic Liquid Crystals: Bifurcation, Control, Coarse Projective Integration

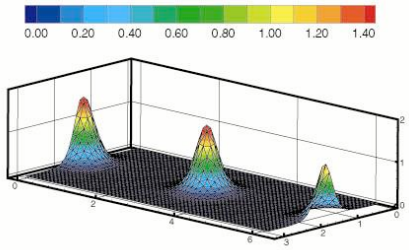
Costas Siettos, Mike Graham, IGK, arXiv.org 02, J.Chem.Phys. 03

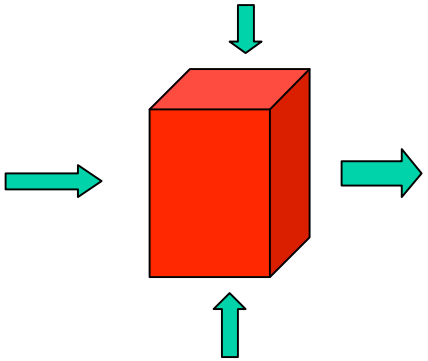
Micro-scale

Meso-scale

Macroscale







Microscopic/ stochastic models

Fokker-Planck

Moment Closures

Brownian Dynamics

Monte Carlo

Spherical Harmonics

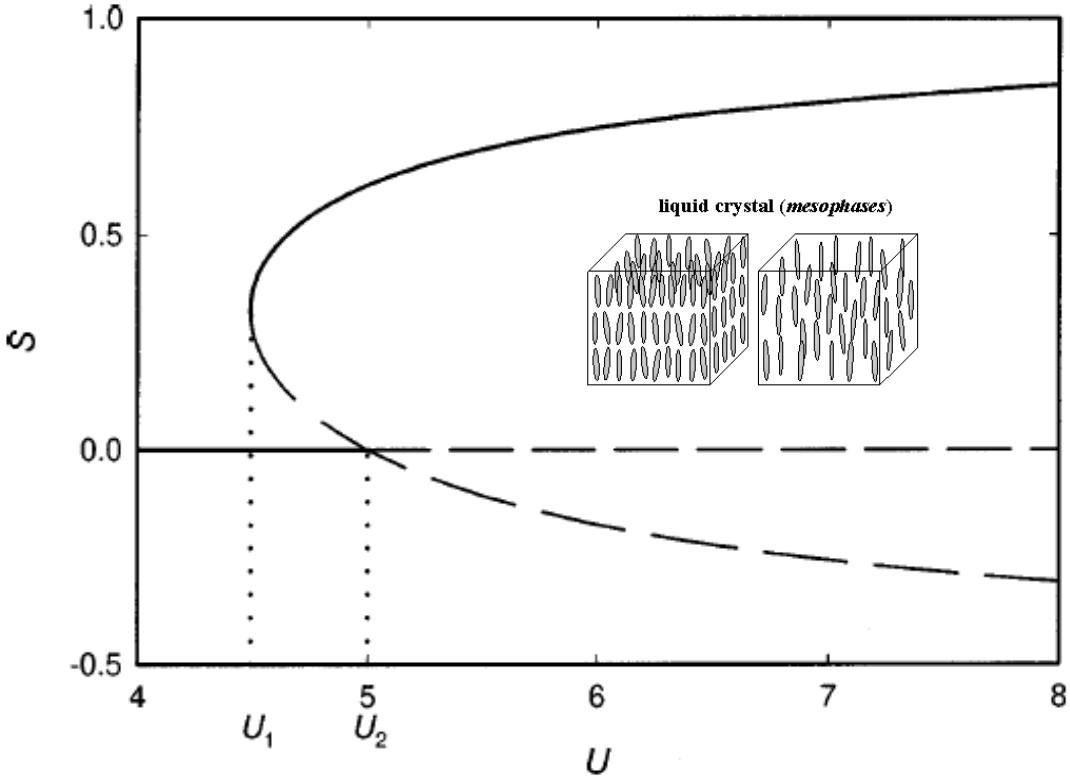
Wavelets

PDE's

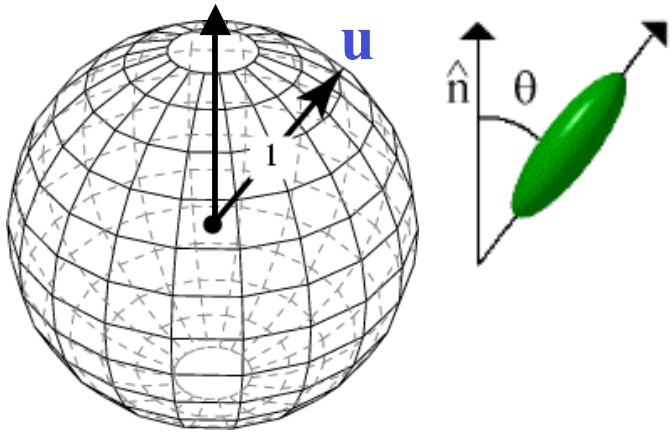
Fokker-Planck of the orientation probability density

$$\frac{\partial \psi}{\partial t} = D\frac{\partial}{\partial \mathbf{u}}\cdot\left[\frac{\partial \psi}{\partial \mathbf{u}} + \frac{\psi}{k_B T}\frac{\partial V(\mathbf{u})}{\partial \mathbf{u}}\right] - \frac{\partial}{\partial \mathbf{u}}\cdot(\dot{\mathbf{u}}\psi).$$

(M. Doi, J. Polym. Sci., Polym. Phys., 1981).



Representation of the orientation in space:



The orientation distribution function $\psi(t,\mathbf{u})$ gives the probability density that a rod is oriented along **u** at time t.

The scalar order parameter **S** represents a scalar measure of the degree of order of the sample, nondimensional potential intensity ***U***.

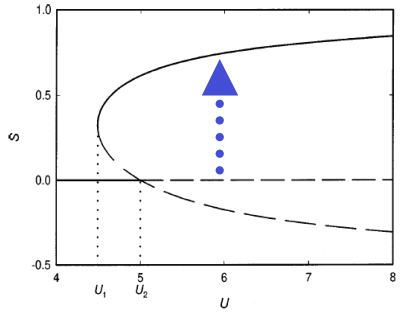
Brownian dynamics: Evolution of the orientation distribution

The evolution of the distribution function is described by the following Stochastic integro-differential equation:

R. G. Larson and H.C. Öttinger, Macromolecules, **24**, 6270 (1991).

$$d\mathbf{u} = (\mathbf{I} - \mathbf{uu}) \cdot \left(- \frac{D}{kT} \frac{\partial V}{\partial \mathbf{u}} dt + \sqrt{2D} d\mathbf{w} \right)$$

, dw: Wiener process with covariance IΔt



u unit vector representing the rigid rod orientation

k Boltzmann constant

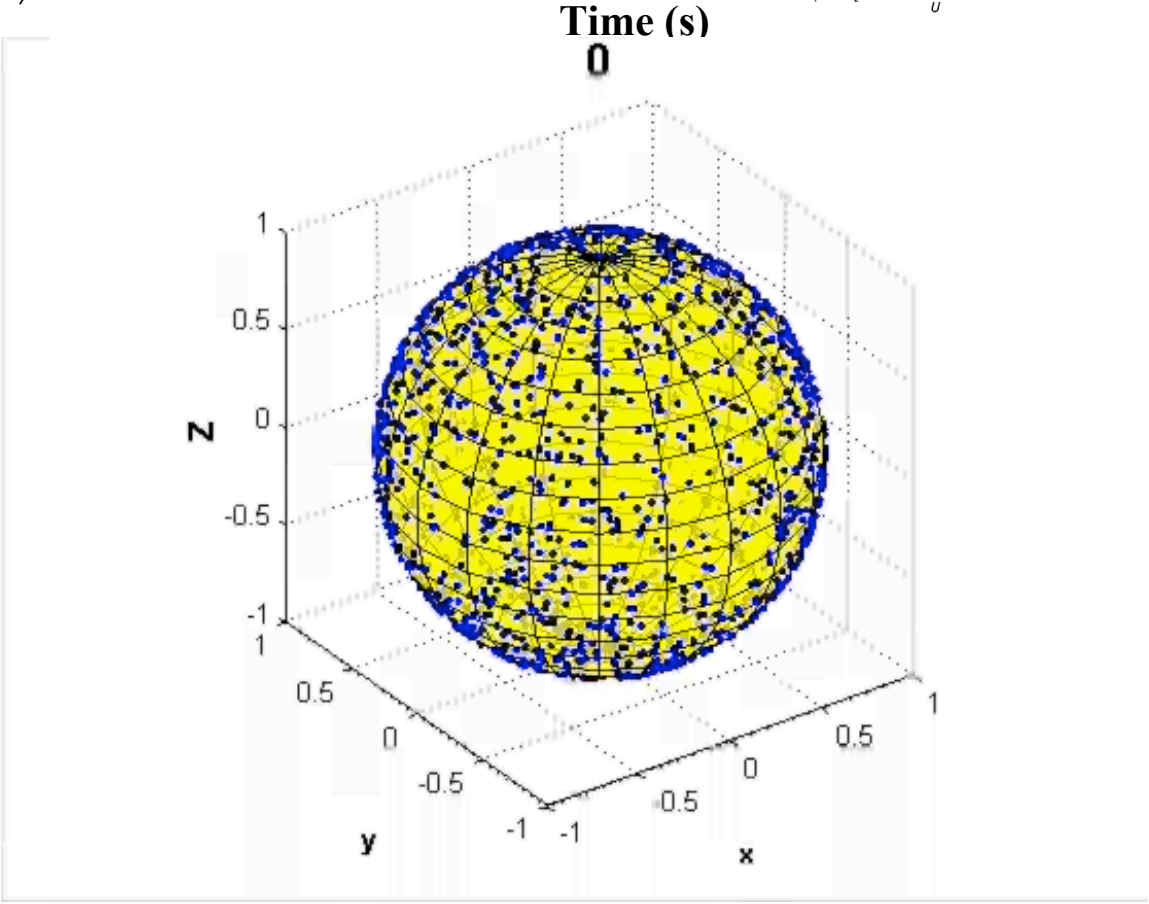
T absolute temperature

D is the rotational diffusivity

V is a nematic potential, a functional of the distribution function

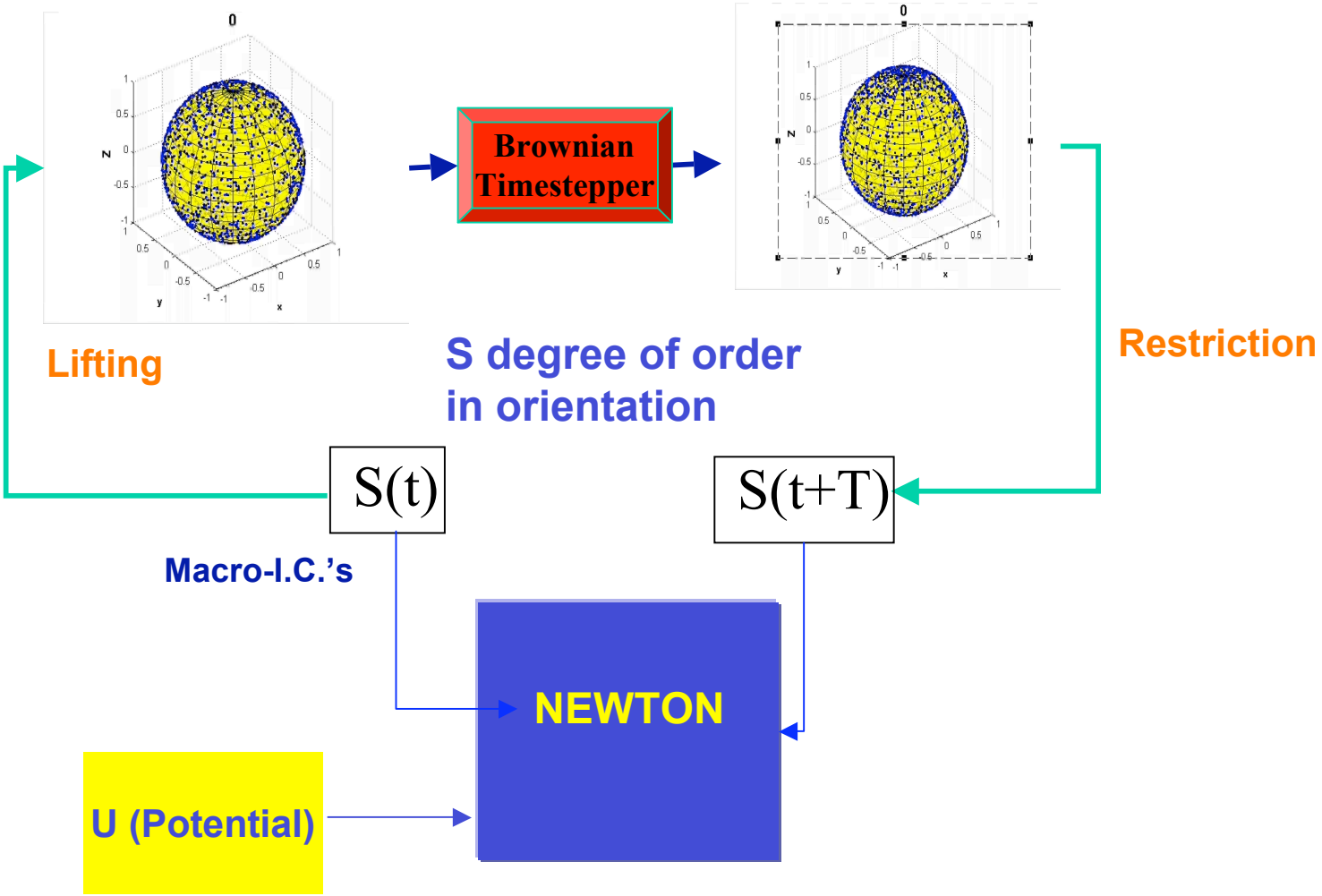
Explict Euler method to time-integrate

$$\mathbf{u}_i(t+\Delta t) = \frac{\mathbf{u}_i(t) - \frac{D}{kT} \frac{\partial V}{\partial \mathbf{u}}\Big|_t \Delta t + \sqrt{2D\Delta}\,\mathbf{w}_i}{\left\| \mathbf{u}_i(t) - \frac{D}{kT} \frac{\partial V}{\partial \mathbf{u}}\Big|_t \Delta t + \sqrt{2D\Delta}\,\mathbf{w}_i \right\|}$$



The microscopic model: Brownian dynamics

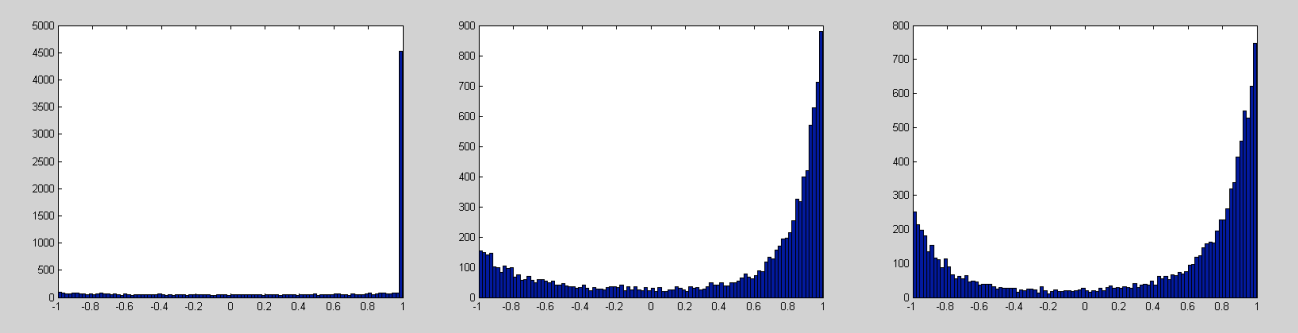
The microscopic model is considered as a “black-box” coarse timestepper



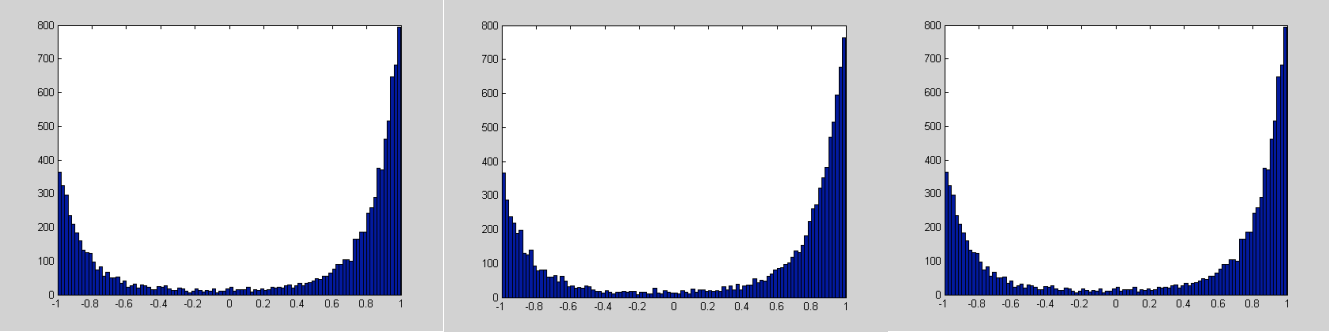
EVOLUTION OF DISTRIBUTION IN TIME

(for 10^3 particles, U=5., dt=0.005,uuzzdes0=0.8)

0.0001 **0.1** **0.3**



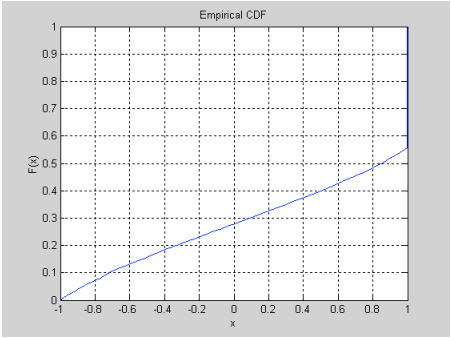
0.5 **0.75** **1.0**



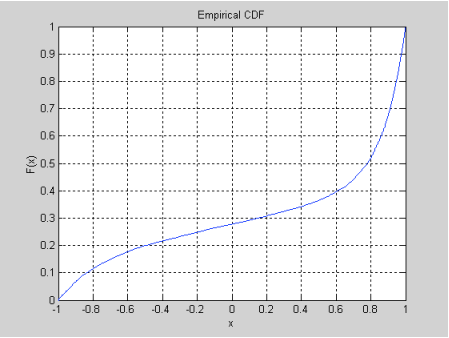
EVOLUTION OF CUMULATIVE DISTRIBUTION IN TIME

(for 10^3 particles, U=5., dt=0.005,uuzzdes0=0.8)

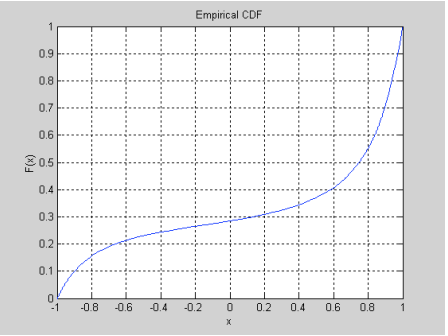
0.0001



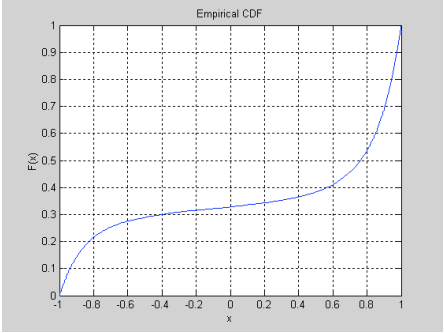
0.1



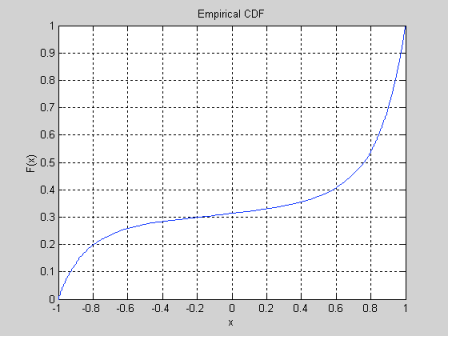
0.3



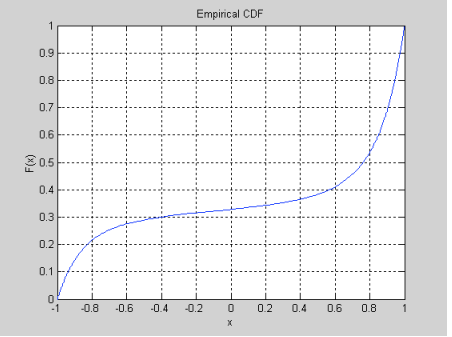
0.5



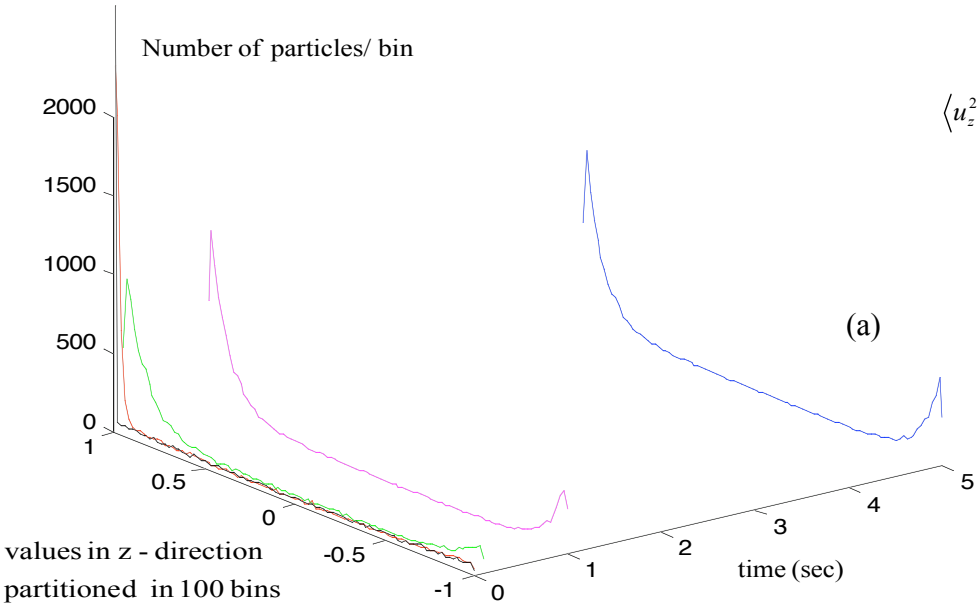
0.75

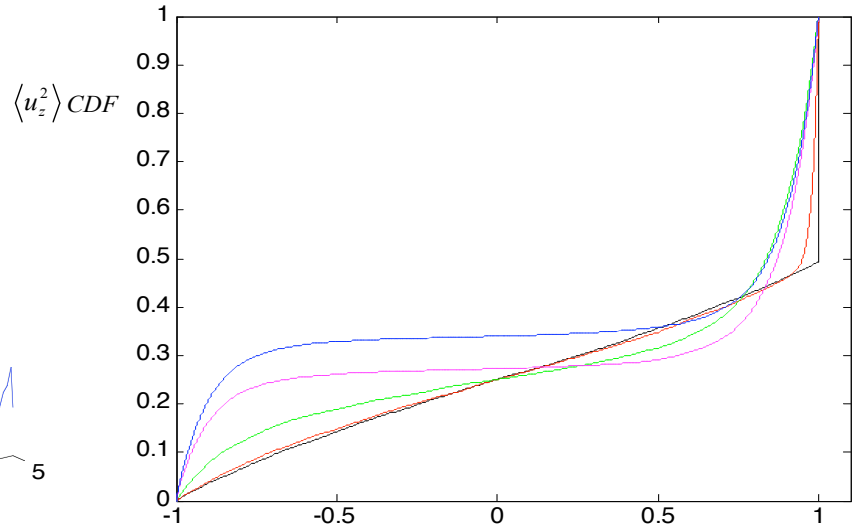


1.0



SLAVING OF THE HIGHER MOMENTS TO THE SLOWER ONES





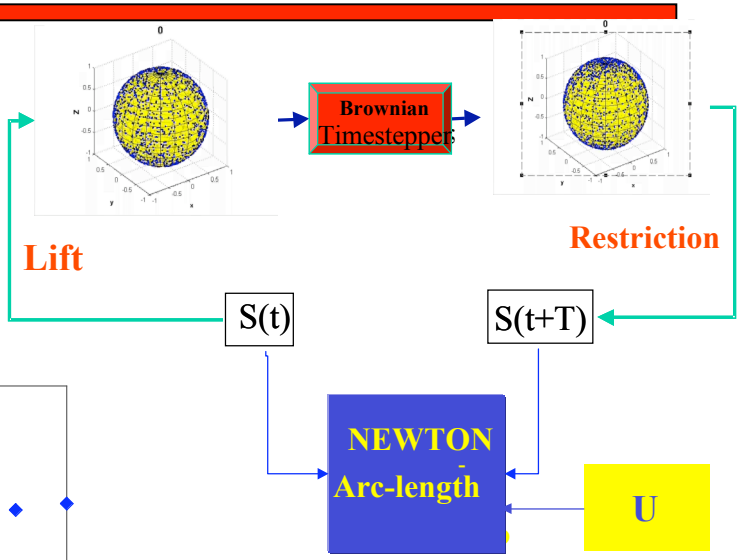
(a) Evolution of the distribution function of **u** (histogram) in the z – direction. The values in the z- direction were partitioned in 100 bins.

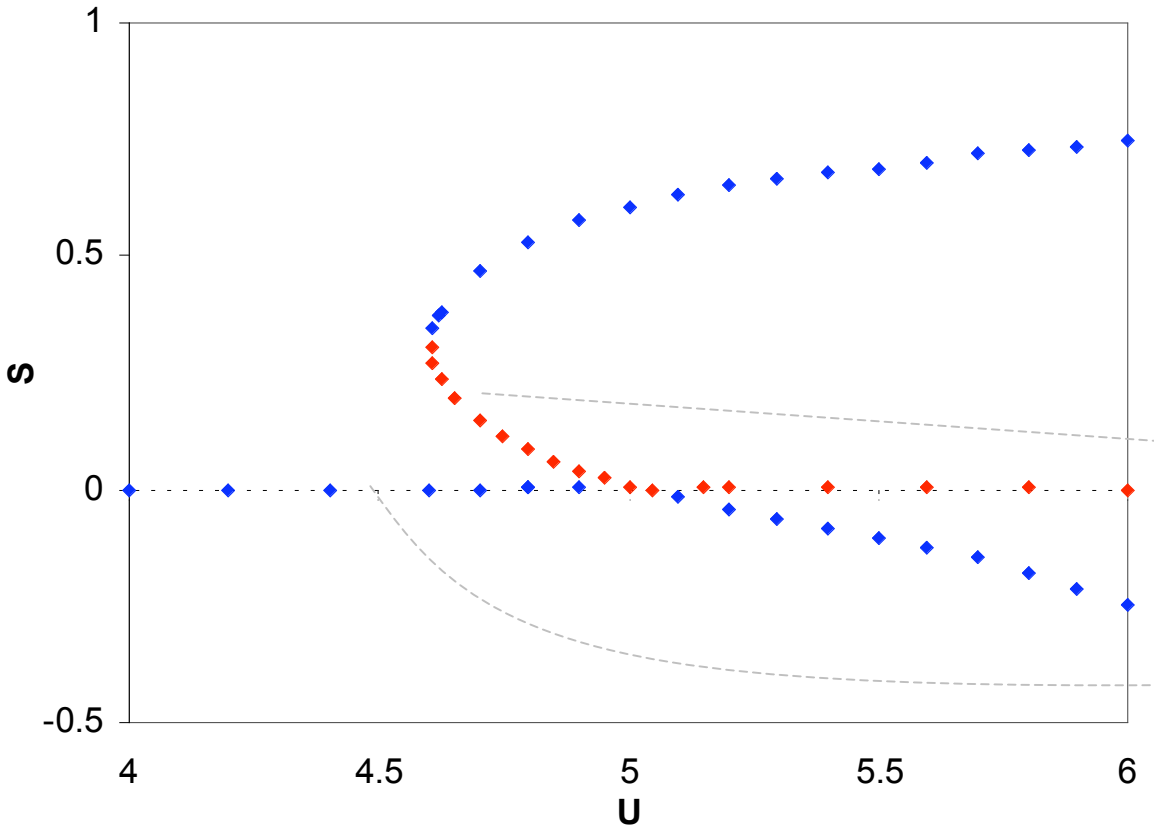
(b) Evolution of the corresponding cumulative distribution function of **u** in the z – direction. The simulations were performed at U= 5.5, $N_{traj}=10^3$, dt=0.001

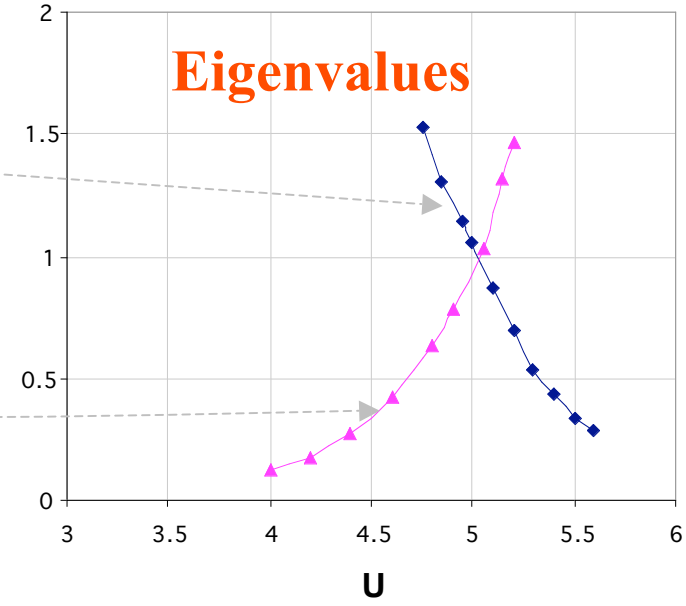
Coarse Bifurcation Diagram

•$3x10^5$ molecules,

Arc-length Continuation wrapped around the coarse timestepper; timestep: T=1.7)







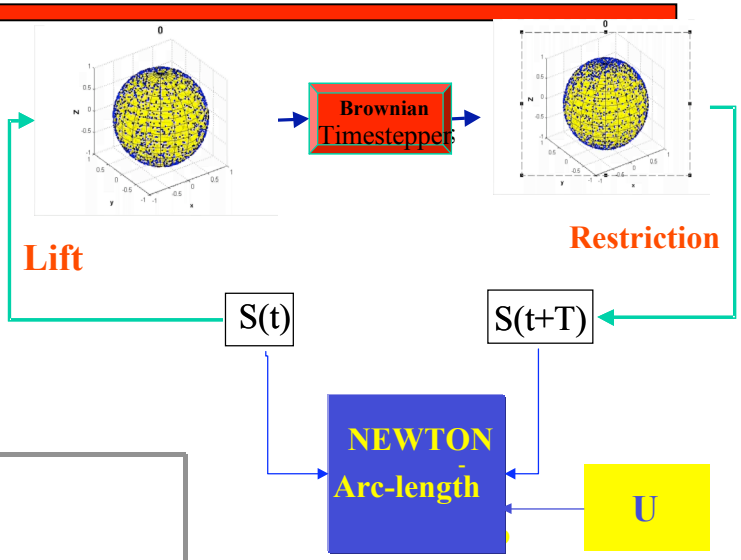
IS ONE COARSE VARIABLE SUFFICIENT ? 2-moment lifting

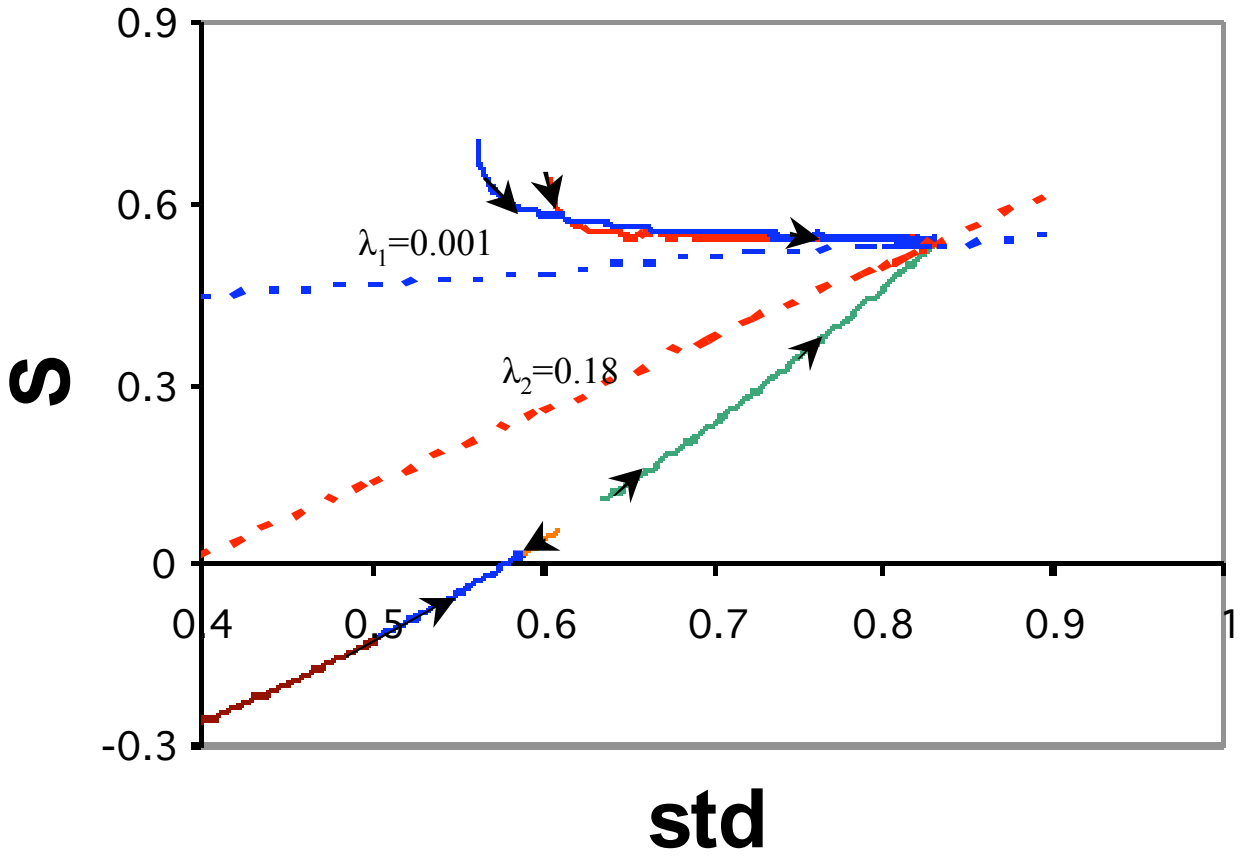
Adaptive computation:

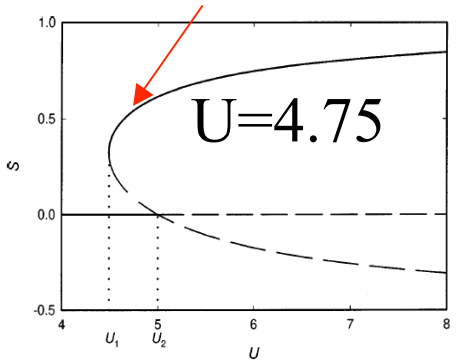
use smaller time steps, estimate error

use more mesh points, estimate error

HERE: use more lifting variables





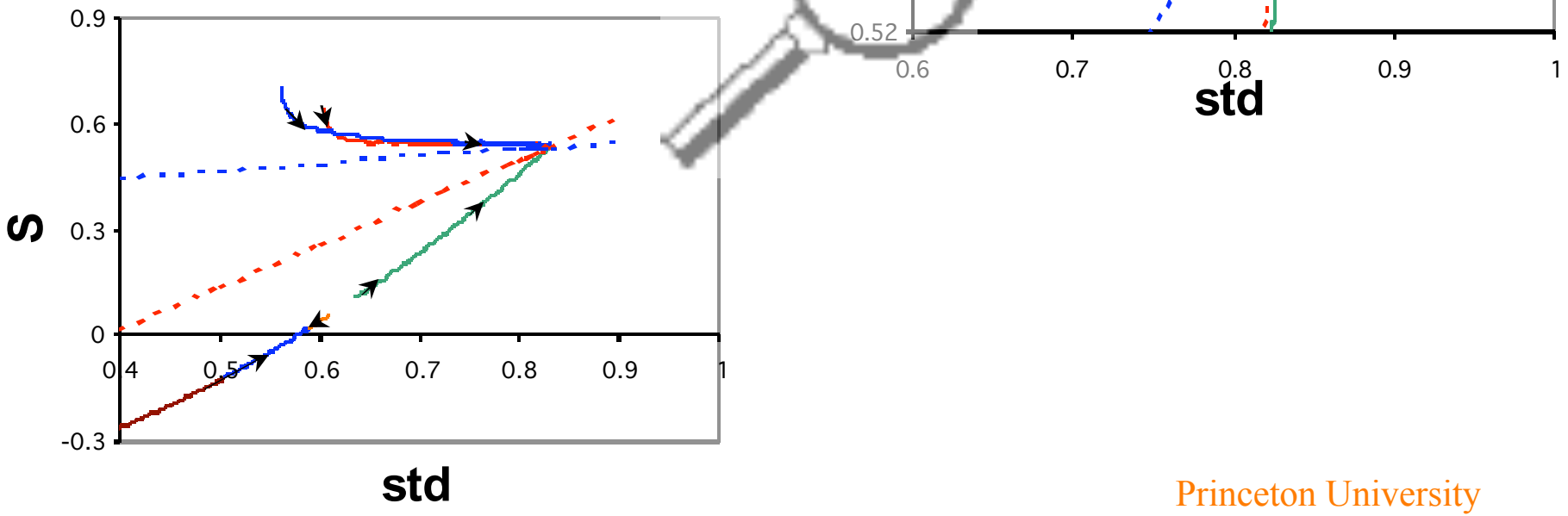


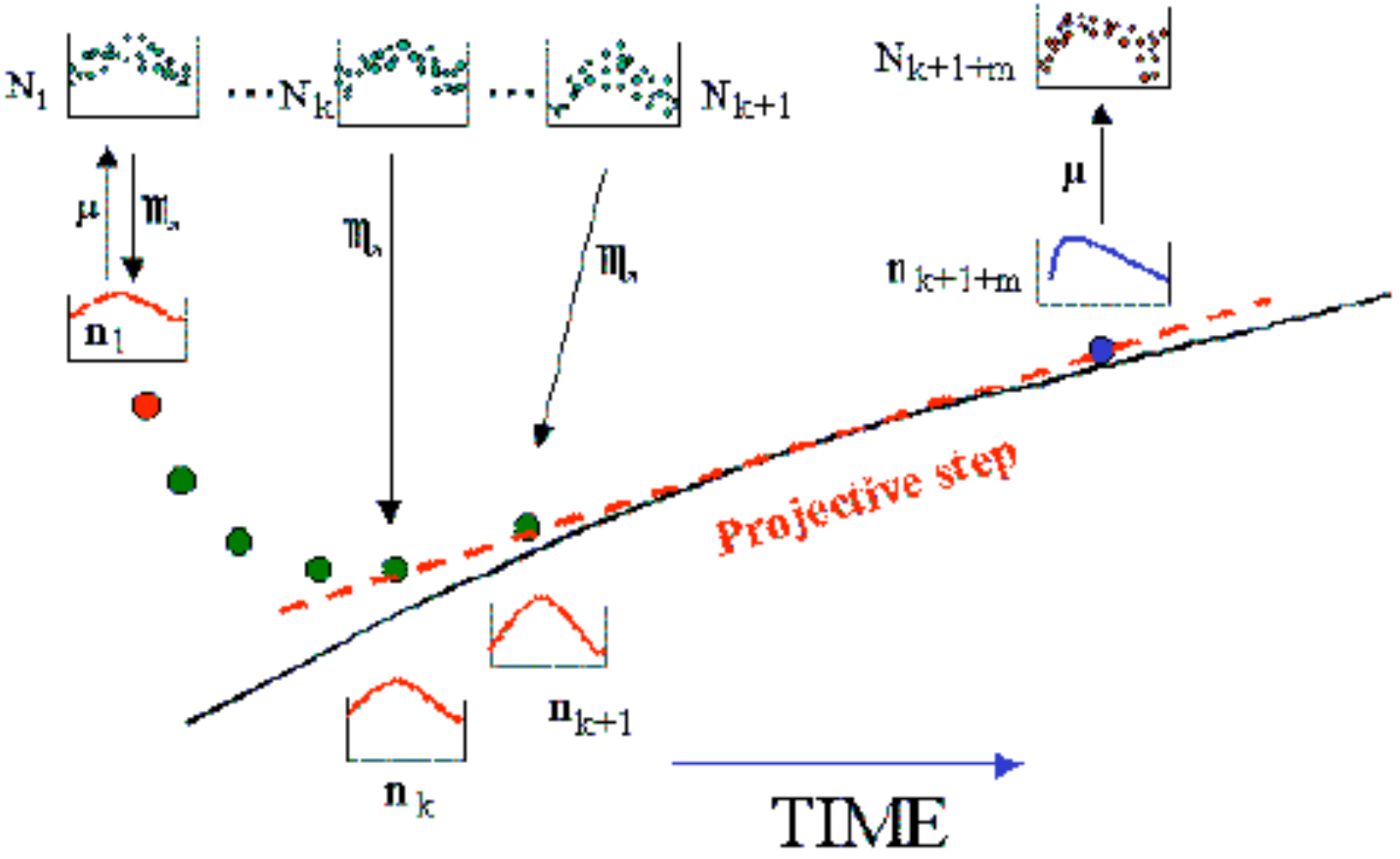
Slaving of higher moments to lower ones

One-dimensional slow manifold

std is slaved very fast to S

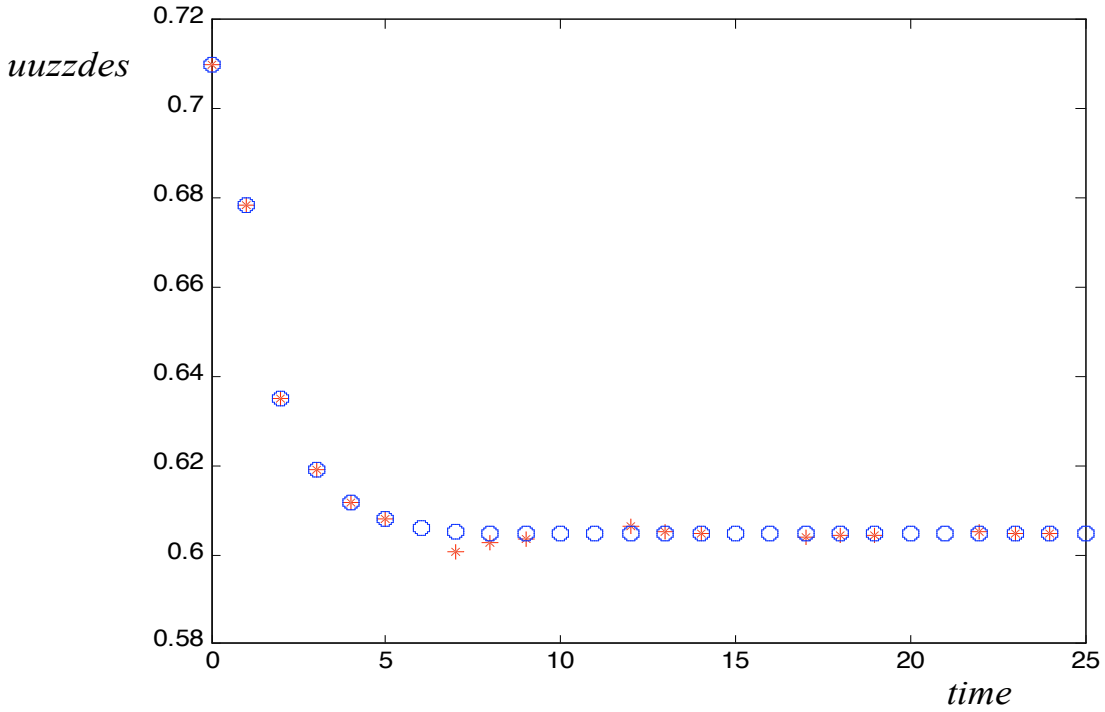
Description with just one moment (the coarse variable S)




N1
Nk
Nk+1
Nk+1+m
μ
μ
n1
nk+1+m
Projective step
nk+1
nk
TIME

COARSE PROJECTIVE INTEGRATION

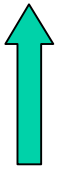
(TimeHorizon =1.0 time units, Ntraj=10^5, dt=0.0005)



Coarse Control

BROWNIAN DYNAMICS TIMESTEPPER

DO "COARSE" CONTROL

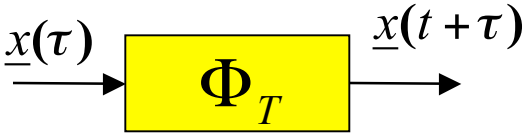


Separation principle

DESIGN DISCRETE COARSE OBSERVERS

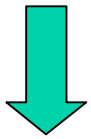
DESIGN DISCRETE COARSE CONTROLLERS

•Compile



•Fixed point: $\underline{x} - \Phi(\underline{x}) = 0$

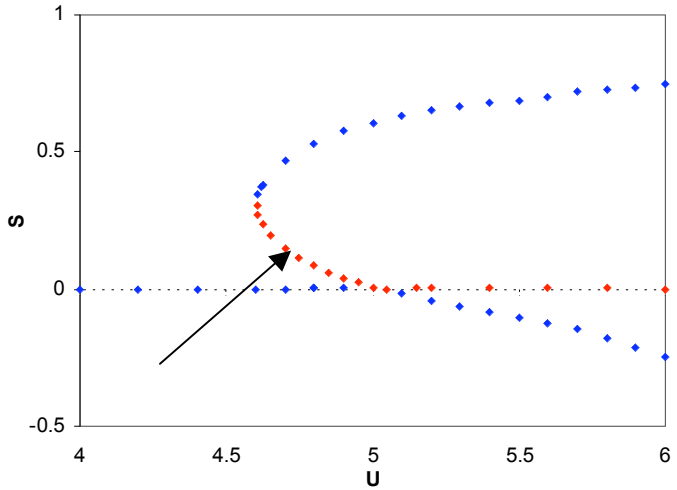
Action of Slow Jacobian

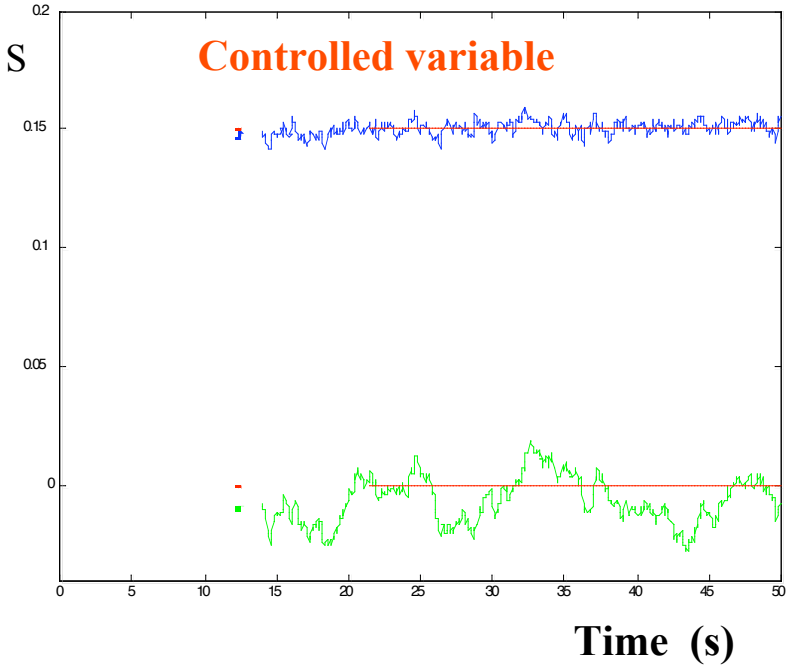


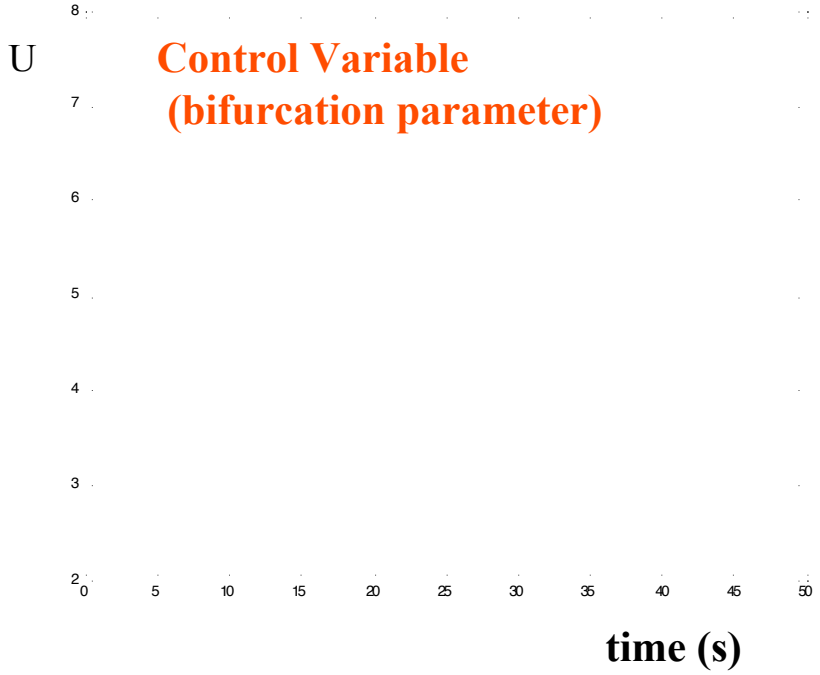
Perform "coarse" system identification

The Control Problem

Stabilization of an open-loop unstable coarse steady state (for U_0=4.7)







Lattice-gas model of the CO oxidation reaction on a square lattice taking into account the lateral CO-CO interactions:

1) $CO_{gas} + (*)_i \rightarrow CO_{ads,i}$ - CO adsorption

2) $O_{2,gas} + (*)_i + (*)_j \rightarrow O_{ads,i} + O_{ads,j}$ - O_2 adsorption

3) $CO_{ads,i} \rightarrow CO_{gas} + (*)_i$ - CO_{ads} desorption

4) $CO_{ads,i} + O_{ads,j} \rightarrow (*)_i + (*)_j + CO_{2,gas}$ - CO_2 formation

5) $CO_{ads,i} + (*)_j \rightarrow (*)_i + CO_{ads,j}$ - CO_{ads} migration

6) $O_{ads,i} + (*)_j \rightarrow (*)_i + O_{ads,j}$ - O_{ads} migration

The "exact" time evolution of the reaction system is described by the chemical *master equation*:

$$d P_\alpha / dt = \sum_\beta (W_{\beta\rightarrow\alpha} P_\beta - W_{\alpha\rightarrow\beta} P_\alpha),$$

where P_α (P_β) is the probability of finding the system in configuration α (β). $W_{\alpha\rightarrow\beta}$ are transition probabilities per unit time for various reactions on a lattice. In general, the master equation cannot be solved exactly, therefore, one has to use some uncontrolled approximations in order to derive the macroscopic evolution equations, or to apply the ***kinetic Monte Carlo*** (KMC) simulations which can provide, at least in principle, the exact solution of the problem.

Routine closures: Mean-field approximation

For the zeroth moments (concentrations) on the lattice

$$d\theta_{CO}/dt = \alpha\theta_* - \gamma\theta_{CO}S_1 - 4k_r\theta_{CO}\theta_O S_2$$

$$d\theta_O/dt = 4\beta\theta_*^2 - 4k_r\theta_{CO}\theta_O S_2$$

where

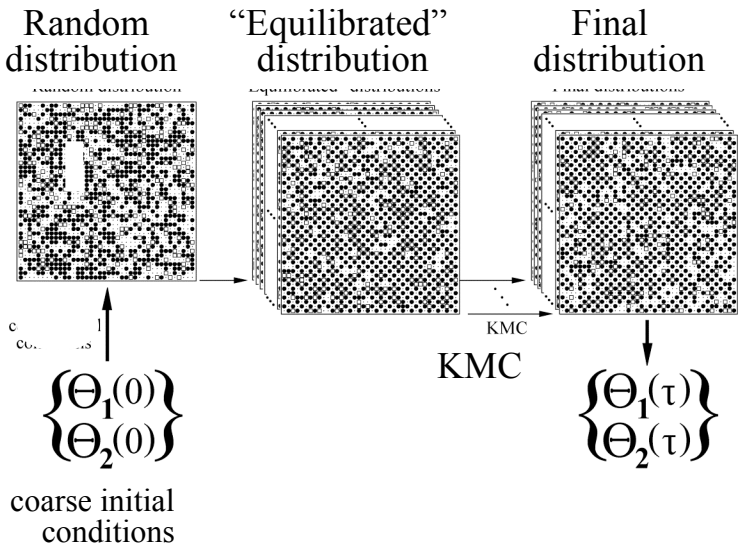
$\theta_* = 1 - \theta_{CO} - \theta_O$;

$S_1 = exp(4\theta_{CO}\varepsilon_{CO\text{-}CO}/(RT))$; $S_2 = exp(3\theta_{CO}\varepsilon_{CO\text{-}CO}/(RT))$;

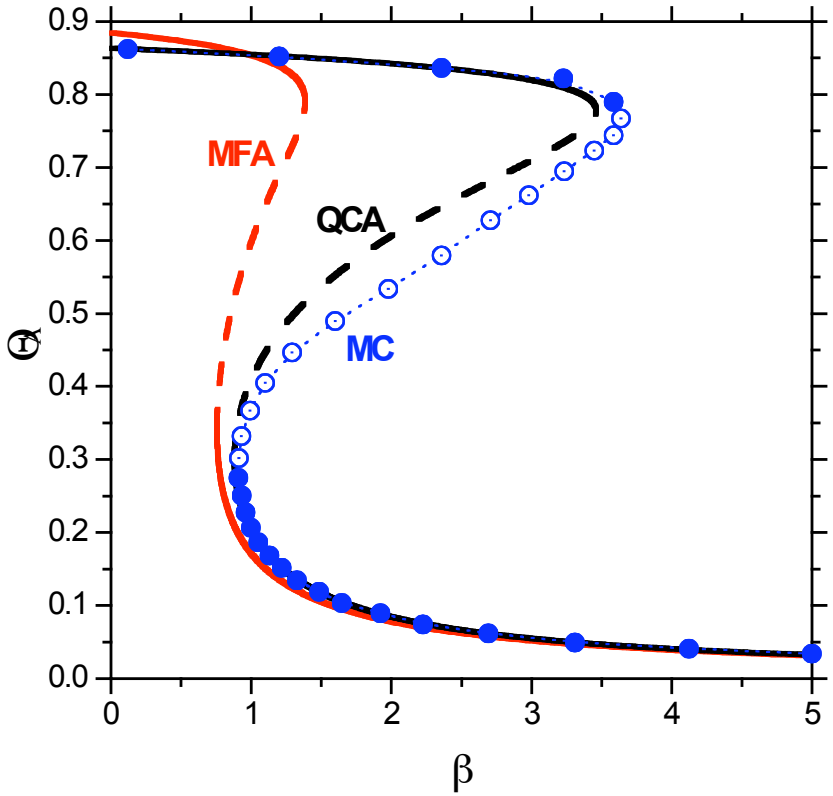
$\varepsilon_{CO\text{-}CO}$ – the energetic parameter of lateral interactions;

T – temperature, R – gas constant; α, γ, k_r – rate constants.

Coarse Timestepper



Weak interactions ($\varepsilon_{co\text{-}co}$ = 1.5 kcal/mol)

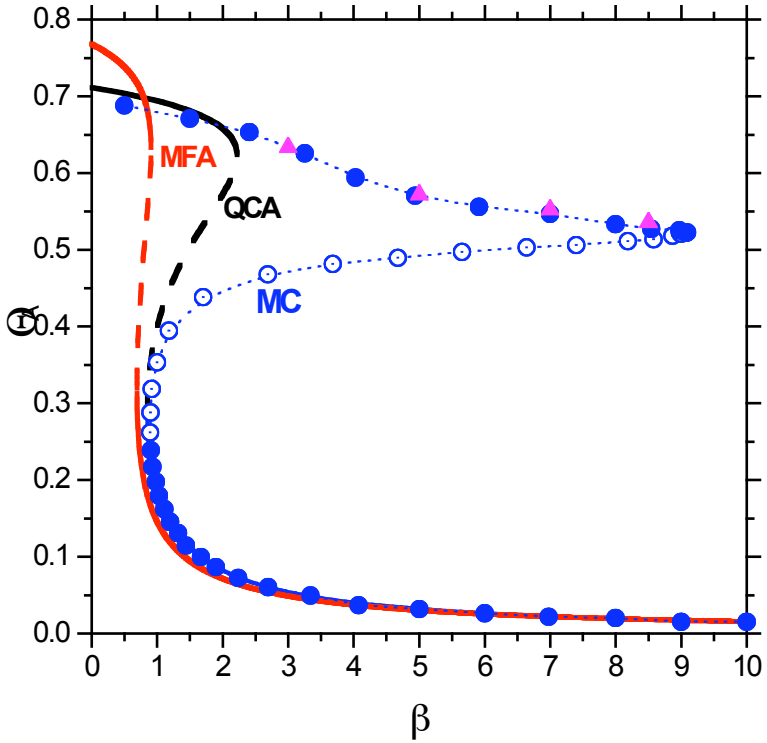


MFA - mean-field approximation

QCA - quasi-chemical approximation

MC - our Monte Carlo + Time-stepper algorithm

Strong interactions ($\varepsilon_{CO\text{-}CO}$ = 2.0 kcal/mol)



MFA - mean-field approximation

QCA - quasi-chemical approximation

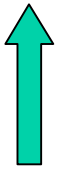
MC - our Monte Carlo + Time-stepper algorithm (500*500 lattice, N_{run} = 400)

Due to repulsive CO-CO interactions, a second order phase transition occurs at Θ_{CO} > 0.25 ML. As a result, an ordered overlayer is formed on surface.

Coarse Control

BROWNIAN DYNAMICS TIMESTEPPER

DO "COARSE" CONTROL

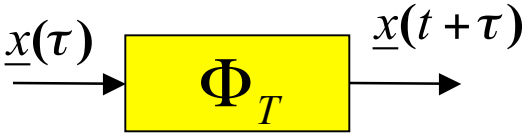


Separation principle

DESIGN DISCRETE COARSE OBSERVERS

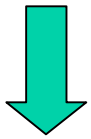
DESIGN DISCRETE COARSE CONTROLLERS

•Compile



•Fixed point: $\underline{x} - \Phi(\underline{x}) = 0$

Action of Slow Jacobian



Perform "coarse" system identification

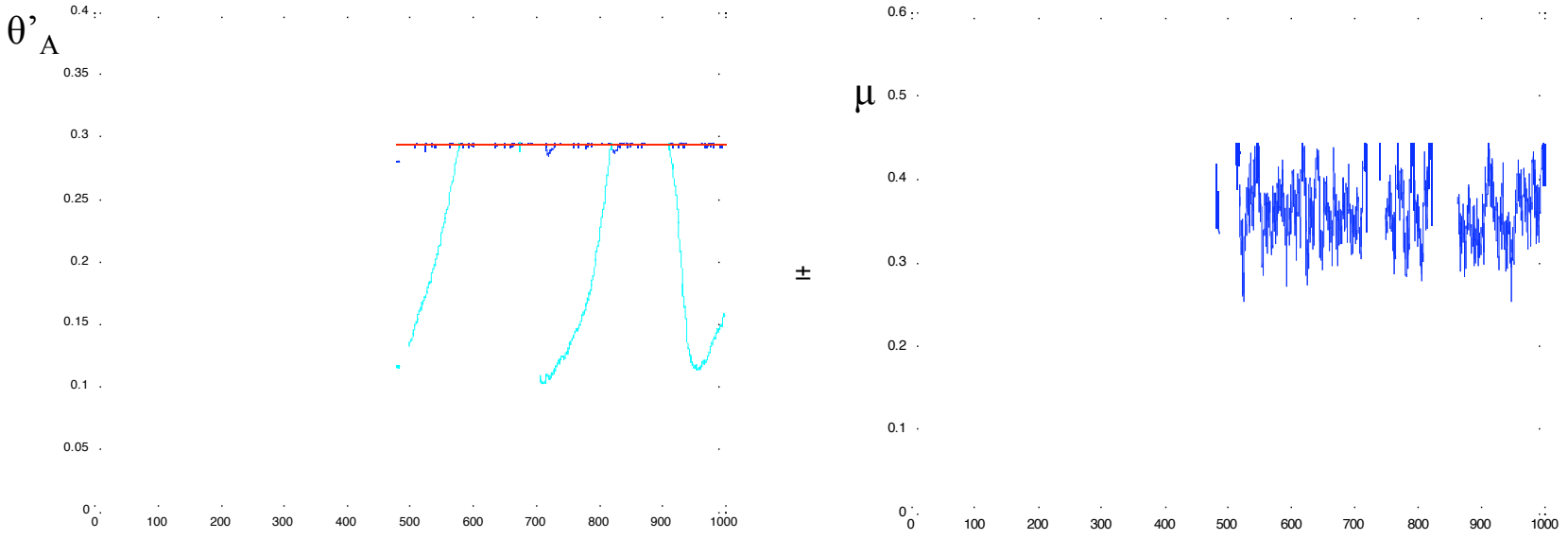
COARSE CONTROL ON A MONTE-CARLO MODEL: Simulation Results

Control objective: Stabilizing the macroscopic (expected) unstable steady state at 20.7

This coarse steady state (as well the coarse Jacobian and control matrix) was estimated through the coarse timestepper)

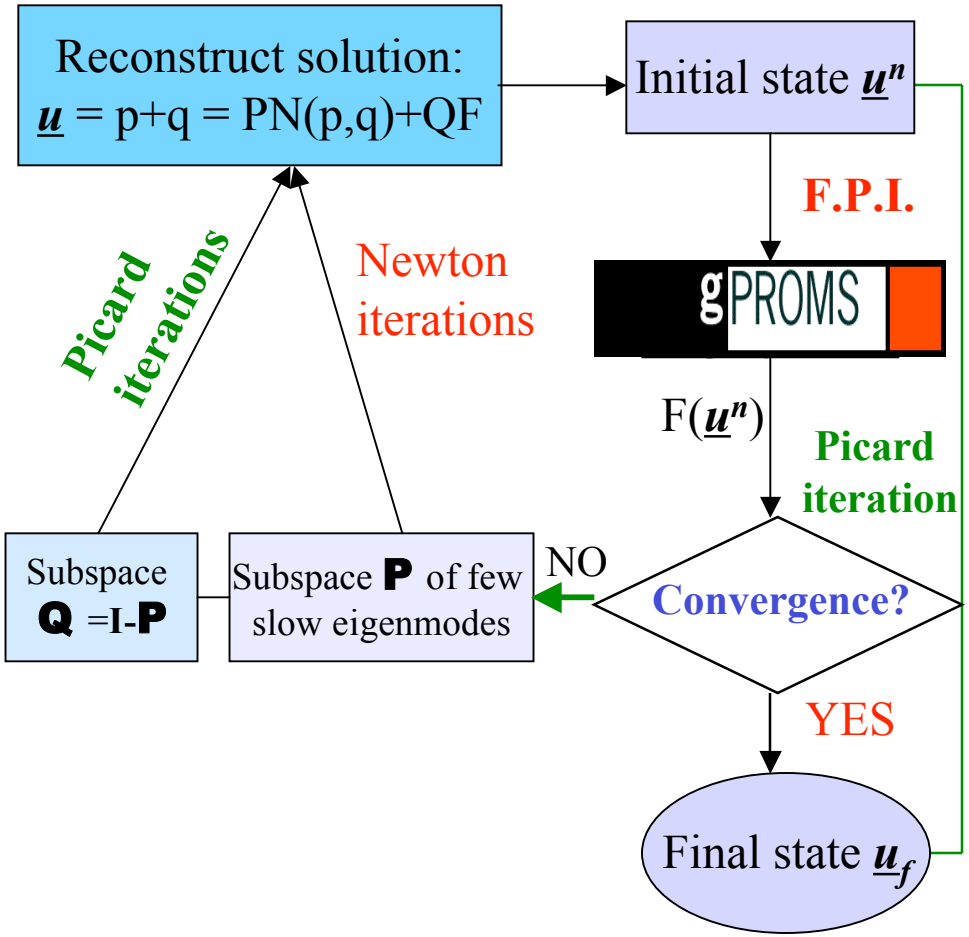
Linear Control Design: Pole Placement and 1 step ahead state prediction using Kalman filter

OPEN LOOP	CLOSED LOOP
$\lambda_1 = 0.87135$	$\lambda_1 = 0.87135$
$\lambda_{2,3} = 1.00130 \quad 0.00529i$	$\lambda_2 = 0.98$ and $\lambda_3 = 0.99$



Open-loop Closed-loop

Recursive Projection Method (RPM)



- Treats timstepping routine, as a "black-box"
  - Timestepper evaluates $\underline{u}^{n+1} = F(\underline{u}^n)$
- Recursively identifies subspace of slow eigenmodes, **P**
- Substitutes pure Picard iteration with
  - Newton method in **P**
  - Picard iteration in **Q** = I-**P**
- Reconstructs solution $\underline{u}$ from sum of the projectors P and Q onto subspace **P** and its orthogonal complement **Q**, respectively:
  - $\underline{u} = PN(p,q) + QF$

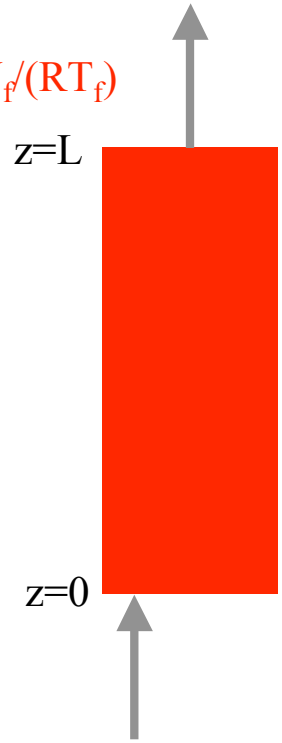
Rapid Pressure Swing Adsorption

1-Bed 2-Step Periodic Adsorption Process

t=0 to T/2

$C_i(z=0)=P_f Y_f/(RT_f)$

$P(z=0)=P_f$



Step 1 : Pressurisation

- Isothermal operation
- Modeling Equations (Nilchan & Pantelides)

Mass balance in ads. bed

$$\varepsilon_t \frac{\partial C_i}{\partial t} + \rho_b \frac{\partial q_i}{\partial t} = -\frac{\partial (vC_i)}{\partial z} + D_i \frac{\partial^2 C_i}{\partial z^2}$$

$$\frac{P}{RT} = \sum_{i=1}^{n} C_i$$

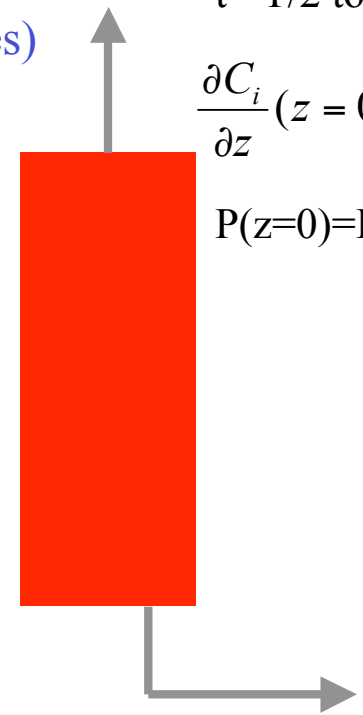
$$\frac{\partial P}{\partial z} = -\frac{180 \mu v}{d_p^2} \frac{(1-\varepsilon_b)^2}{\varepsilon_b^3}$$ Darcy's law

$$\frac{\partial q_i}{\partial t} = k_i (m_i p_i - q_i)$$ Rate of ads.

t= T/2 to T

$\frac{\partial C_i}{\partial z}(z=0) = 0$

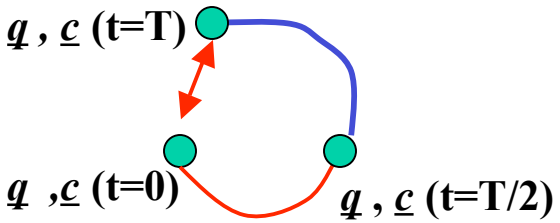
$P(z=0)=P_w$



Step 2: Depressurisation

Rapid Pressure Swing Adsorption

1-Bed 2-Step Periodic Adsorption Process



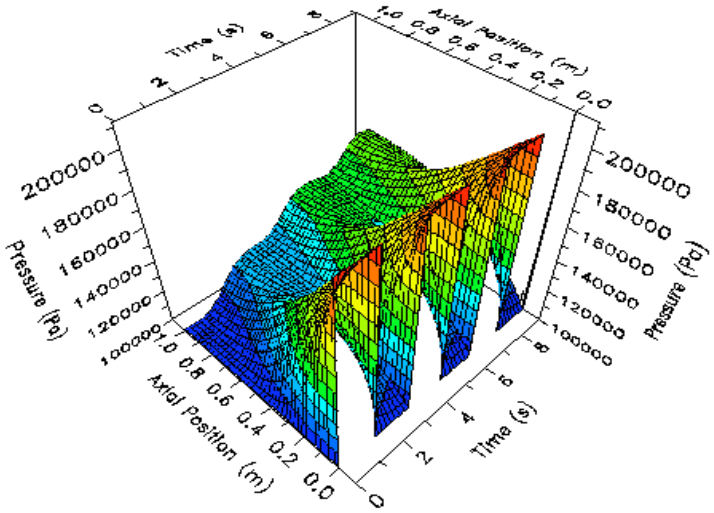
Must obtain:

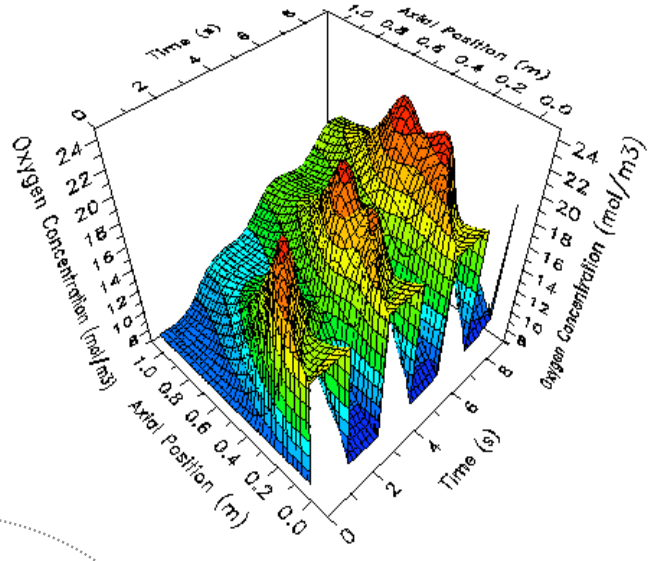
$q , \underline{c}\ (t=T) = q , \underline{c}\ (t=0)$

- Production of oxygen enriched air
- Zeolite 5A adsorbent (300μm)
- Bed 1m long, 5cm diameter
- Short cycle
  - 1.5s pressurisation, 1.5s depressurisation
  - T= 3s
- Low feed pressure (P_f = 3 bar)
- Periodic steady-state operation
  - reached after several thousand cycles

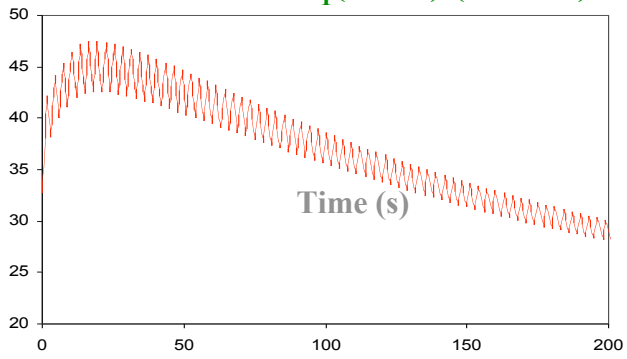
Typical RPSA simulation results

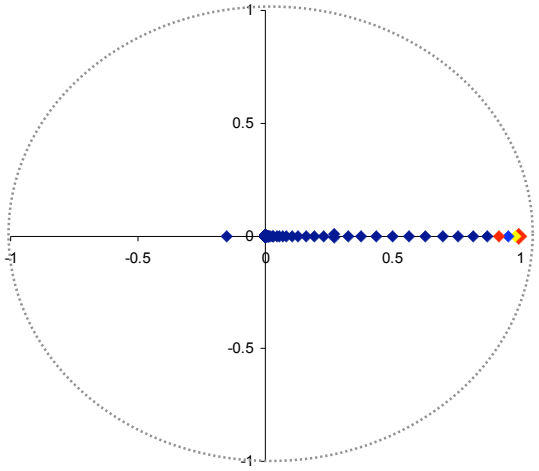
(Nilchan and Pantelides, *Adsorption*, 4, 113-147, 1998)



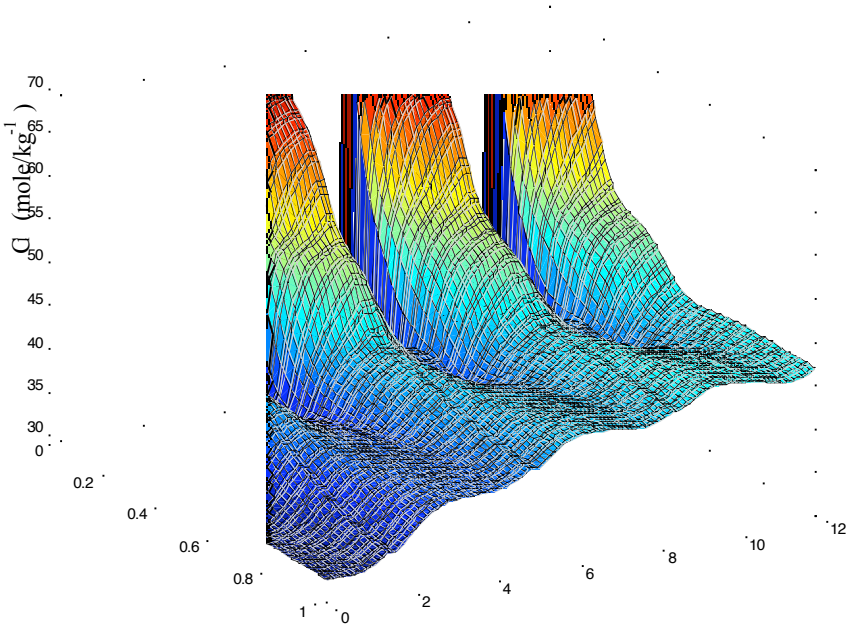


$c_1(z=0.5)$ (mol/m^3)

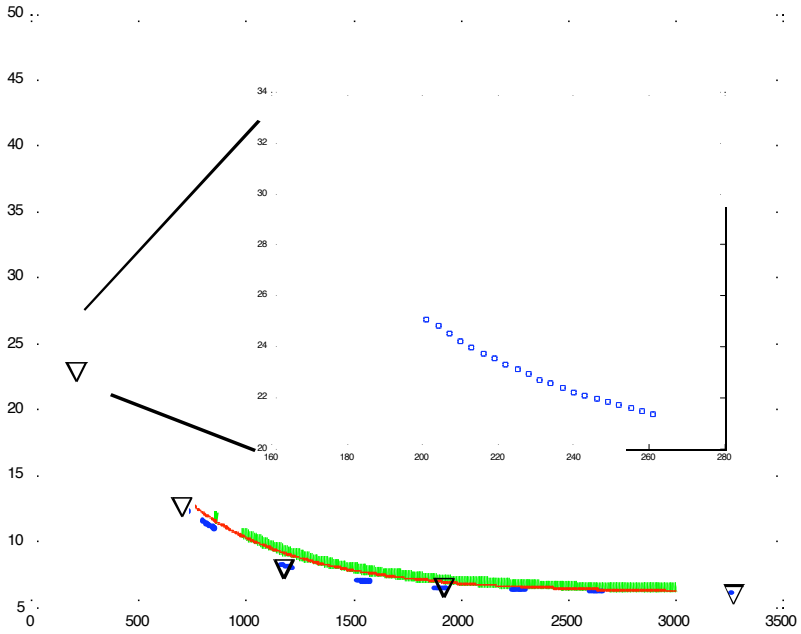




RPSA simulation results



COARSE INTEGRATION



PRM-gPROMS Spatial Profiles (t=T)