Computational MultiScale Modeling

C. W. Gear

NEC Laboratories America (retired)

Princeton University (part time)

Computational Modeling Objectives

- **Fundamental understanding – investigating consequences of assumptions**
- **Design (via analysis)**
  - **for example**
    - **Chemical processing for optimal production**
    - **Airflow for efficient aircraft design**

The greater the accuracy of the model, the better the understanding, or the larger the increase in the efficiency

Computer models of Physical, Economic, Social, and other processes

- Models have to be simplified to keep computational time and space manageable.
- Usually simplified to exclude details thought to be unimportant – but these may require simulation to determine whether they are unimportant!
- Many details are stochastic in nature, although models may only describe average behavior.
- Simplification usually means ignoring small-scale behavior although it may play a significant role in some regions.

Multiple Scale Issues

- Higher accuracy usually requires considering smaller-scale phenomena.
- Modeling on small scales in space also requires resolving small scales in time for 2 reasons:
  - Small spatial scales features frequently have fast time responses
  - To maintain numerical stability (due to the CFL condition and/or parabolic terms) requires small time steps when spce steps are small
- We are usually interested much longer times and larger spatial regions than the fine scale models. For example:
  - Turbulence on a plane wing is on a small scale compared to the size of the plane.
  - Molecular interaction times in chemical/biological models may be at the pico-second level or faster, while we are interested in hours or days.

“Closure” of fine-detail models

- Many high-dimensional Differential Equations (DEs) have lower-dimensional manifolds to which all solutions tend. Trivial Example: all solutions of

$$\frac{dy}{dt} = \begin{bmatrix} y_1' \\ y_2' \\ y_3' \\ y_4' \end{bmatrix} = \begin{bmatrix} -10^6 & 0 & 0 & 0 \\ 0 & -10^6 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} y + \begin{bmatrix} 10^6 y_3^2 \\ 10^6 y_4^3 \\ 0 \\ 0 \end{bmatrix}$$

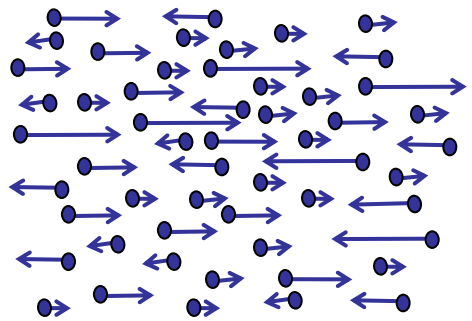
  tend towards the 2-Dimensional manifold $y_1 = 10^6 y_3^2, \;\; y_2 = 10^6 y_4^3$

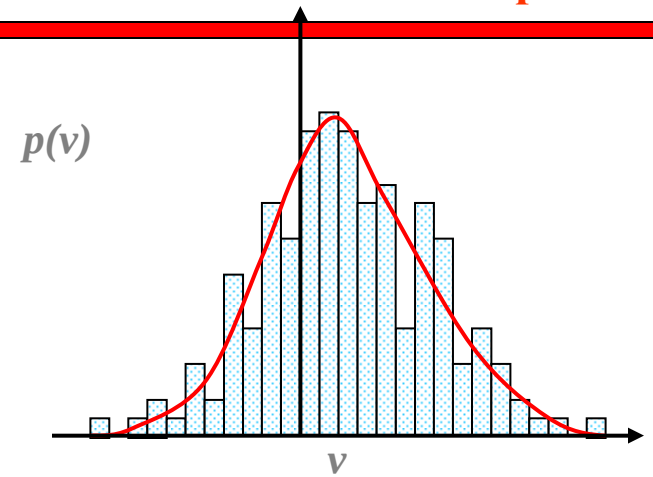
- Other very-high-dimensional problems have low-dimensional descriptors of *macroscopic* variables.

  Example: Fluid (gas) flow. Can be described accurately and microscopically as a collection of a vast number of interacting particles. We could describe this by the positions and velocities of every particle, by the probability distribution functions for the particle velocities, or in terms of the *macroscopic* variables pressure, density, and velocity. These are low-order *moments* of the probability distribution function of the particles. If we described the particle solution by all of its moments (instead of the probability distributions functions), we would find that the higher-order moments *close* on the lower order ones. , that is to say, they become functionals of the lower order ones.

  If the solution lies in this *closure* (low-dimensional manifold) we would like to compute only in that manifold and avoid the fine detail.

DISTRIBUTIONS & MOMENTS – A low dimensional description





Descriptions: 1: Particle positions and momenta 2: Approximate distribution of velocities

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

…..etc…….

Multi-scale Computational Modeling

Modeling the process at different scales in different regions. Either:

1. Same mathematical model in every region, or
2. More detailed model in finer-scale regions

We will concern ourselves with case 2. Case 1 is mostly a problem of using adaptive meshes.

Microscopic modeling: using descriptions at the fine scale

Macroscopic modeling: using descriptions at the coarse scale

Usually, macroscopic models are approximations that are sufficiently accurate in regions where the solution does not have a fine structure and the microscopic phenomena do not dominate. In this case, we wish to use microscopic models only in regions where macroscopic models are insufficiently accurate

In some cases, the macroscopic model is inaccurate even though the solution does not have a fine structure, because we do not know how to derive good approximations to the *closure equations.* In this case we must use the microscopic model everywhere.

Two types of problems in Multi-Scale Modeling:

1. If we know both the microscopic models and accurate macroscopic models, we can use microscopic models in regions where they are needed for accuracy and macroscopic models elsewhere.
   - One major issue is how to link the different regions together: e.g. a particle model region to a continuum region
2. If we do not know the macroscopic closure of the microscopic model (or we have an inaccurate macroscopic model) we must use the microscopic model wherever we compute, and find ways to avoid computing over the whole region.
   - The issue is how to use microscopic information to determine macroscopic behavior. We will focus on problem 2 in this talk.

The two levels of description

- Fine level (such as a particle description)
  - Very high dimensional
  - Able to represent fine detail
- Coarse level (such as finite element model of the density, pressure, and velocity)
  - Low-dimensional
  - Useful for large time and space scales

We have to have a mapping between them

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.

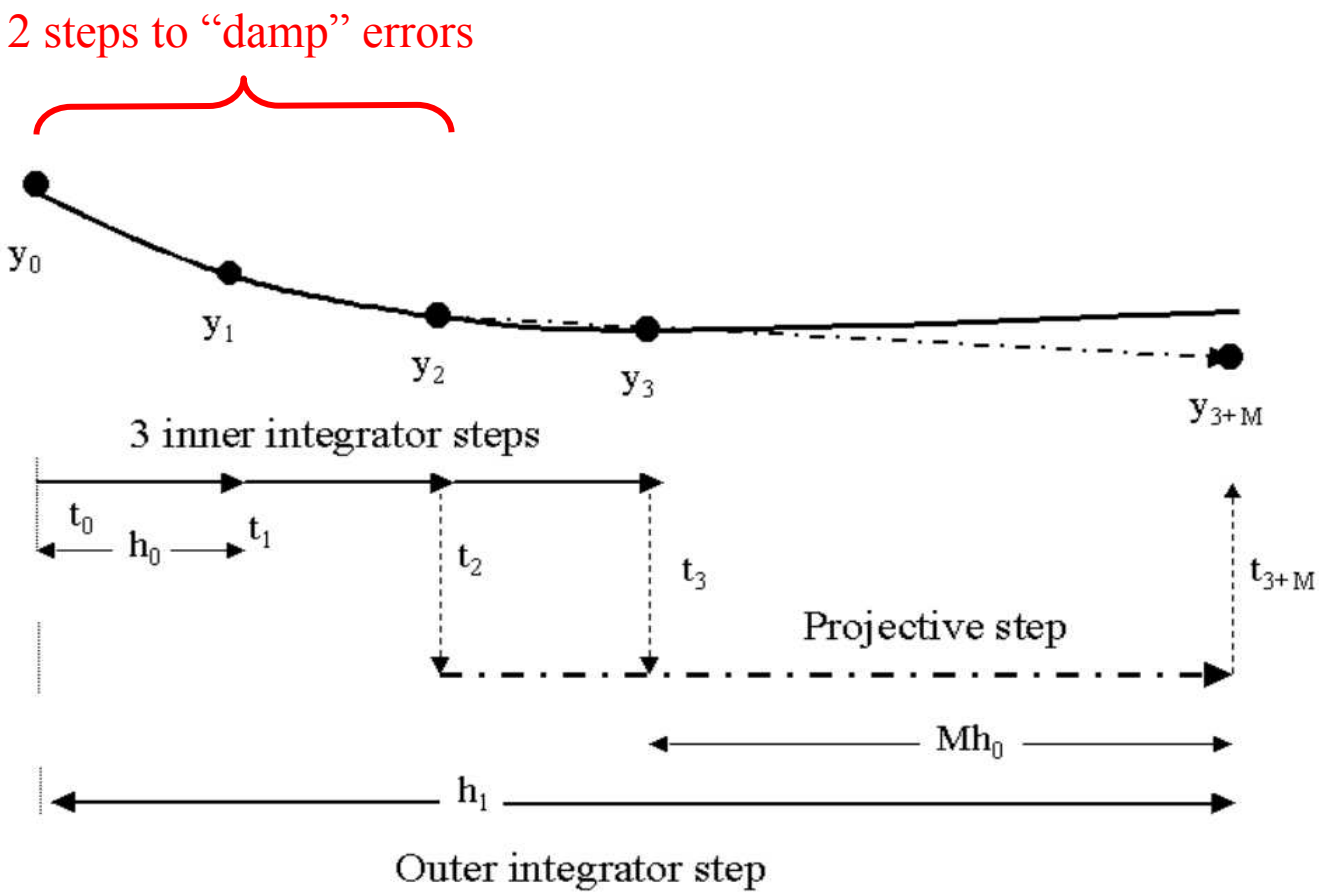
We do the macroscopic tasks in the low-dimensional description.

Time-domain multi-scale modeling

- Assumption that long-time behavior is slowly changing (smooth)
- Integrate microscopic model forward in time for short periods
- Use polynomial extrapolation to jump forward in time and reduce the total amount of microscopic simulation

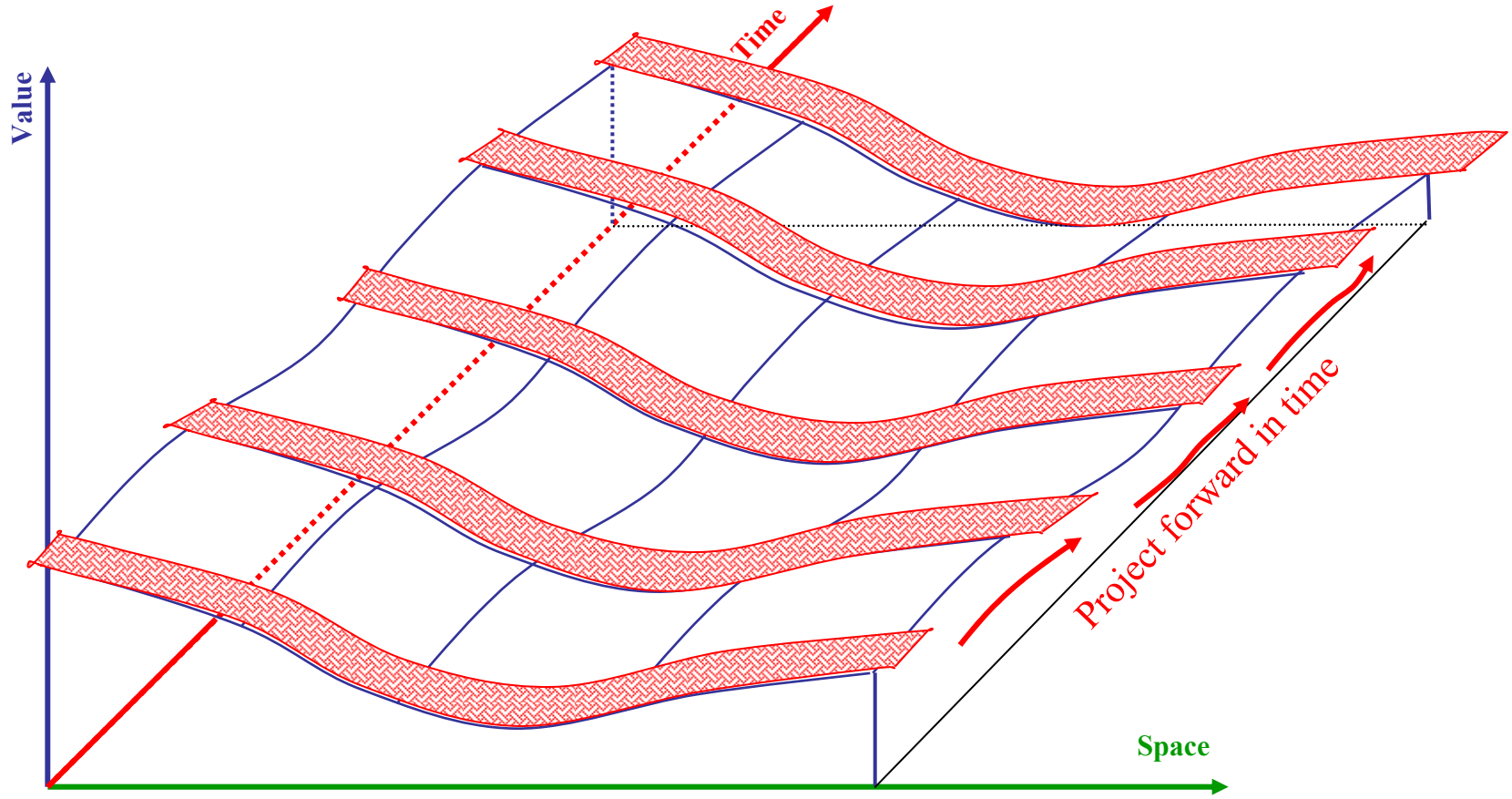
We call this ***Projective Integration***

Projective Integration - a simple example



Projective Forward Euler Method - linear fit to last two points after several "damping" steps - here, 2.

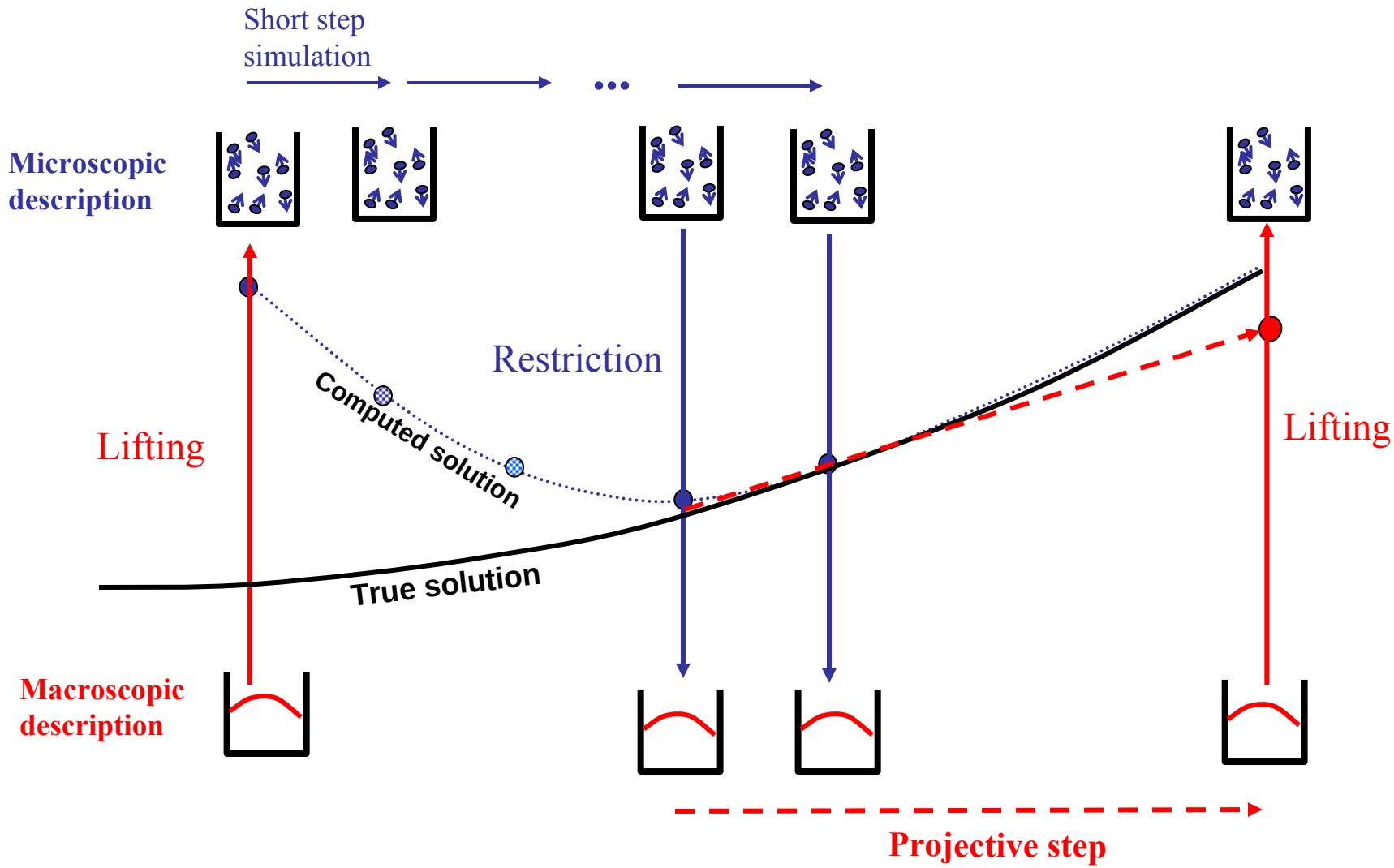
Time-domain multi-scale modeling



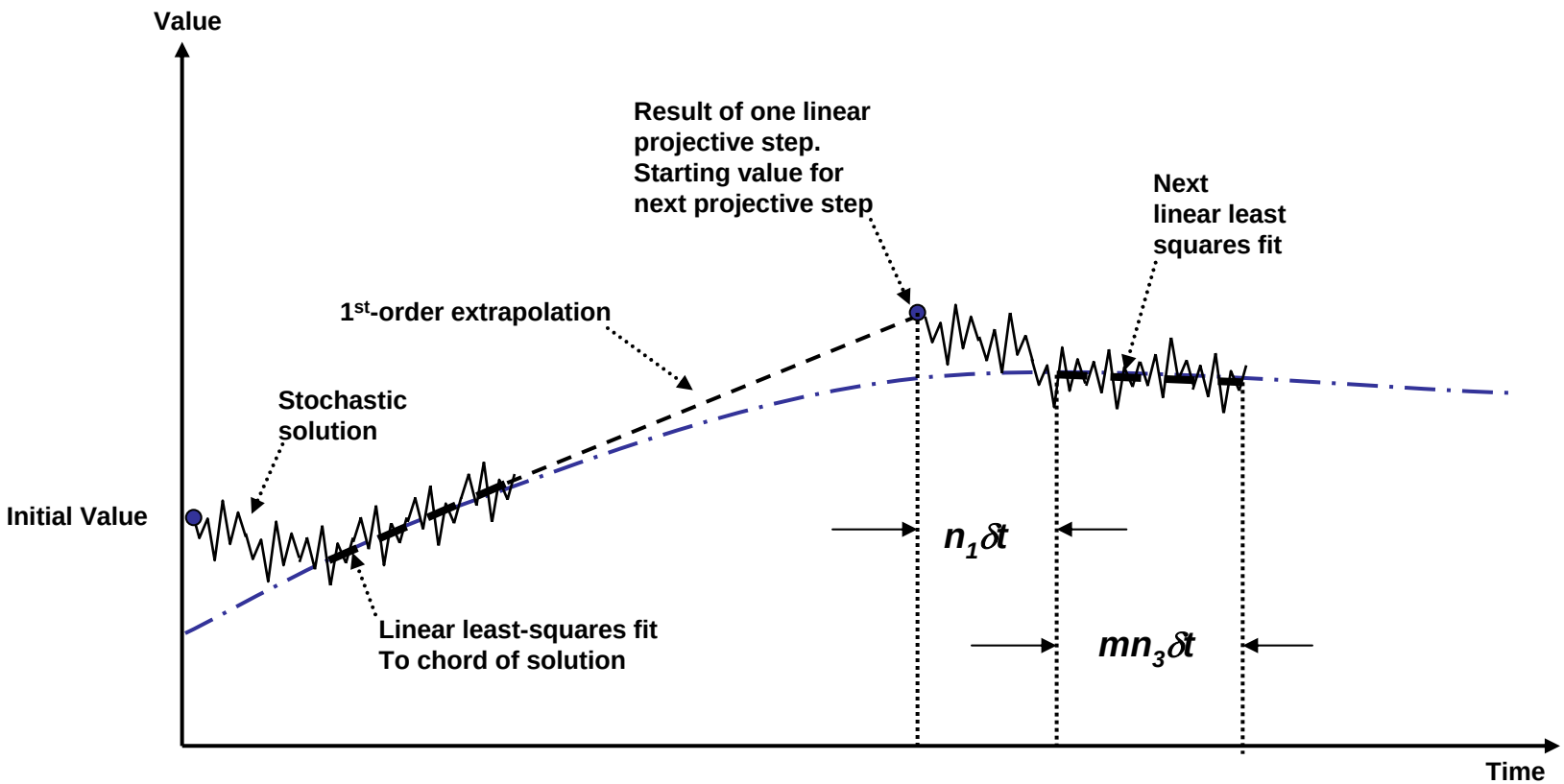
Projective methods in time:

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

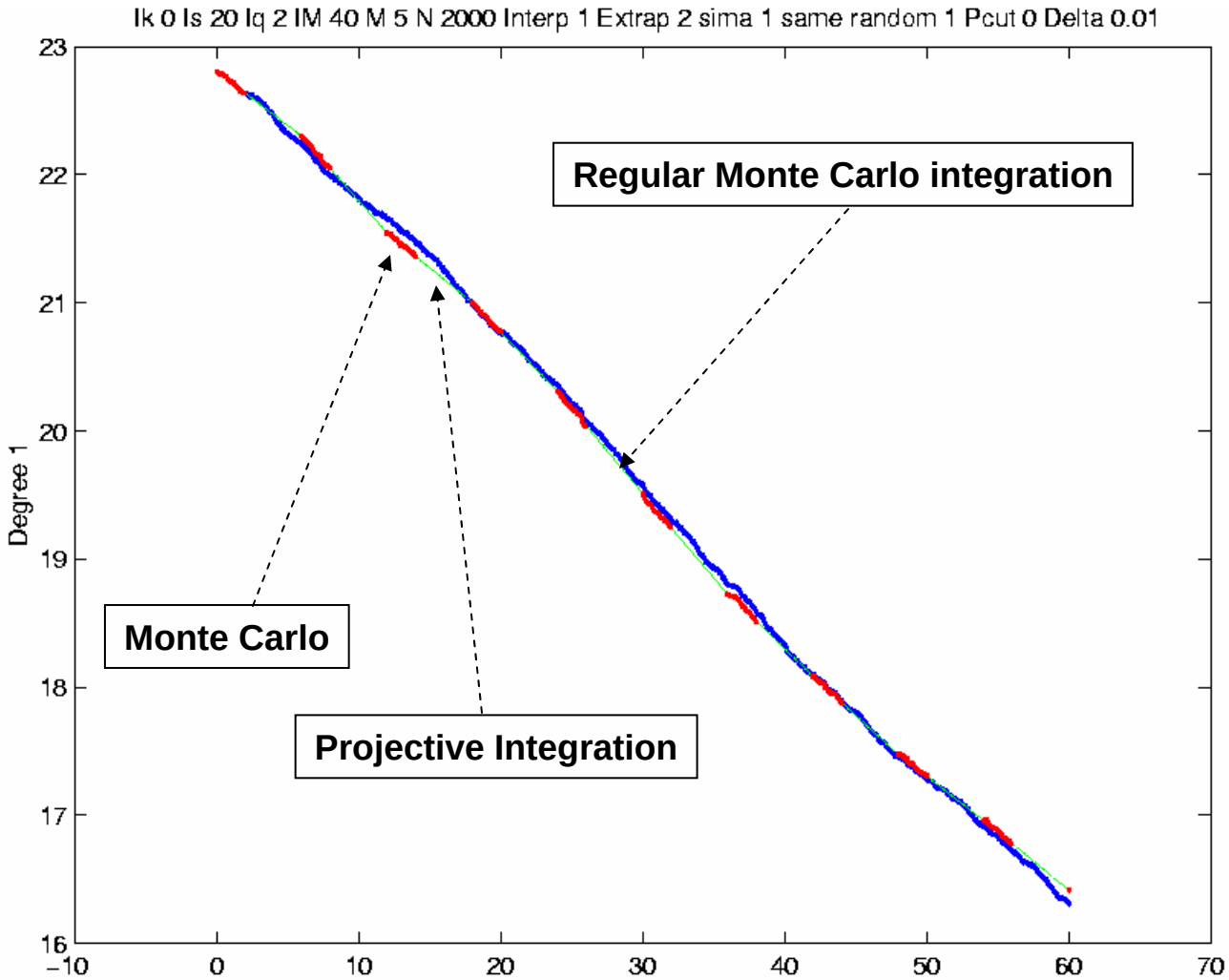
Restriction & Lifting with Projective Integration

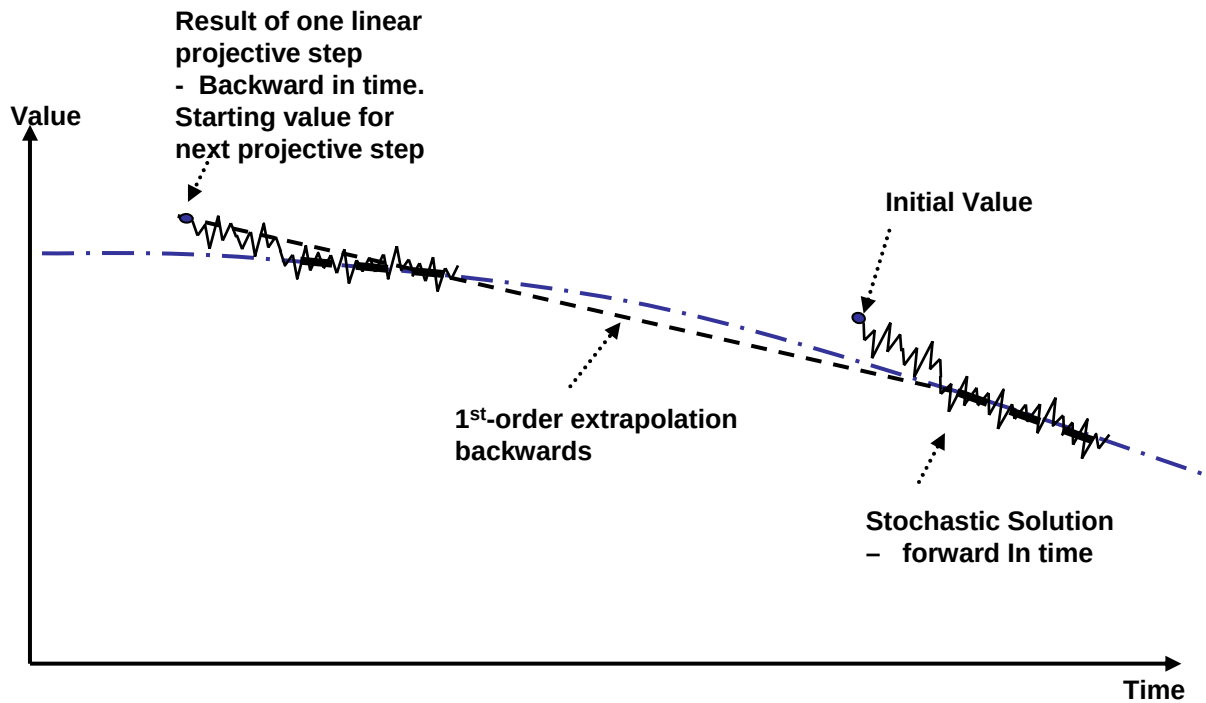


If the integrator is “noisy” (stochastic) then fit chord over several steps:



Example integration of a Stochastic Problem





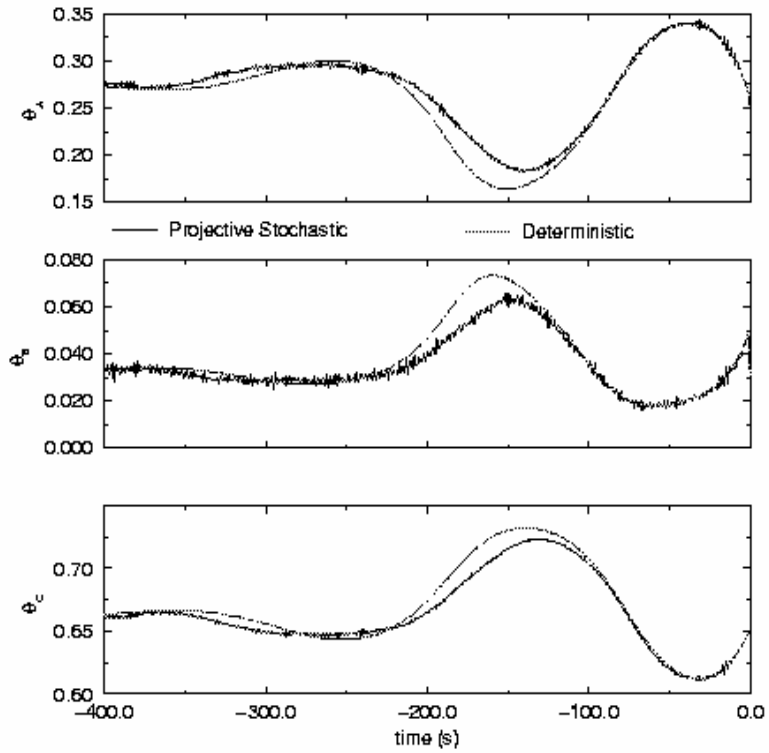
Although Monte Carlo particle solution can only proceed forward in time, we can integrate in reverse direction by taking the projective step backwards in time

Example: a simplified model of oxidation of CO (Θ_A) by dissociatively adsorbing oxygen (Θ_B) on a Pt catalyst in the presence of an addition inert species (Θ_C). Simulated by kinetic Monte Carlo (forward) and projective reverse (compared with direct solution of known ODEs – deterministic)

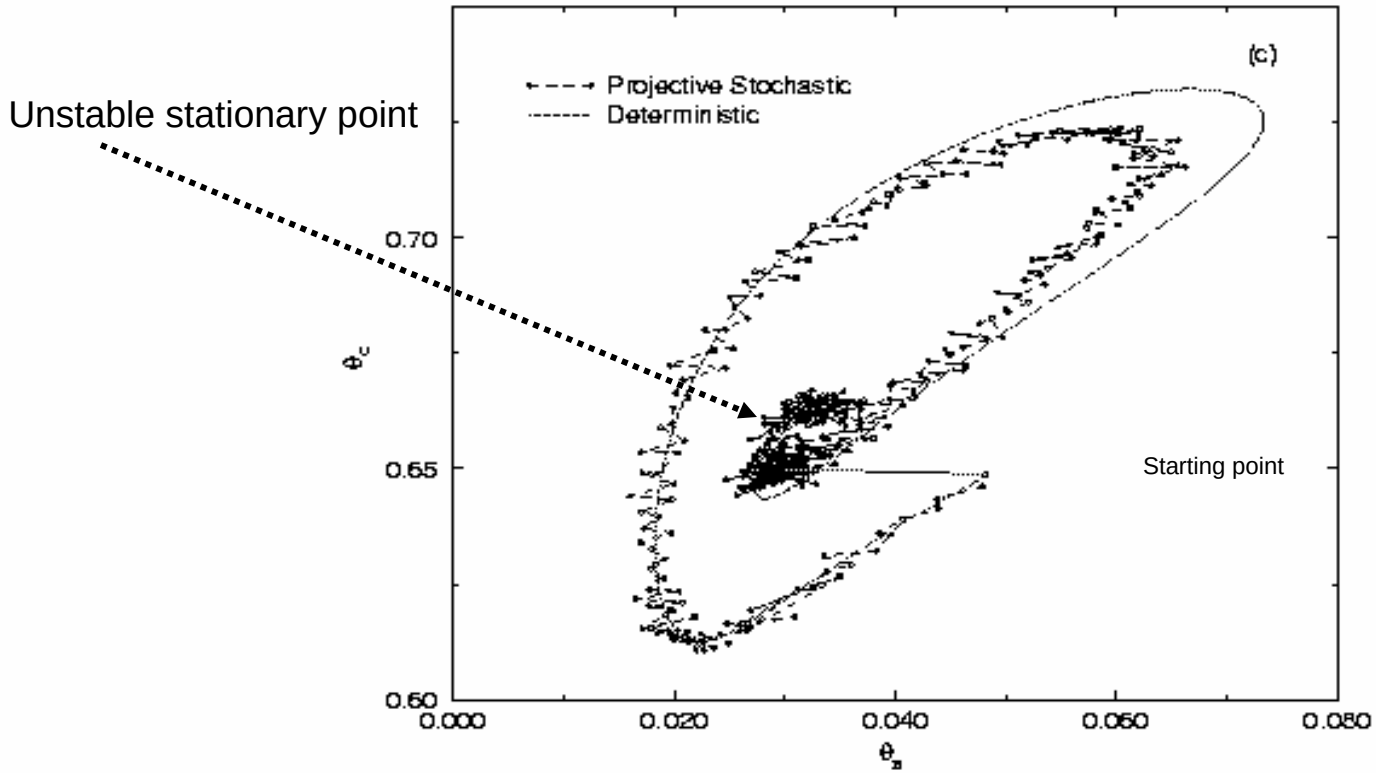
ODEs

$$\frac{d\theta_A}{dt} = \alpha\theta_* - \gamma\theta_A - 4k_r\theta_A\theta_B$$
$$\frac{d\theta_B}{dt} = 2\beta\theta_*^2 - 4k_r\theta_A\theta_B$$
$$\frac{d\theta_C}{dt} = \mu\theta_* - \eta\theta_C$$

The solution is integrated backwards from t = 0.0 to t = -400.0, at which time it is near an unstable stationary state, as shown on the next slide:



Phase-plane plot of the second and third components in a reverse integration.



The next slide shows the results of a *chemotaxis* model of moving bacteria. In this model each cell is either moving or stationary. It switches from stationary to moving at random times, and chooses a random direction (here either left or right). It switches back to being stationary at a random time, but this event occurs with lower probability if the cell is sensing that it the local concentration of *chemoactrant* (a chemical that it likes, such as food) is higher than it was a little while ago. (This is handled via time delay mechanisms in the cell and simulated by differential equations.)

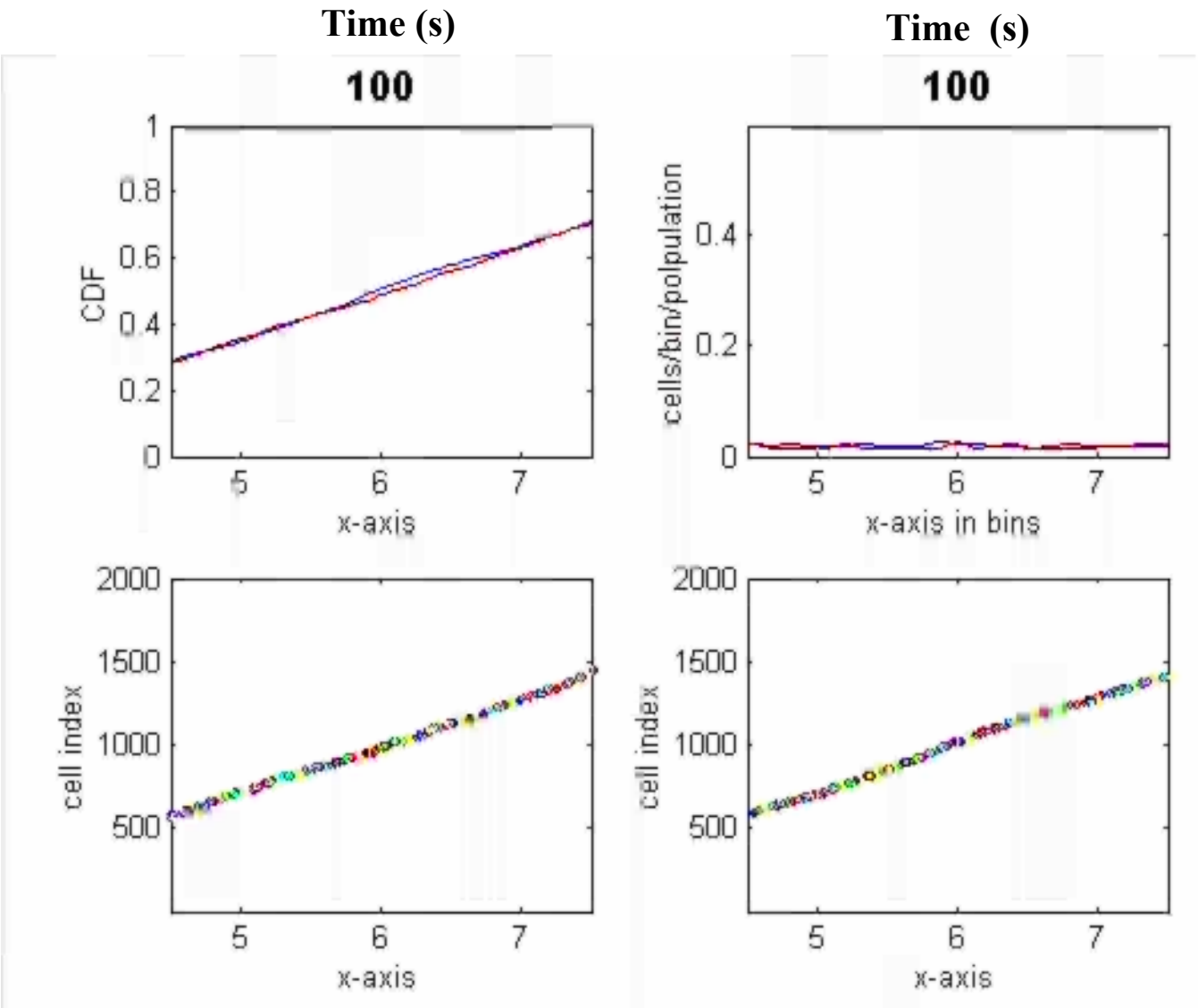
Because it tends to remain moving for longer periods when it is moving in a direction of increasing chemoactrant levels than when they are decreasing, the net movement is in the increasing direction, so the cell is “attracted” to high levels of chemoactrant.

One expects that it should be possible to work with a macroscopic description of the system in terms of the density of the cells, rather than at the microscopic level of individual cells. (Indeed, this has been done for many situations.)

We simulate at the microscopic level and use as a macroscopic model a low-order approximation to the *cumulative distribution function* of the assumed probability density function of the cells in space.

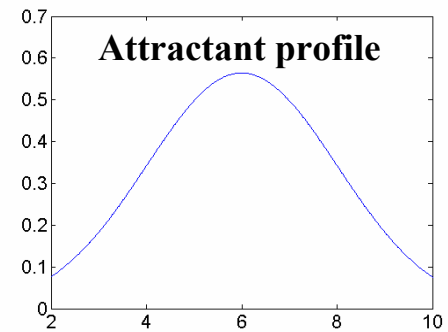
Chemotaxis: Coarse Projective Integration

(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

Note: this slide contains movies that can only be viewed in power point.

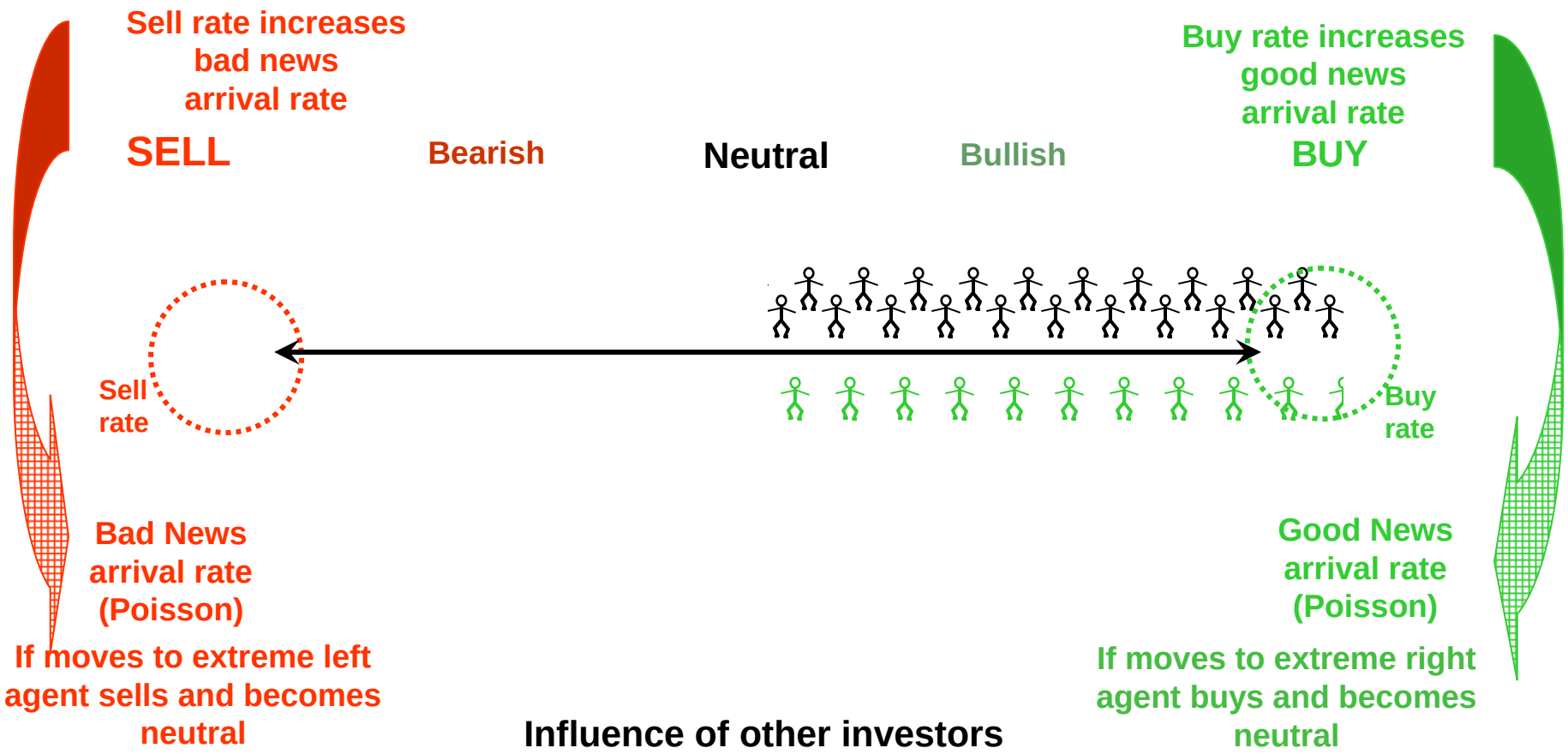
A Second Example

- Here we are dealing with a population of people each in a particular state (in this case, with a value between -1 and 1) and changing states on a stochastic basis.
- If we have large number of people we can think of the probability density function of states, x, between -1 and 1. If this is p(x) it means that the expected number of people with states between x and x+dx is Np(x)dx if the total number of people is N.

Example: a Financial market model

- A collection of semi-independent *agents*
- Each has a “value” between -1 and 1 that indicates their desire to sell (a stock) if -1, or to buy it if +1. 0 means neutral.
- If they hear nothing, their value drifts to zero.
- Random arrival of good news and bad news causes them to jump positively or negatively.
- If they reach +1 they buy and return to zero
- If they reach -1 they sell and return to zero
- The rate of buying (selling) increases the arrival rates of the good (bad) news by a factor ***g***.

These rules are illustrated in the next slide which has to be viewed in power point



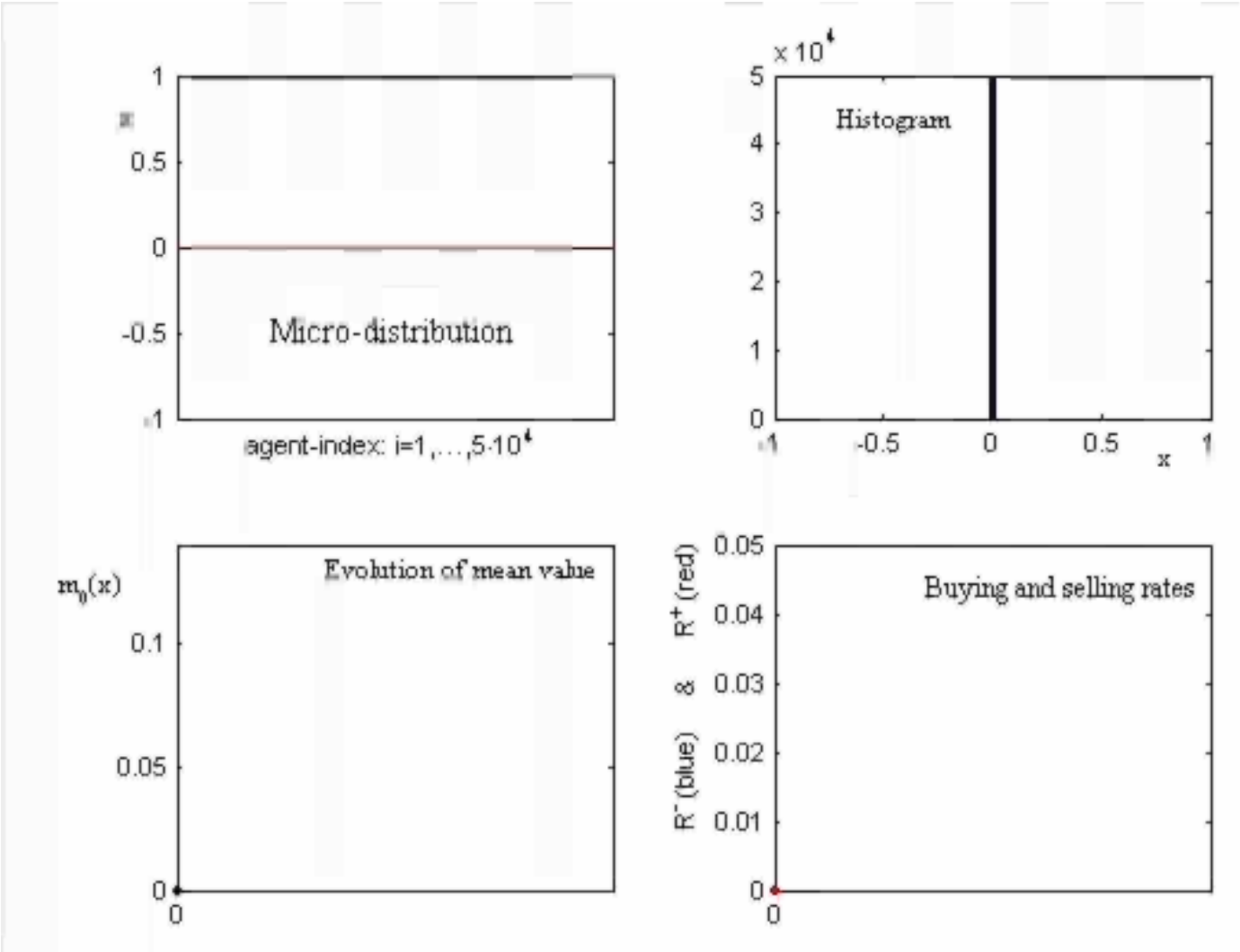
Influence of other investors

- or lemmings falling off a cliff

SIMULATION RESULTS

50000 agents, g=35, ε^{+}=0.075, ε^{-} = - 0.072, $v_0^{+}=v_0^{-}=20$

Open loop response, g = 42

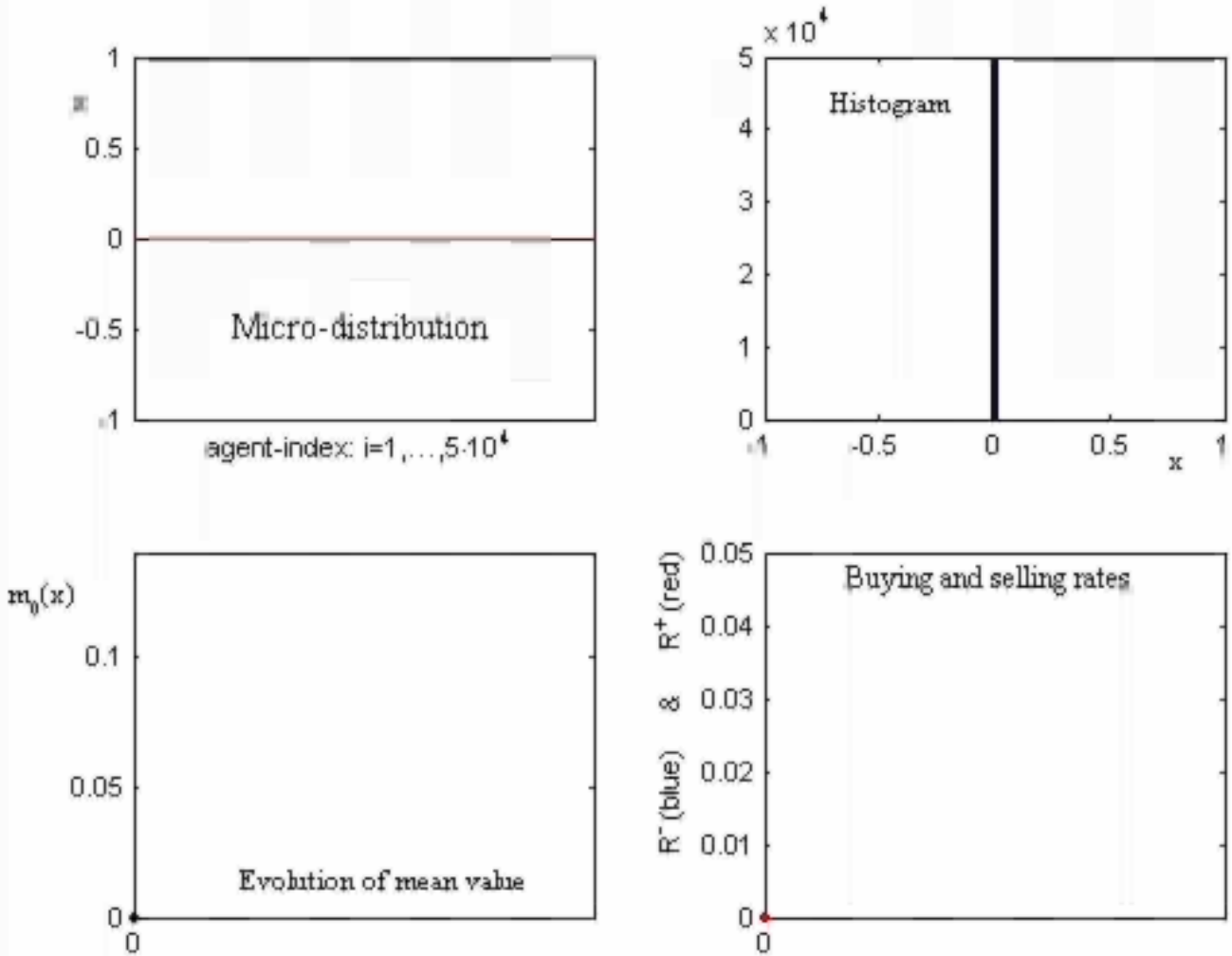


Note: this and the next two slides contain movies that can only be viewed in power point.

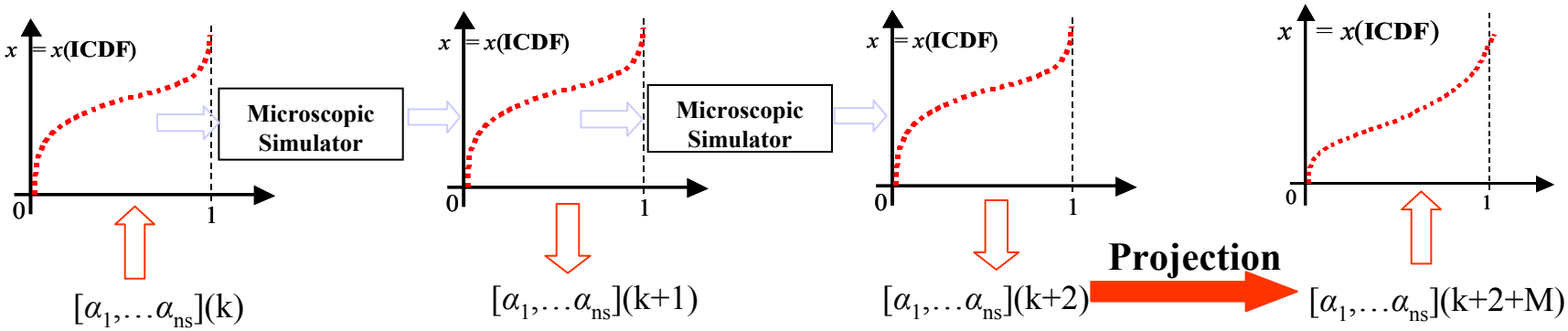
SIMULATION RESULTS

50000 agents, g=35, ε^{+}=0.075, ε^{-}=-0.072, $v_0^{+} = v_0^{-}$=20

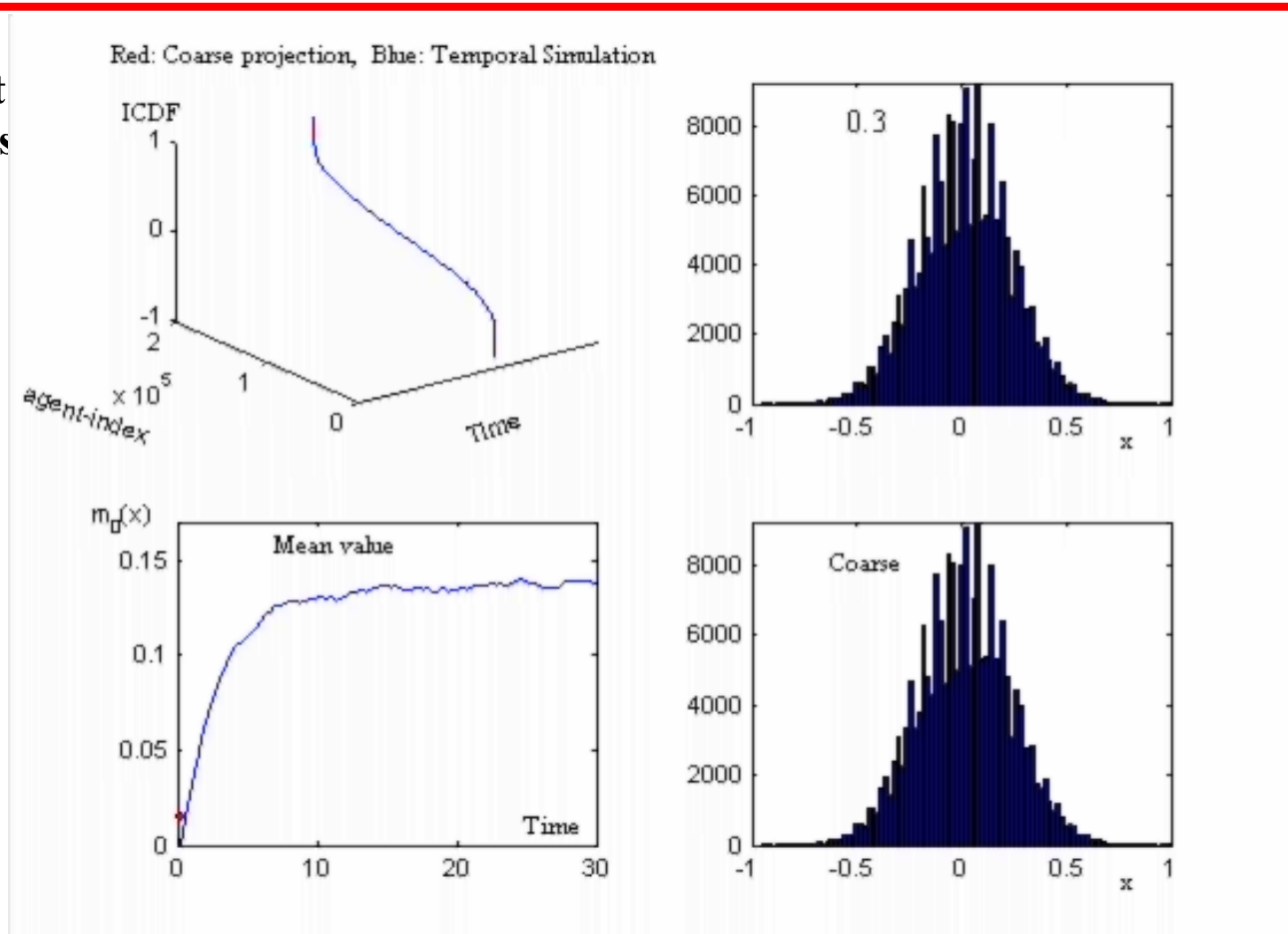
Open loop response, g = 46.5



Coarse projective integration: Accelerating things



Simulation results at g = 35, 200,000 agents



Run for 5x03 t.u. Project for 5x0.3 t.u.

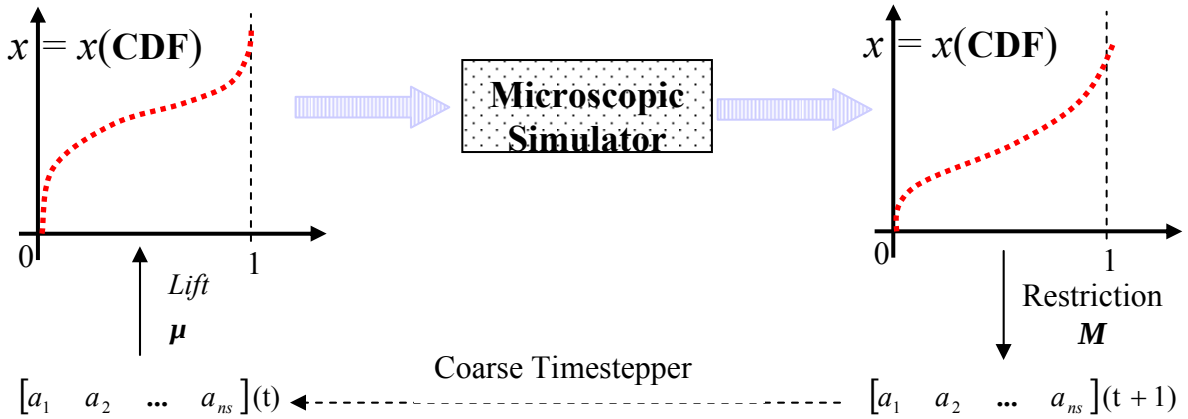
Bifurcation diagrams

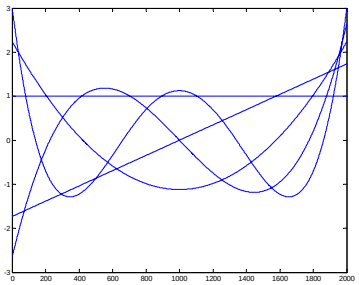
- When we are designing systems that are modeled by differential equations we are interested in how the steady states change with the change in parameters. Typically we find that as a parameter is changed, the values at the steady state change. If they are changing to more desirable values (indicating, perhaps, a more efficient operating point), we naturally want to increase the parameter further. Typically there are sudden discontinuities in the behavior as a function of a parameter, called *bifurcations,* where the system may lose its steady state. A bifurcation diagram shows the behavior of the stationary states as a function of a parameter, and we often wish to calculate these.
- This financial problem has a stable and unstable stationary states for small values of the parameter ***g*** but as ***g*** increases, the stable and unstable states approach each other, and then the system becomes unstable.

Finding stationary states

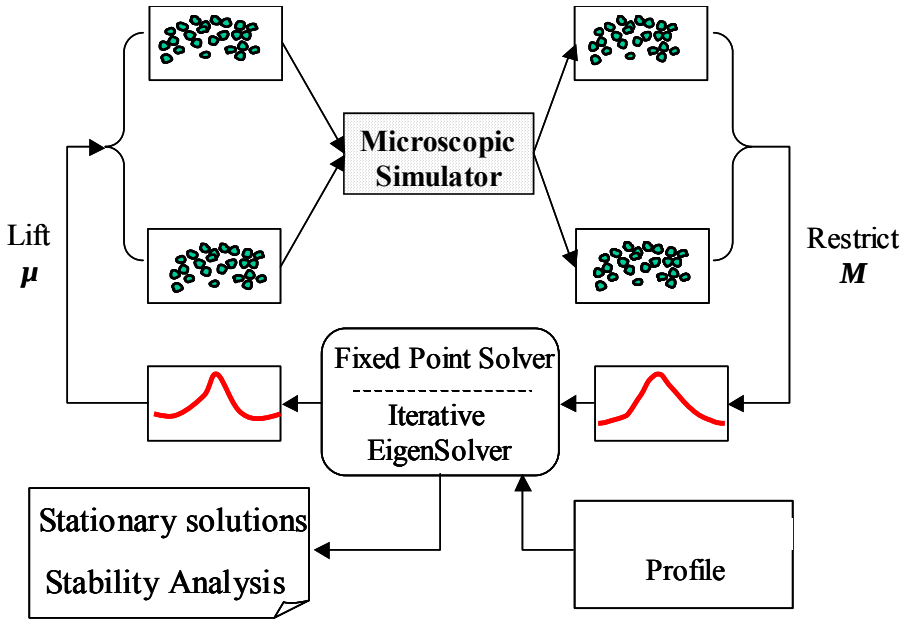
- We wish to find stationary states of the macroscopic variables.
- If we had a differential equation $y' = f(y)$ for the macroscopic variables we could find stationary states by solving $f(y) = 0$.
- When we only have a microscopic model, we must find the stationary state of the macroscopic *restrictions* of the microscopic model by simulation.
- The mechanism for this is shown in the next slide, and the process is illustrated for the financial model on the following two slides which contain movies that have to be run in power point.

The Coarse Timestepper





Orthogonal Polynomials, Fourier series ….

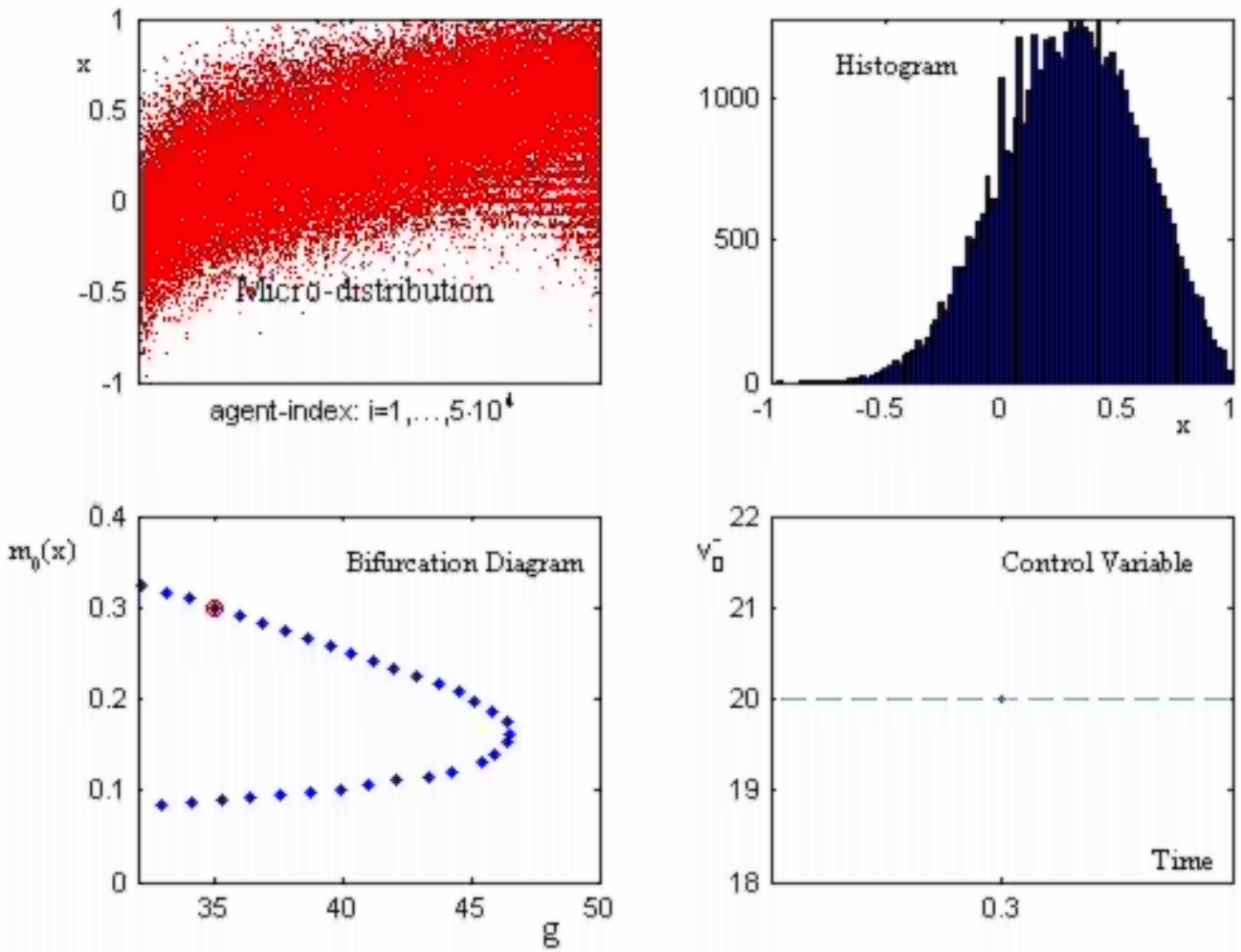


Coarse Stability/ Bifurcation Analysis

The Bifurcation Diagram

50000 agents, g=35, ε^{+}=0.075, ε^{-}=-0.072, $v_0^{+}=v_0^{-}$=20

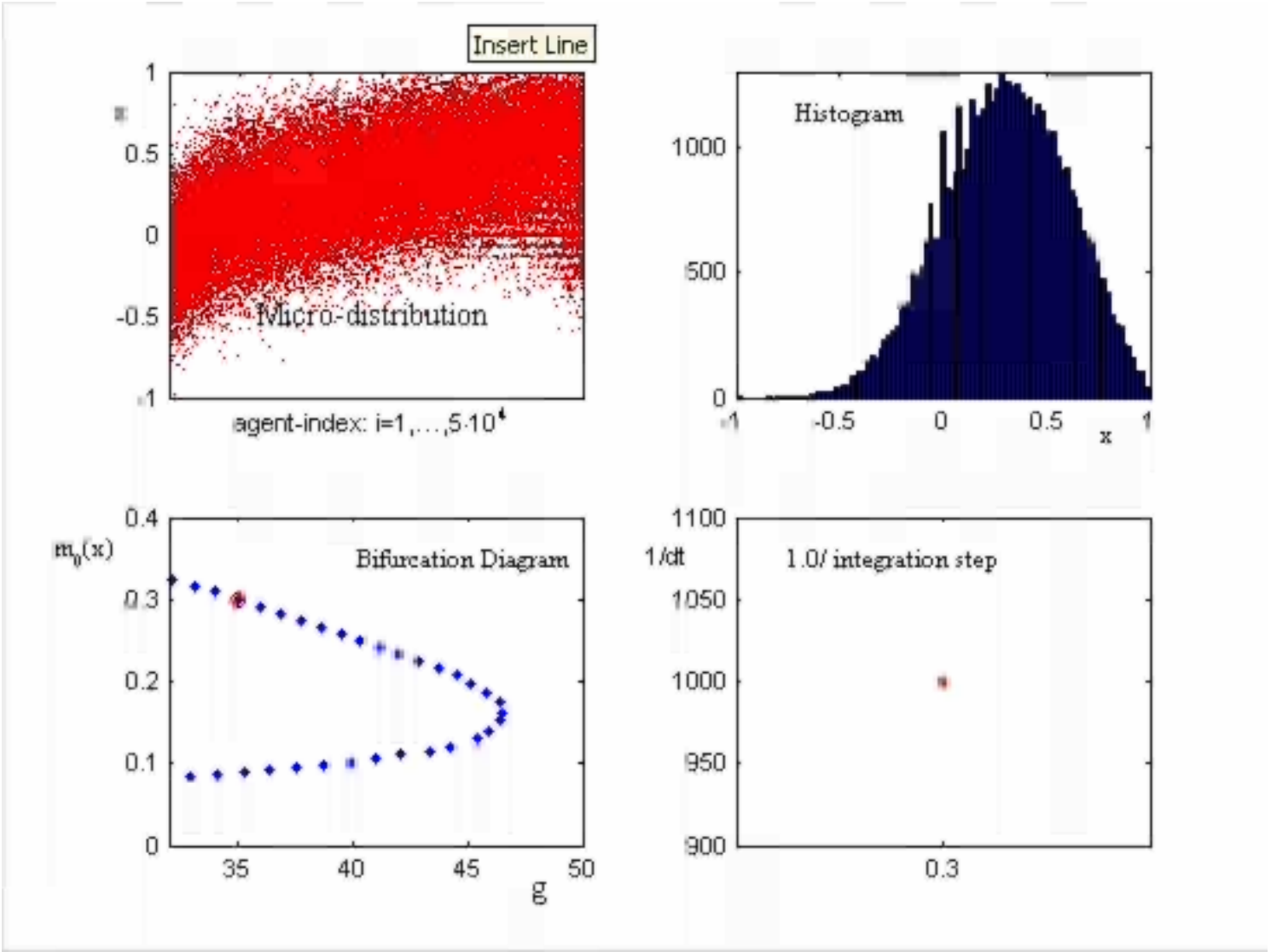
Open loop response. From unstable to stable markets



The Bifurcation Diagram

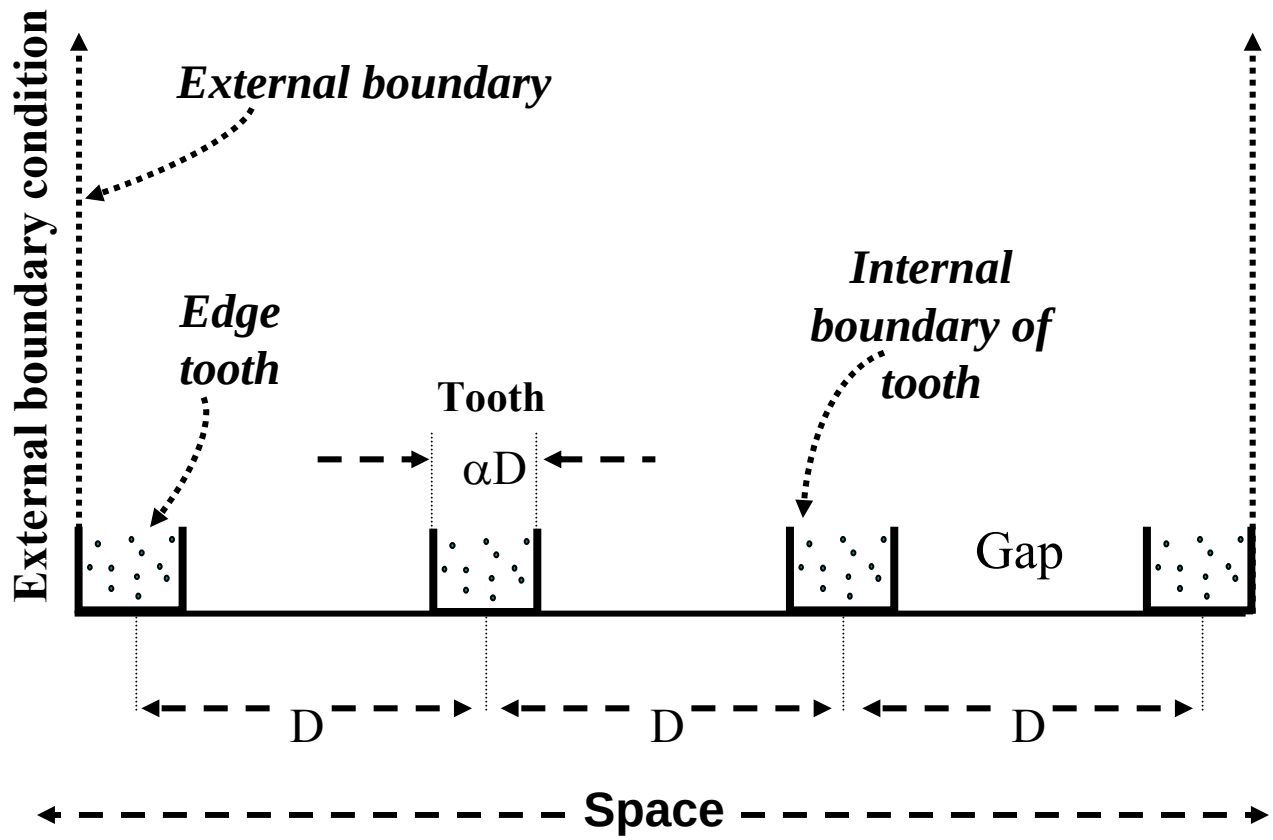
50000 agents, g=35, ε^{+}=0.075, ε^{-} = -0.072, $v_0^{+} = v_0^{-} = 20$

Open loop response. Blow up



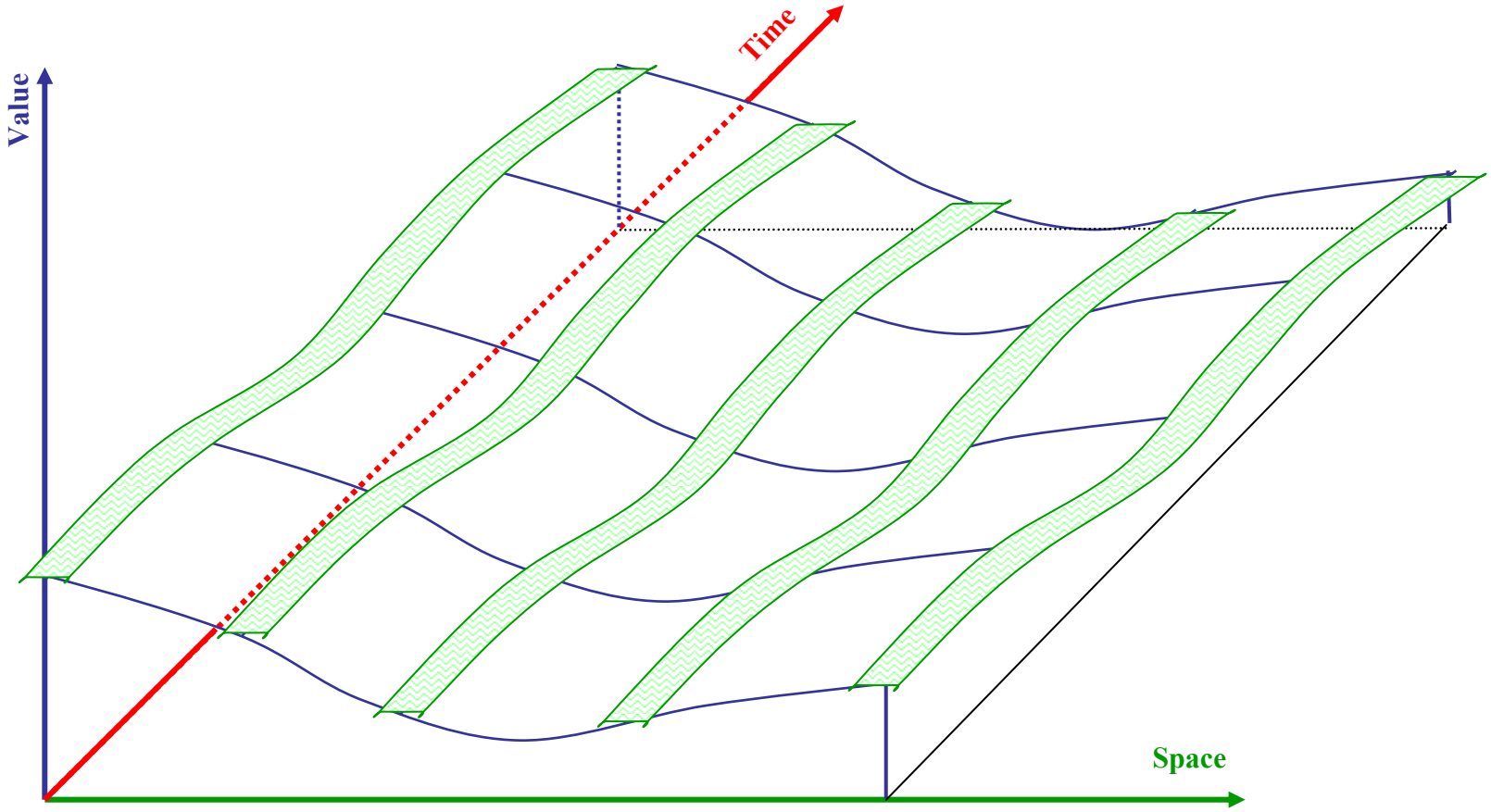
Space-domain multi-scale modeling

- Assumption spatial behavior in the macroscopic variables is smooth
- Integrate over small spatial regions, and assume that smooth functions can be used to interpolate between them
- Use these smooth functions to determine boundary conditions of small spatial regions



We call this the ***gap-tooth scheme***

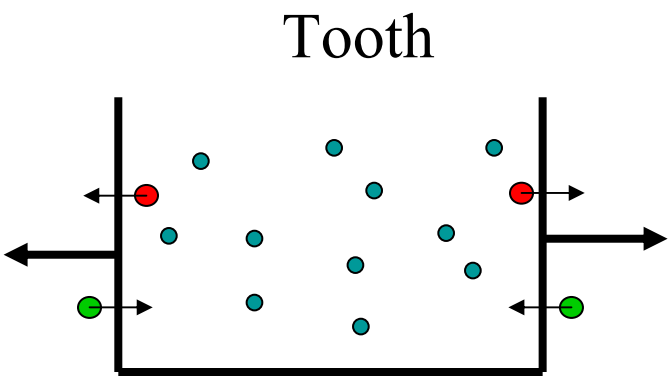
Space-domain multi-scale modeling



The Gap-Tooth method: detailed modeling in small spatial boxes with interpolation between boxes. Major problem is how to provide boundary conditions for small boxes.

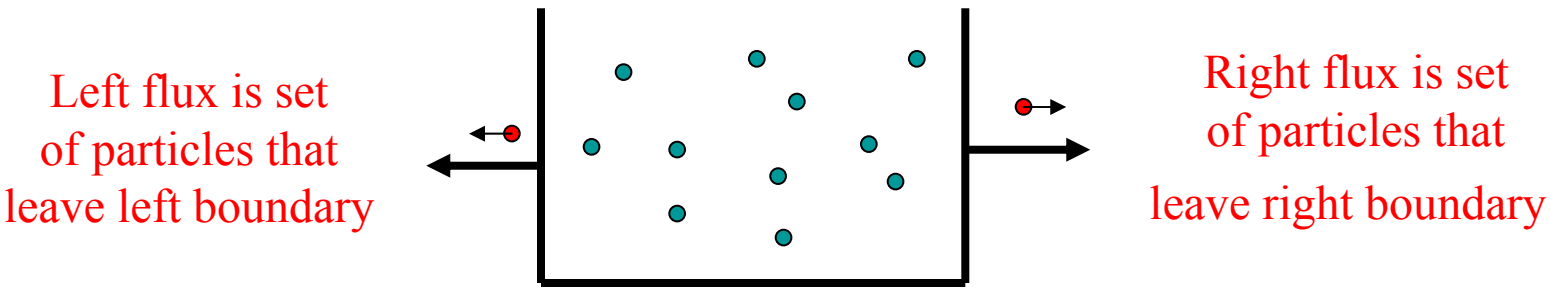
A novel scheme for gap boundaries

- when the microscopic simulation is particle based



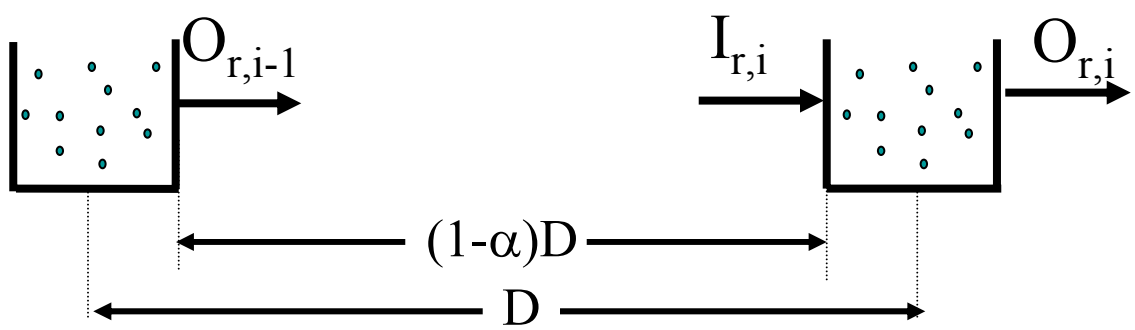
Simulation within tooth determines particles that *leave* tooth.
We only need to provide particles that *enter* tooth boundaries

Tooth

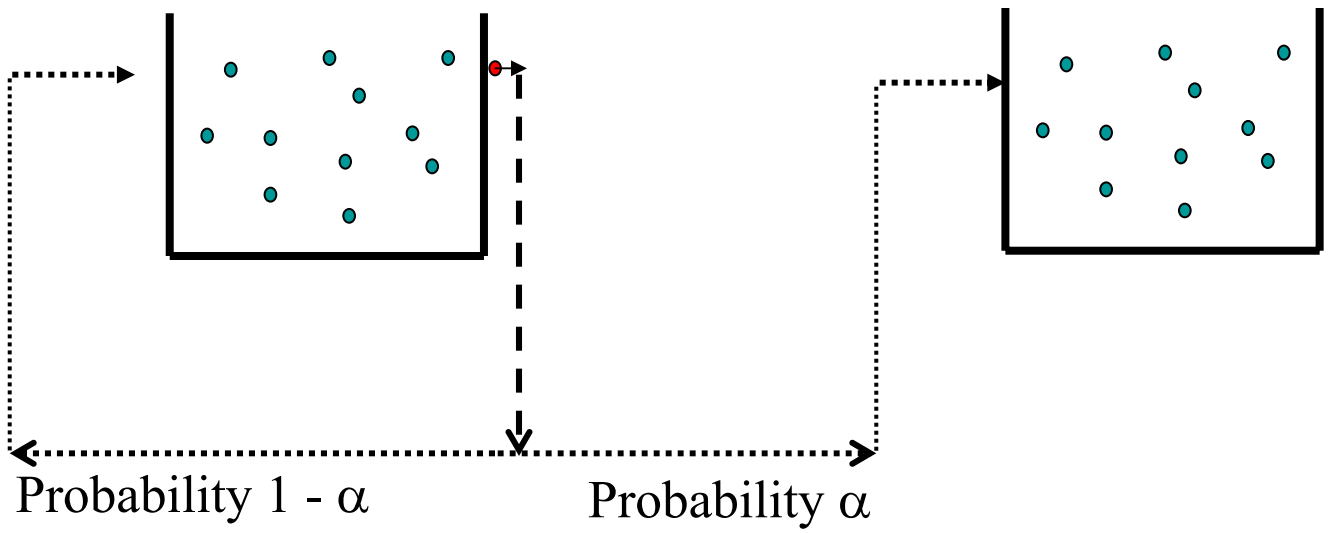


Suppose ratio of tooth width to spacing is α.
Let outgoing flux to right from box i be $O_{r,i}$ and incoming right flux be $I_{r,I}$ If it is reasonable to assume that flux is *smooth*

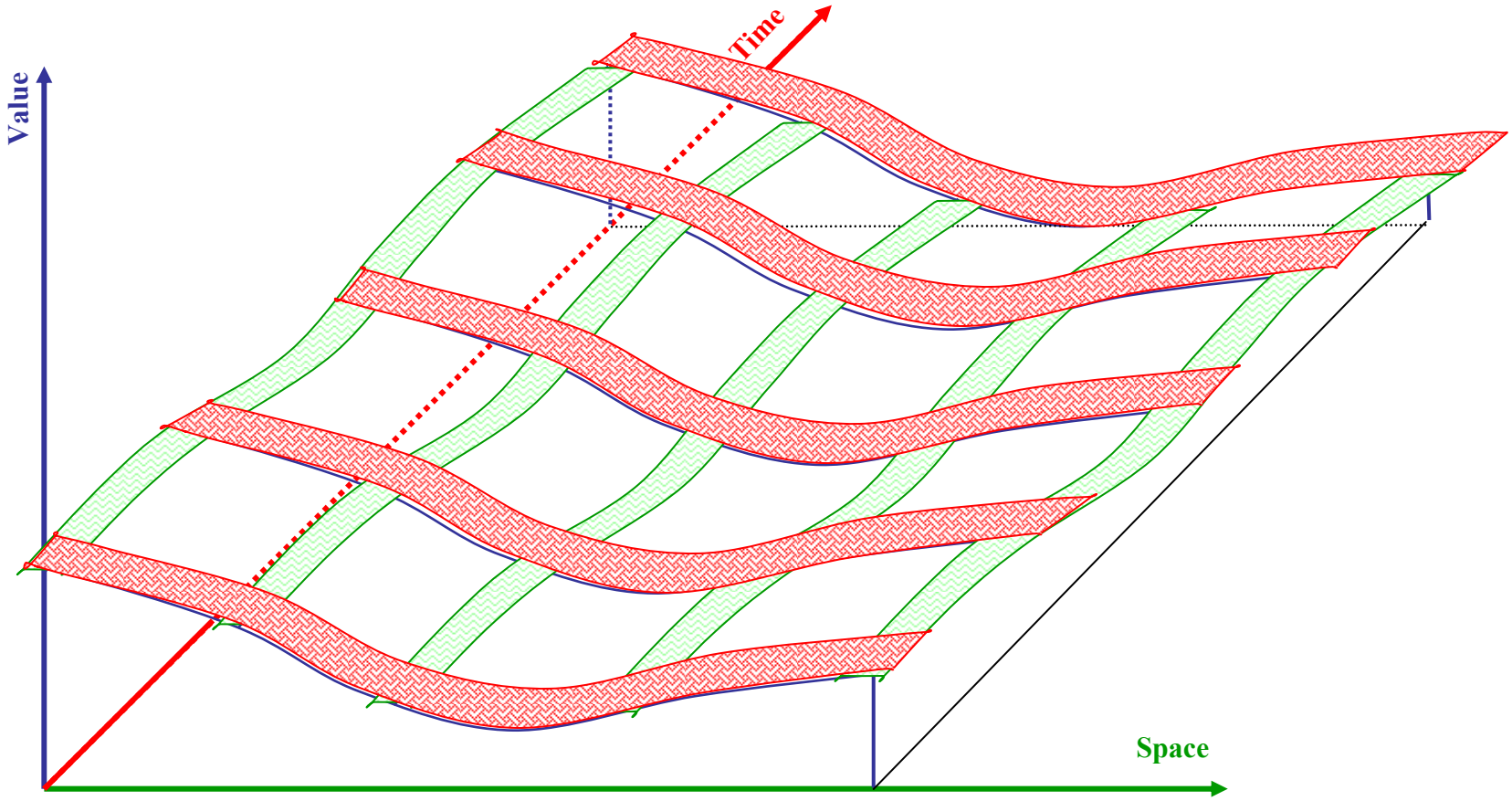
then, $$I_{r,i} = \alpha O_{r,i-1} + (1 - \alpha) O_{r,i}$$
is a first-order interpolation formula



Difficult to “interpolate” discrete events (particles leaving teeth). Instead, handled by sending α of the outgoing particles to next box and recycling $(1 - \alpha)$ of the particles to the same box. Called ***flux redistribution.***

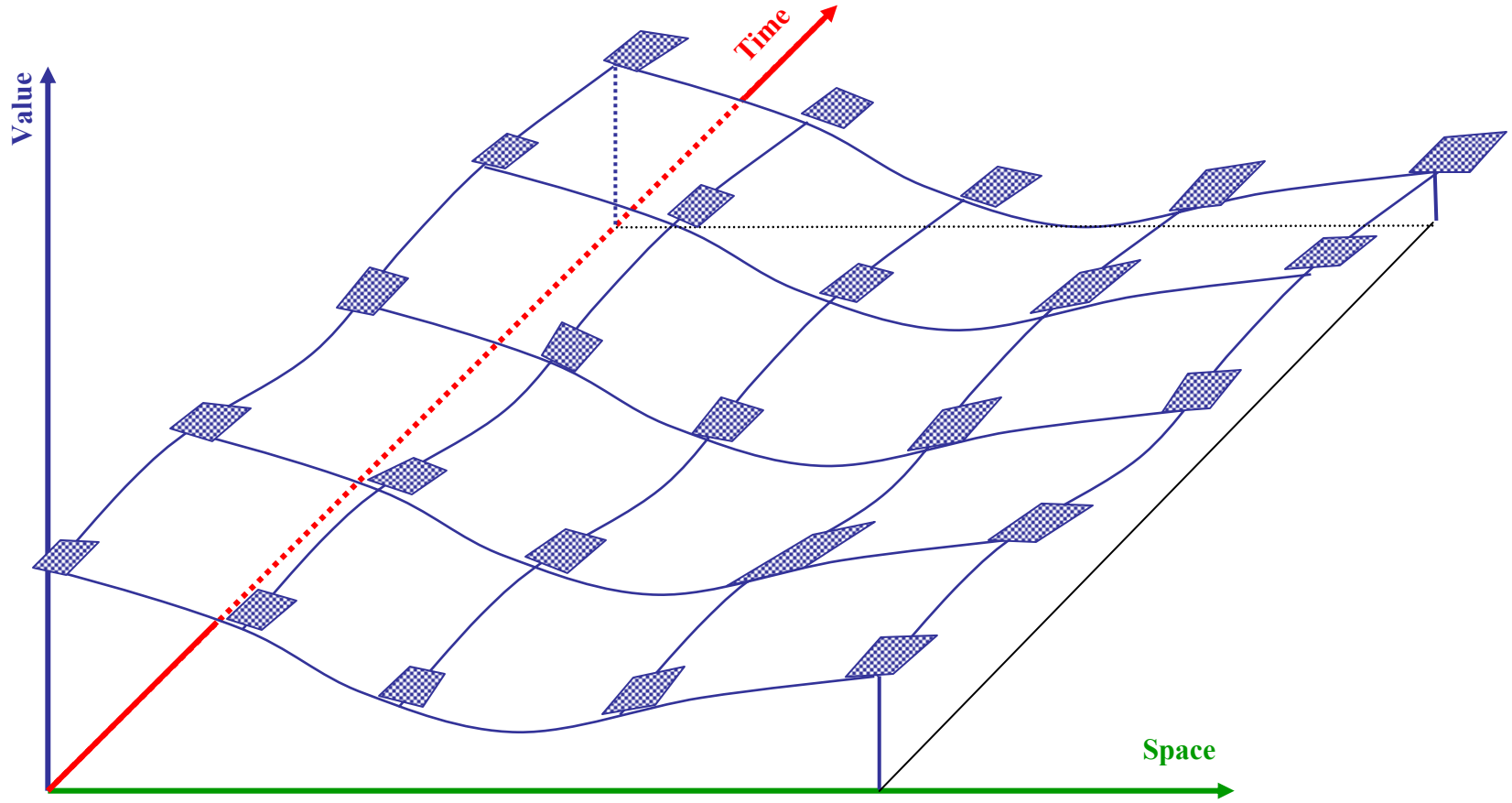


Multiscale Modeling Challenges:



Can we combine gap tooth with projective integration in time?

Multiscale Modeling Challenges:



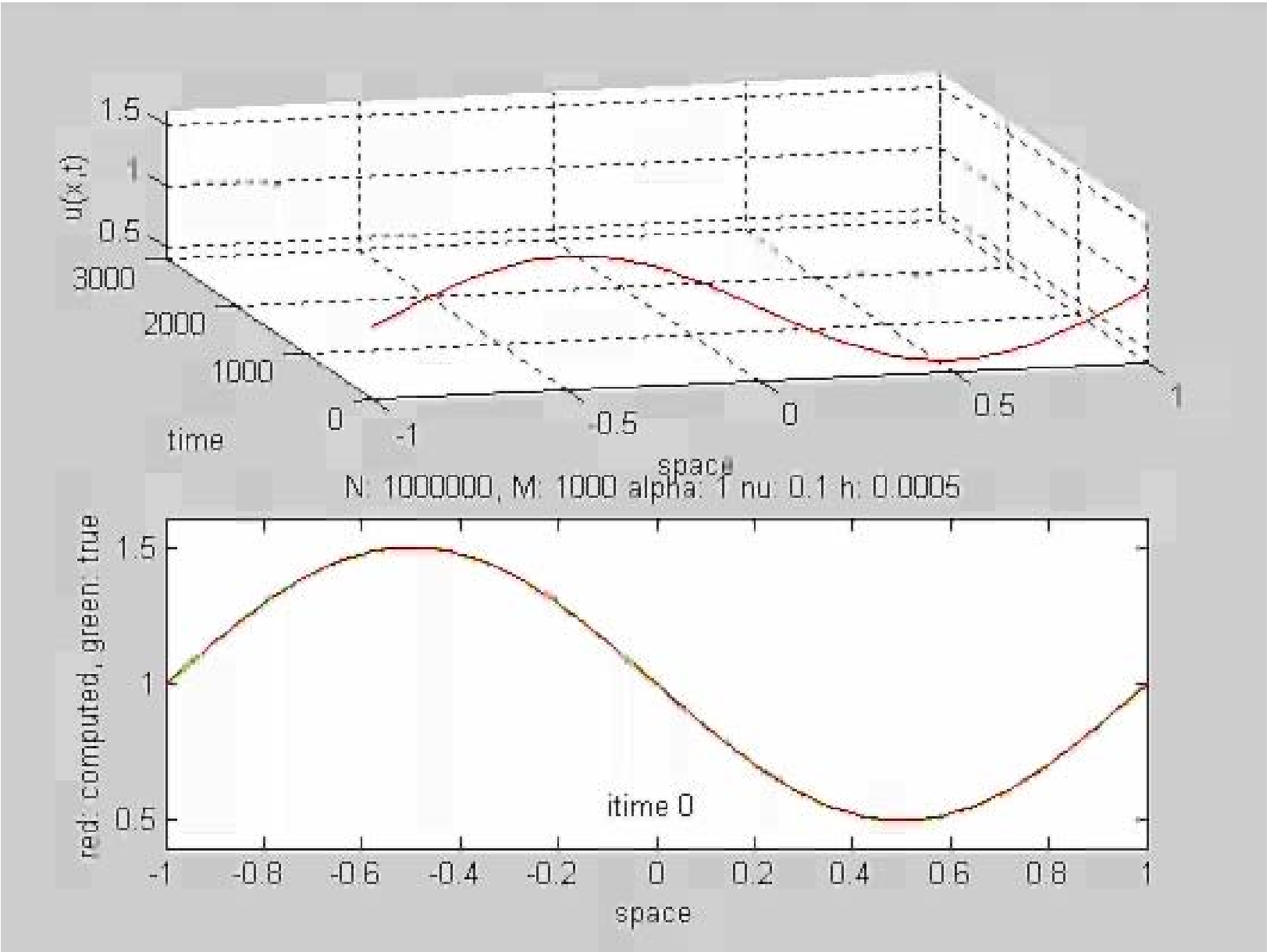
The "action" is going on at the intersection of the strips

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

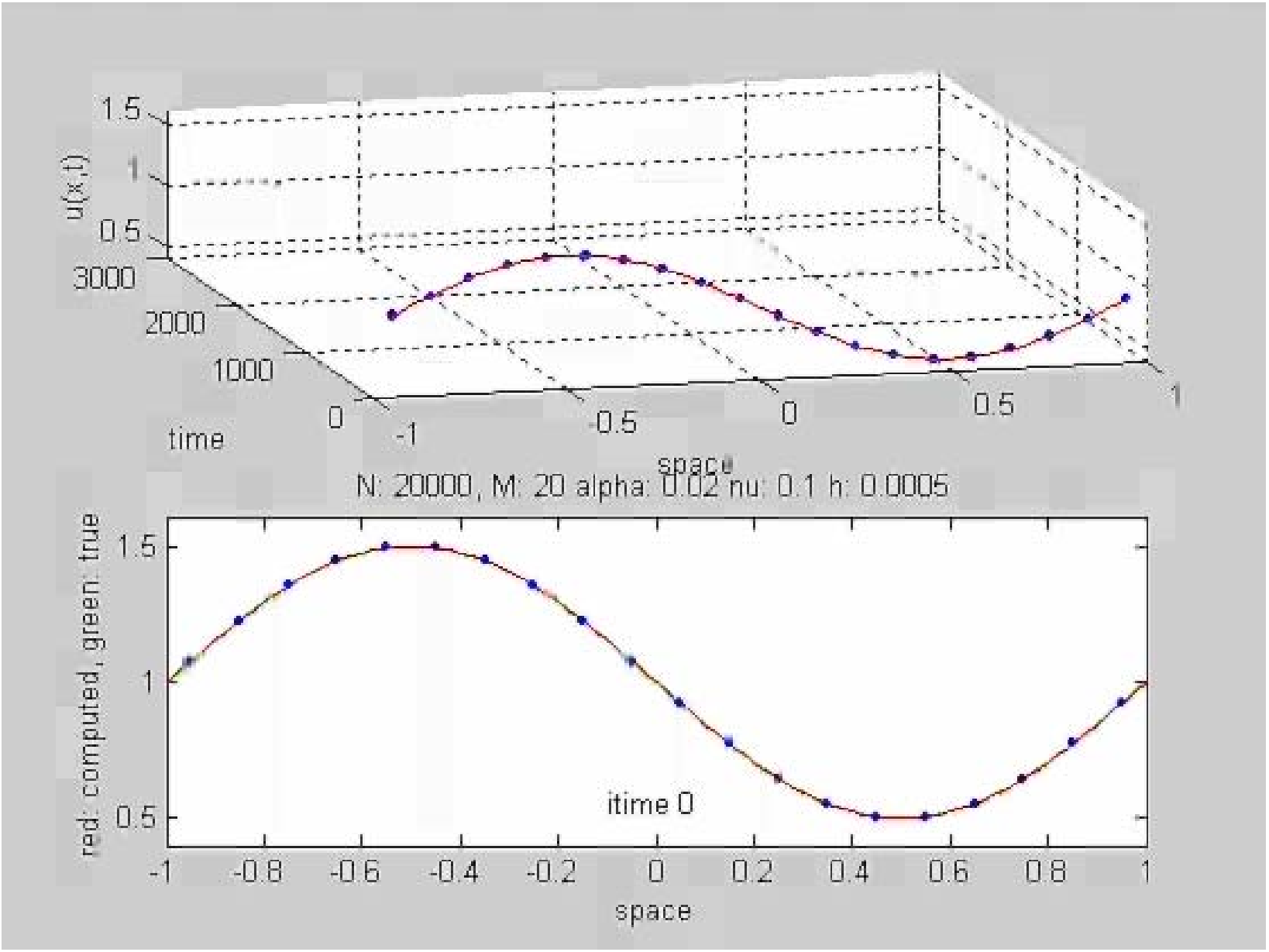
Example: Simulation of Viscous Burgers Equation

- Burgers equation is $$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}$$
- Can be simulated by particles as follows:
  - After each time step of length δt every particle is moved by a Gaussian noise of mean 0 and variance proportional to δt. This handles the second order x derivative.
  - Each particle is also moved to the right a distance proportional to the local density of particles. This handles the first order x derivative.
- The next two slides have movies showing (1) a particle simulation of the equation with periodic boundary conditions using 1,000,000 particles over the full region, and (2) a particle simulation using teeth occupying only 2% of the space (98% of the region is gaps) with 20,000 particles to do 1/50 of the total work. It also uses projective integration jumping over 2/3 of the time domain.

Burgers Equation by particle Monte Carlo full space and time



Burgers Equation by particle Monte Carlo with gap-tooth scheme and projective integration



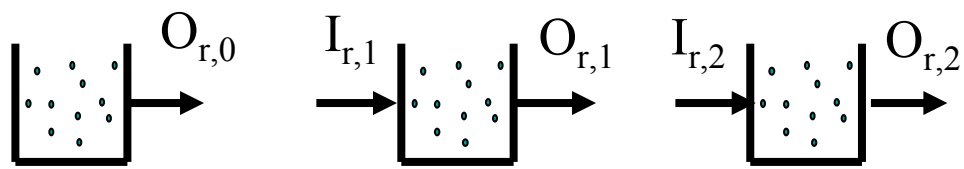
Concluding remarks

- We have discussed one form of Multiscale computing.
- It can be used to determine macroscopic behavior of systems for which we only know microscopic models.
- By its nature, the approach is well adapted to parallel computing because each tooth can be simulated on a different processor.
- It must be emphasised that, like much of multiscale computing, this is relatively new work so that there are still many issues to be worked out.

- The remaining slides are not part of the talk, but are here in case of certain types of questions.

Higher order-interpolation of internal boundaries is needed for higher-order problems:

Burgers equation requires second-order interpolation.



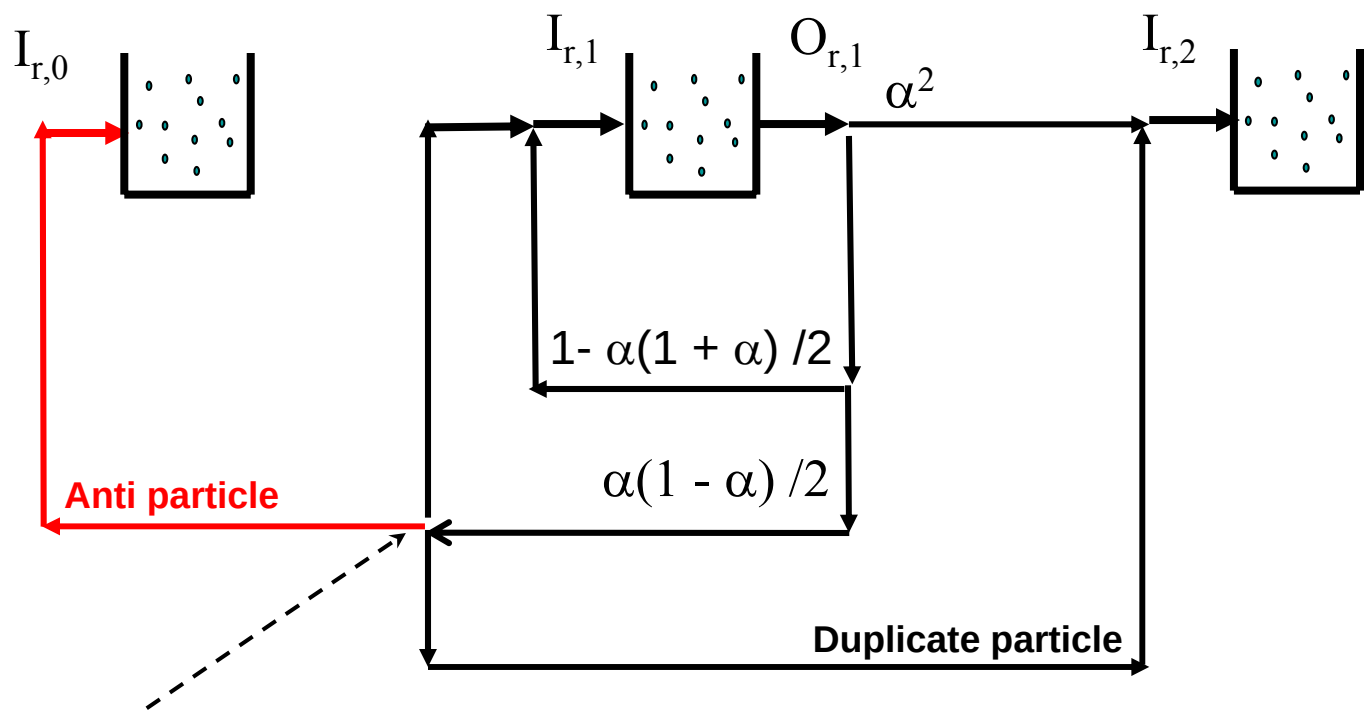
$$I_{r,i} = \frac{\alpha(1+\alpha)}{2} O_{r,i-1} + (1-\alpha^2) O_{r,i} - \frac{\alpha(1-\alpha)}{2} O_{r,i+1}$$

How to handle a negative number of particles?

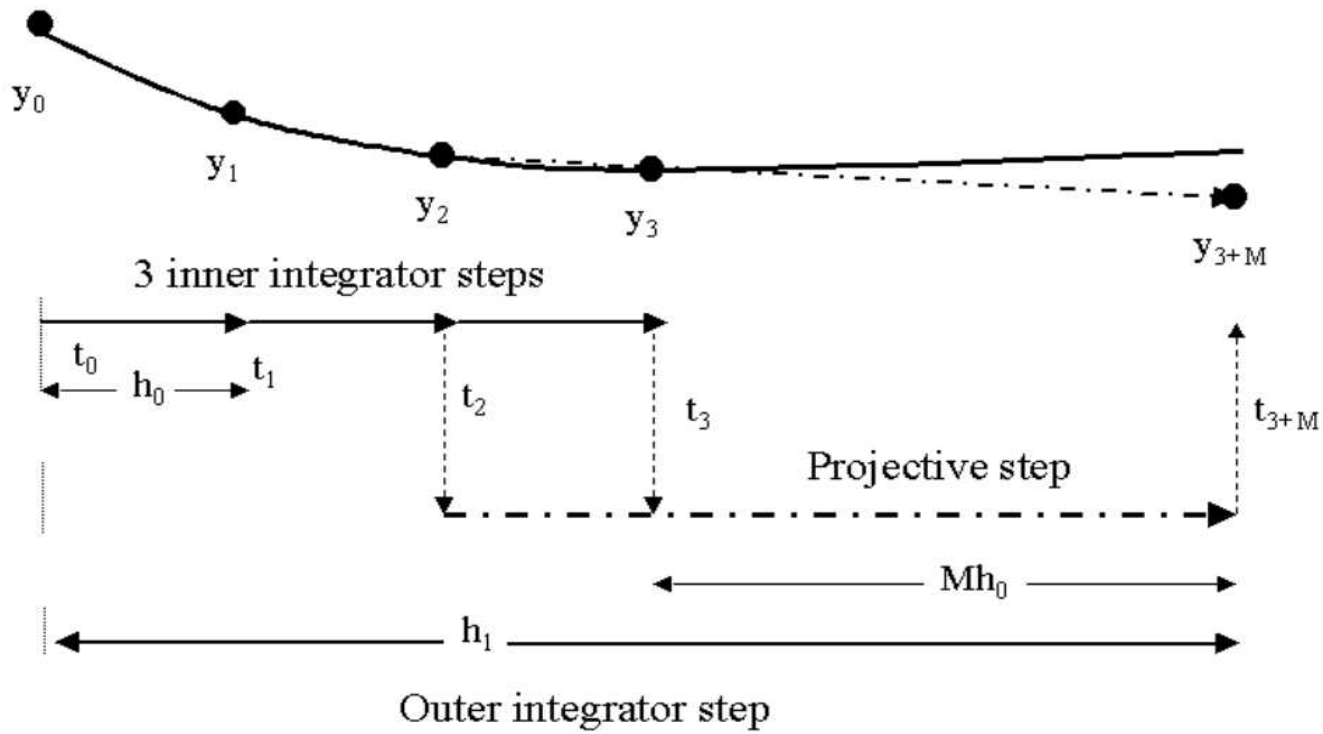
(Any higher-order interpolation formula must have some negative coefficients.)

Total “Flow” needed:

$-\alpha(1 - \alpha)/2$ $\quad$ $1-\alpha^2$ $\quad$ $\alpha(1 + \alpha)/2$



Particle “splits” into itself, a duplicate, and an “anti particle”



Projective Forward Euler Method - linear fit to last two points

Projective Integration

We suppose we have a one-step integrator $\Phi(y)$ that computes $y(t+h)$ from $y(t)$ for a the differential equation $y' = f(y)$

Starting from $y_0 = y(t_0)$ we apply this k times to form $y_1, y_2, \ldots, y_k$. Now we apply it one more time and use y_k and y_{k+1} to estimate the derivative.

We then use this derivative to integrate (extrapolate) forward a distance Mh. It can be written as:

$$\mathbf{y_{k+1+M} = (M+1)y_{k+1} - My_k}$$

This is the *Projective Forward Euler* method

Stability analysis – simple case:
Projective **F**orward **E**uler (**PFE**)

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

Assume that one step of the supplied integrator $\Phi(y)$ (the *inner integrator*) has an amplification of $\rho(h\lambda)$

- for Forward Euler $\rho(h\lambda) = 1+h\lambda$
- for an "exact" integrator $\rho(h\lambda) = \exp(h\lambda)$

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

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

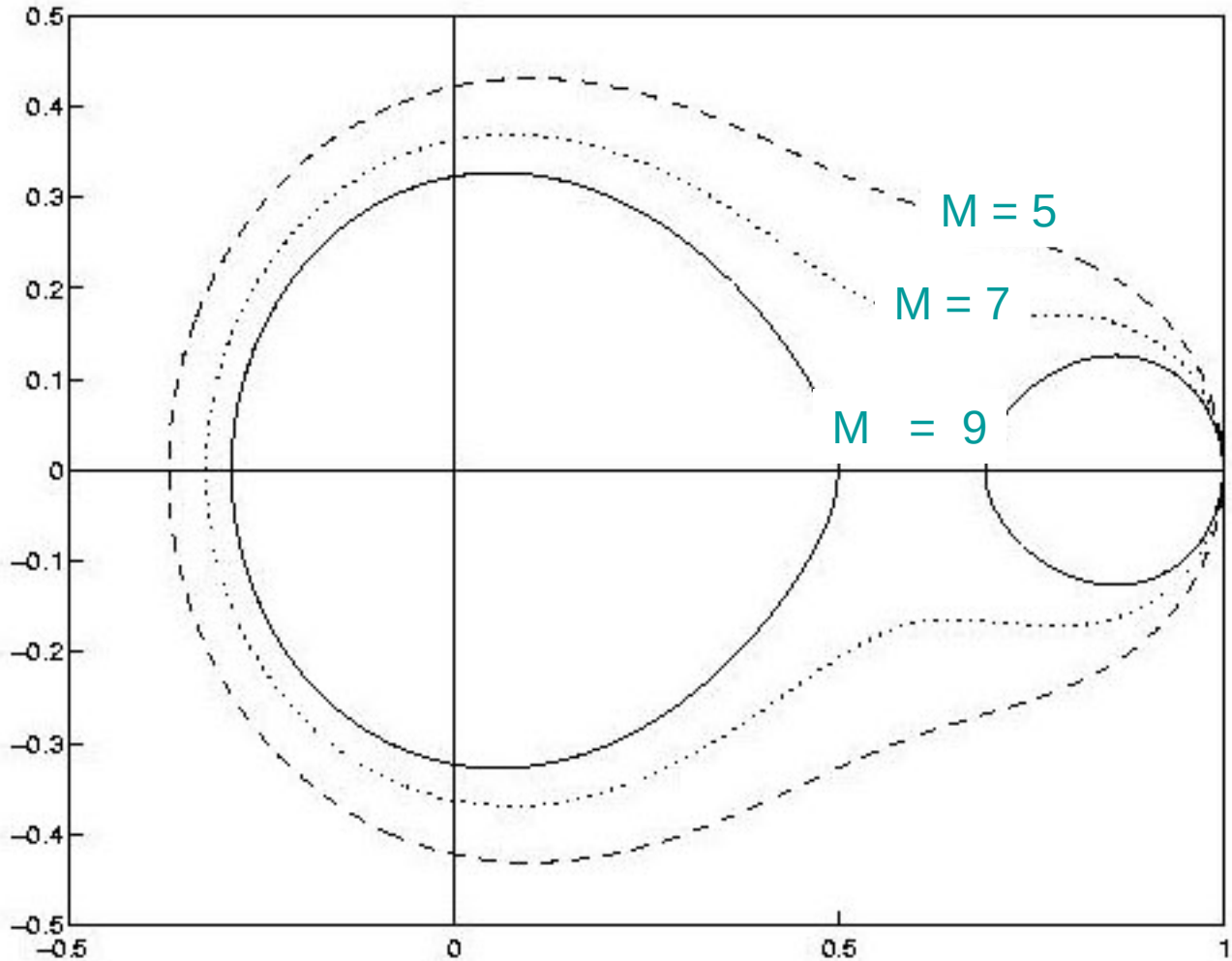
which is $\rho^k[1 + (M+1)\, h\lambda]$ if inner is Forward Euler

Region of absolute stability: Set of ρ such that $|\sigma| \leq 1$.

(Note: we express stability in terms of ρ making σ independent of the form of the inner integrator - which we may not know.)

Compute stability region by finding all ρ such that $|\sigma| < 1$

The region of absolute stability of takes one of two forms:

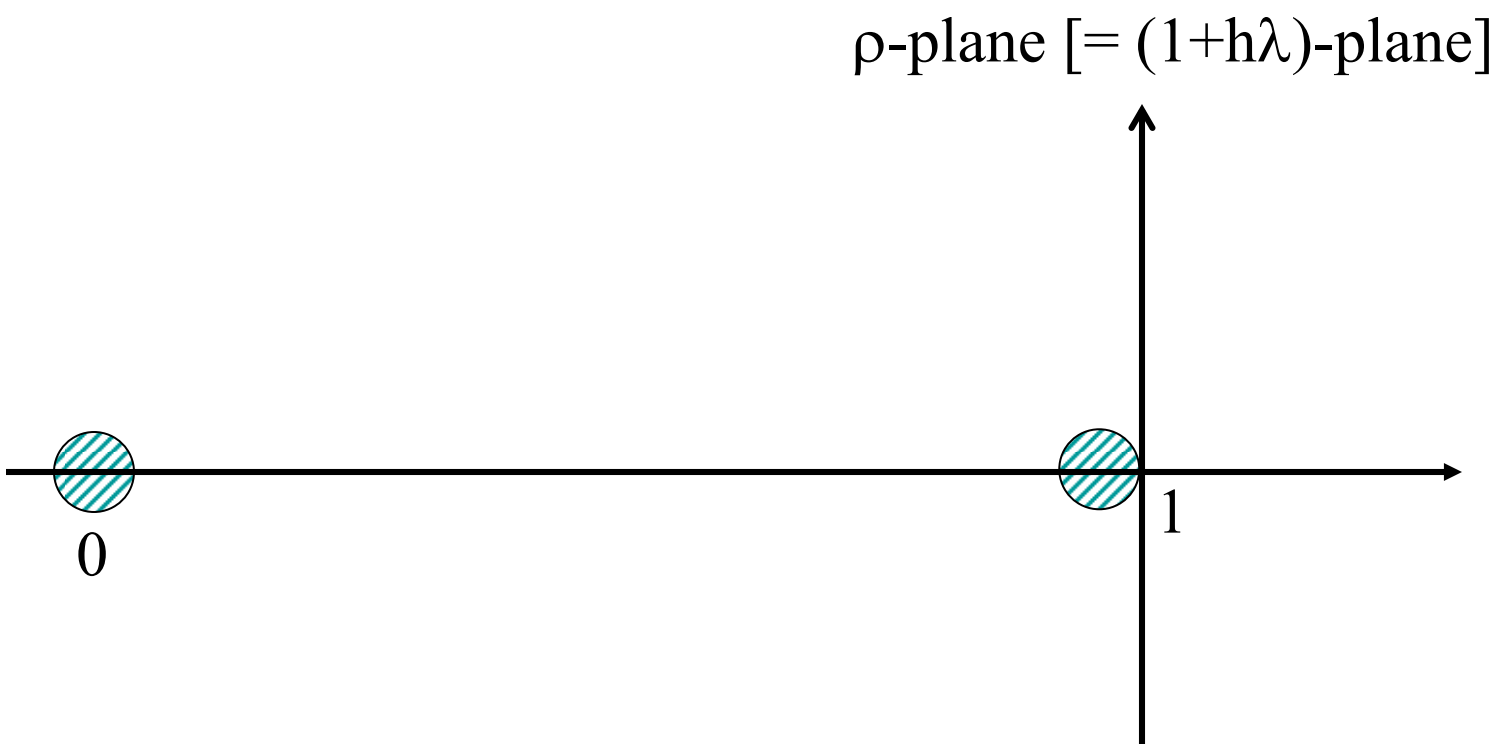


ρ-plane plot $k = 2$, $M = 5, 7,$ and 9

As M increases, change from one region to two. This happens when $M/k \approx 3.6$

Note: $\rho = 1+h\lambda$ if inner integrator is Forward Euler this is usual stability plot.

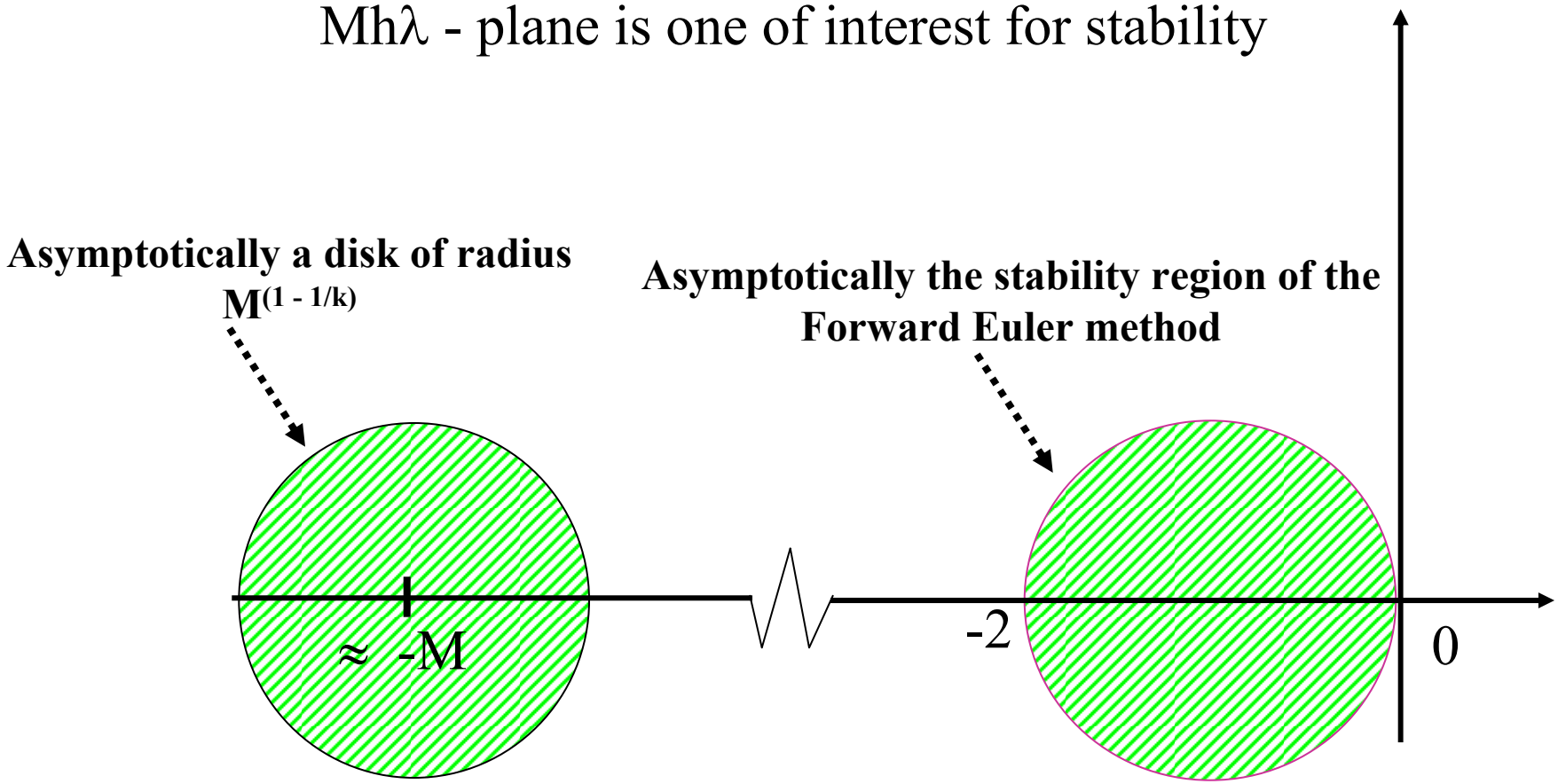
If M is large:



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

Consider large M: Outer Integrator step $\approx Mh\lambda$:

$Mh\lambda$ - plane is one of interest for stability



Telescoping Projective Methods
(handling multiple clusters of eigenvalues)

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) Let the inner integrator be Φ_0
First level outer integrator is Φ_1 e.g., for Projective Euler

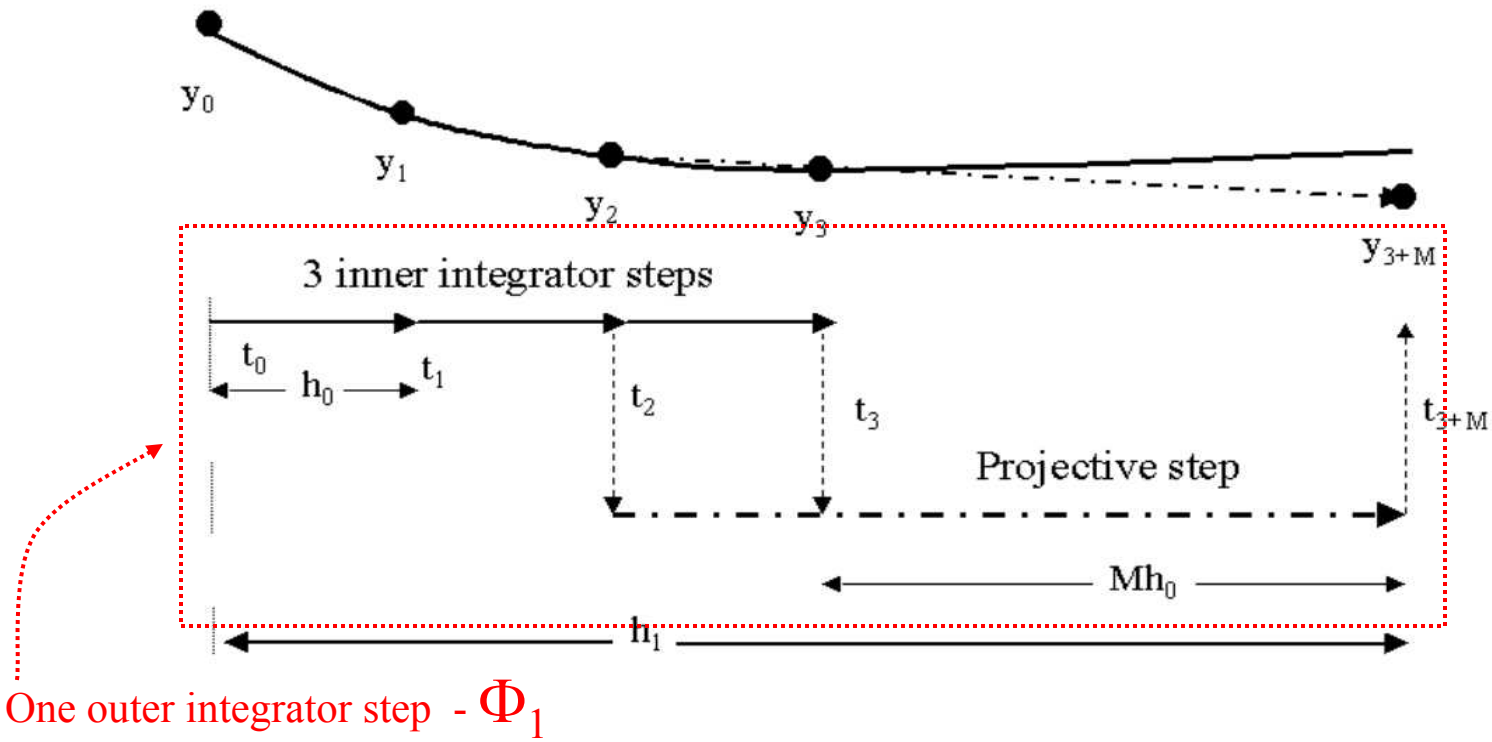
$$\Phi_1 = \Phi_0^k((M+1)\,\Phi_0 - M)$$

Projective integrator based on Φ_i is Φ_{i+1}

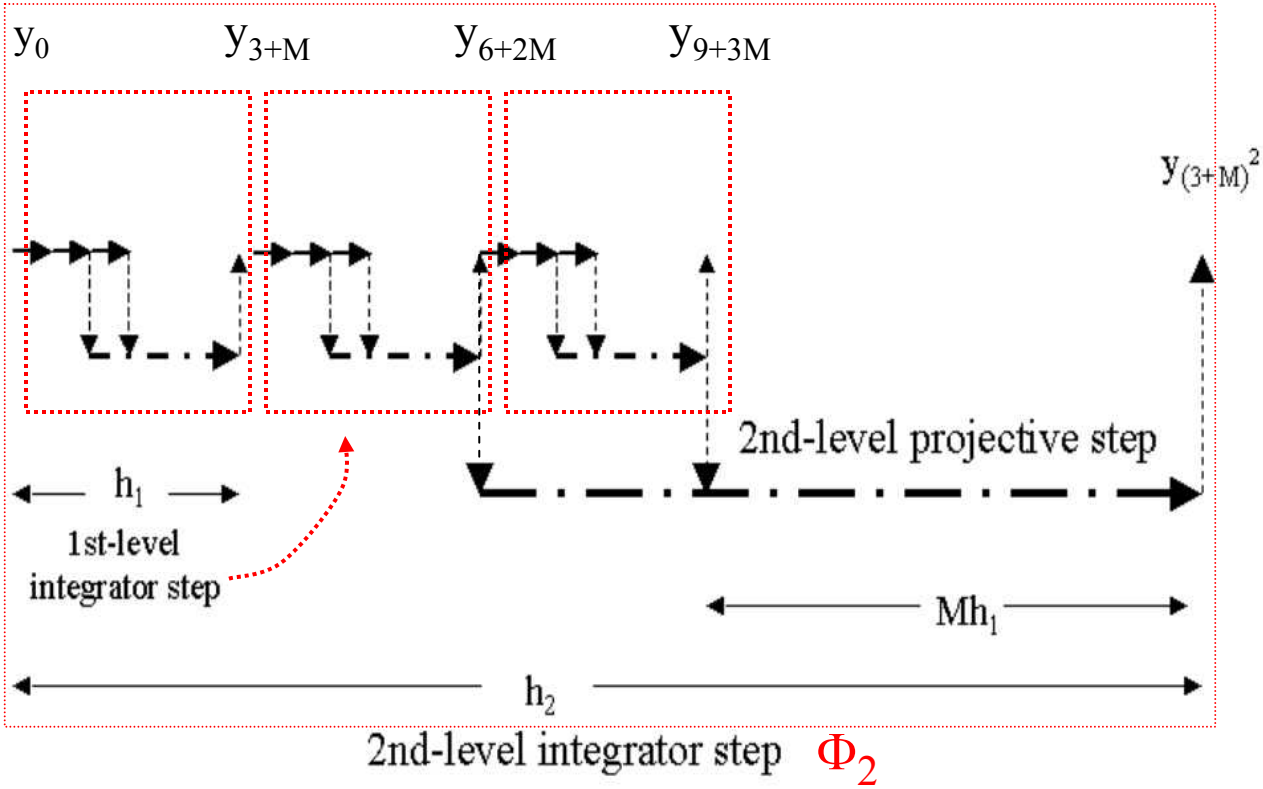
$$\Phi_{i+1} = \Phi_i^k((M+1)\,\Phi_i - M) \quad \text{for PFE (if k and M are fixed)}$$

It uses step size $h(k+q+M)^{i+1}$ - growing exponentially, or, more generally,

$$h_{i+1} = (k_i + q_i + M_i)h_i$$



Projective Forward Euler Method - linear fit to last two points



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

Let us consider the stability of Telescopic methods.

No need to use same method at each level, or same k and M at each level, but to illustrate concepts we will start by considering the use of (projective) forward Euler at all levels.

In practice, we will use first-order methods at all levels except for the outermost level since at the inner levels the step size/method is determined by stability only. At the outer level the step size/method must be chosen for accuracy.

Telescoping Projective Methods:

What is stability? Consider Projective Forward Euler.

Let amplification of level i integrator be σ_i We have

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

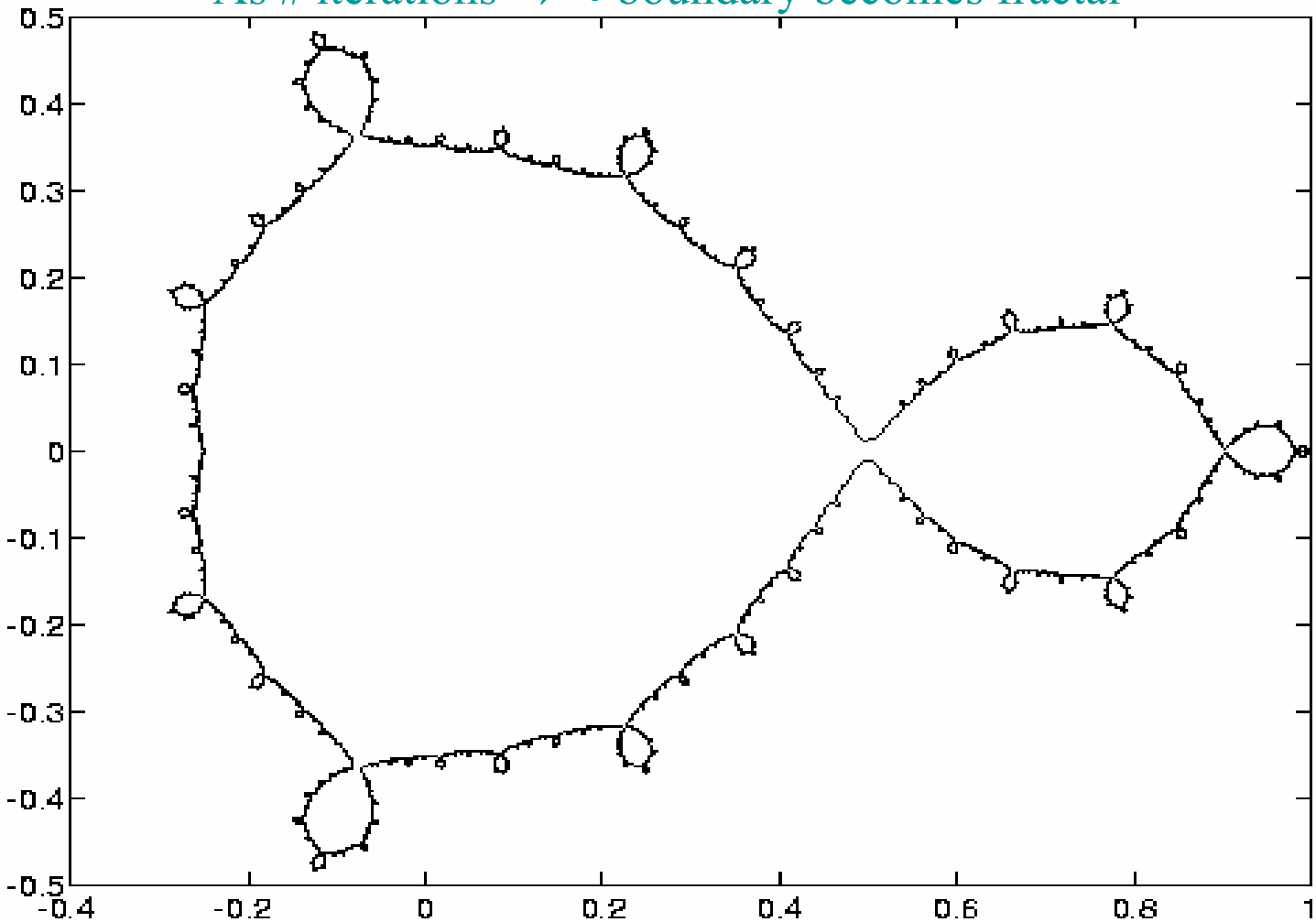
Stability region is set of σ that remain in unit disk (or, remain bounded) in iteration

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

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. Outer stepsize is 60,466,176h_0

As # iterations $\rightarrow \infty$ boundary becomes fractal



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

In this case, the stability boundary touches the real axis 10 times.

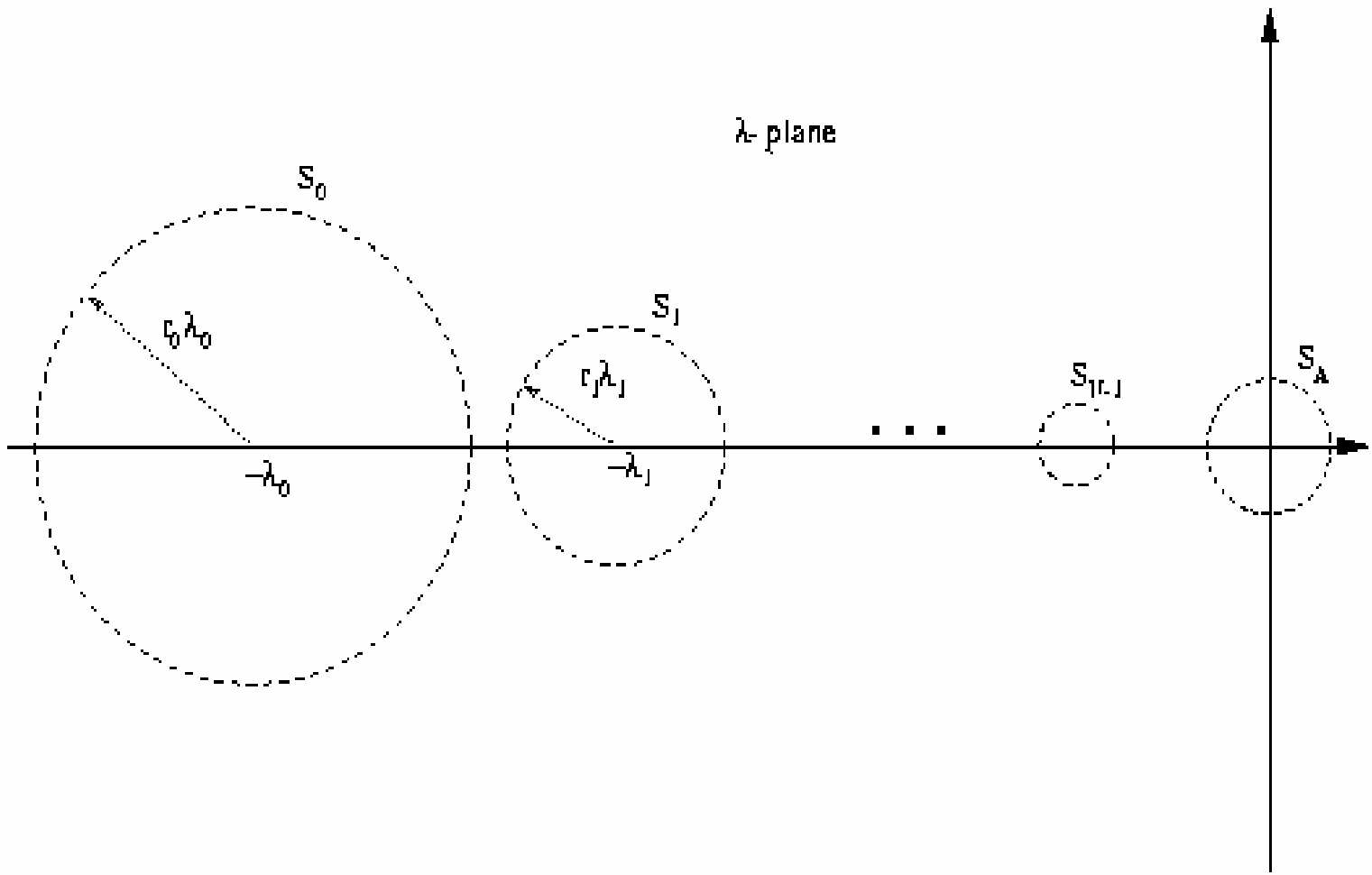
Use of a slightly smaller M would place [-β,1) in the interior of the stability region.

We can show that for any M there is a k_0 such that for all $k \geq k_0$ the stability region includes the negative axis from 0 to 1 in the ρ-plane in the case of the Forward Euler method.

In some ways, these are like high-stage number Runge Kutta Methods which have been used to extend the region of stability. (The previous slide's 10-level projective integrator is a 59049-stage RK method!)

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:



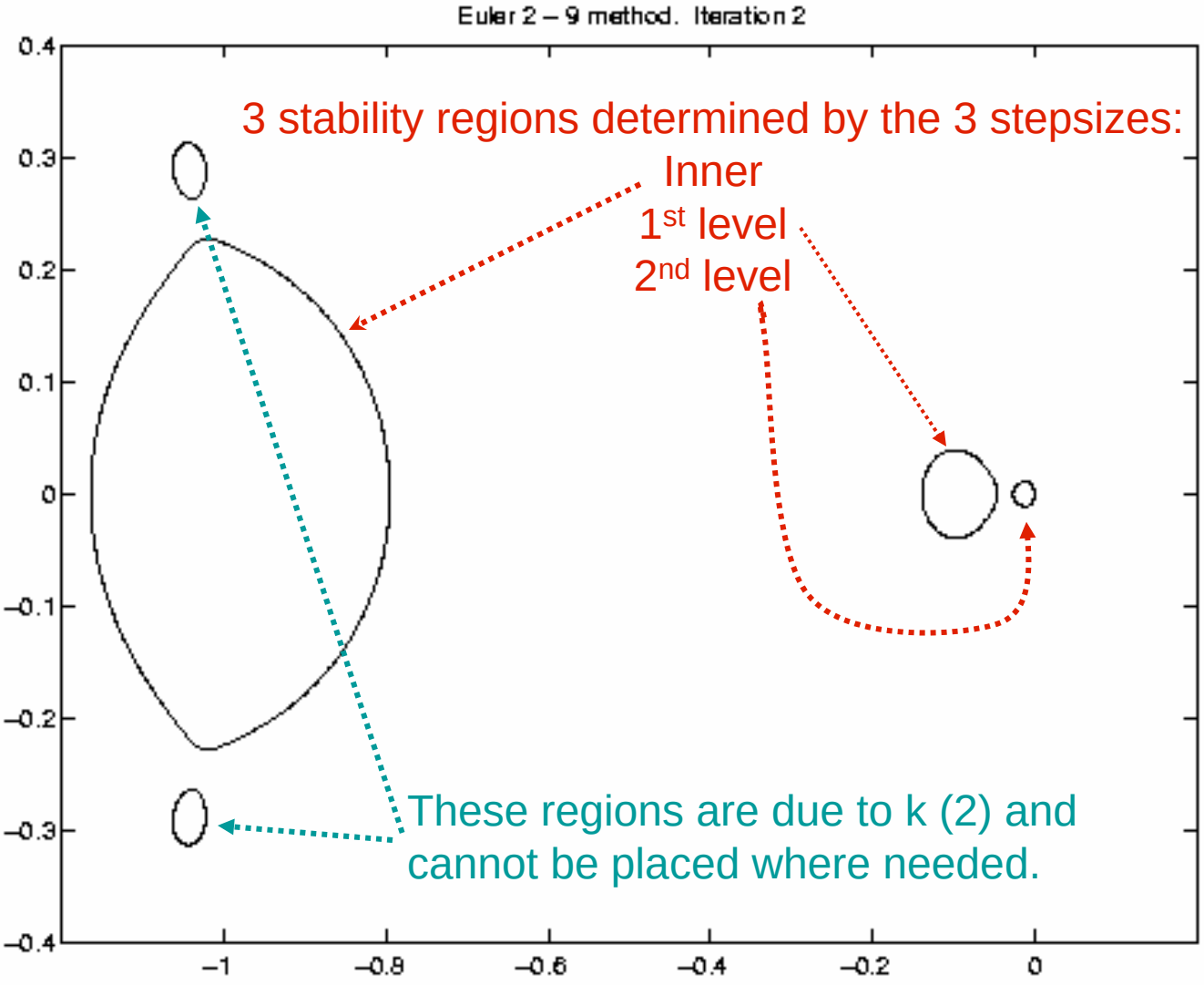
… by choosing an inner step size of $1/\lambda_0$ and then using an effective step size at the outer levels of projective integration of $1/\lambda_i$ for the i-th level we can choose values for k_i (the number of steps at the (i -1)-st level before projecting at the i-th level to achieve stability over these disks.

The step size at the i-th level is defined as

$$h_i = h_{i-1}(k_{i-1} + M_{i-1} + 1)$$

A particular case with h = 1, 11, and 121 is shown in the next slide:

A two-level PFE2-9 method



In effect, each successive stepsize damps eigenvalues of the order of the inverse of that stepsize.

Thus, if we know where the “stiff” eigenvalues are, we can select a sequence of stepsizes to damp them.

In a sense, this method is a little like using a Chebshev type iteration to remove eigencomponents in decreasing order of size, except that it is spread out over many steps. (If the problem is highly non-linear in a way that cause the eigendirections to change rapidly, more iterations (that is, inner steps) are needed, and the method becomes less efficient.

NOTE

A telescopic method is NOT the same as using a collection of different step sizes. The telescopic method can be designed so that the number of steps at any given level before a projective step is determined solely by the ratio of the step sizes at the two adjacent levels, independent of the number of levels of telescoping.

If, on the other hand, just a collection of step sizes are used, more smaller steps are needed as the size of the largest step increases.

What about complex eigenvalues?

Use some sort of two-stage method (RK-like) which will have $\rho(h\lambda) = 0$ when $h\lambda$ has complex conjugate values (if used as the inner integrator) or $\sigma(\rho) = 0$ when ρ has complex conjugate values for intermediate stage integrators..

For example,

$$y_{n+1} = y_n + (1-\beta)hy_n^{'} + \beta hy^{'}(y_n + \alpha hy_n^{'})$$

This has

$$\rho(h\lambda) = 1 + h\lambda + \alpha\beta(h\lambda)^2$$

which has roots at

$$h\lambda = -1/(2\alpha\beta) \pm i\sqrt{4\alpha\beta - 1}/(2\alpha\beta)$$

and are complex for $r = \alpha\beta > 1/4$. (Second order if $r = 1/2$.)

Similar technique can be used for intermediate level projective step.

Alternatively (and more simply) one can use k+2 inner steps and then do a projective step using the last three points in:

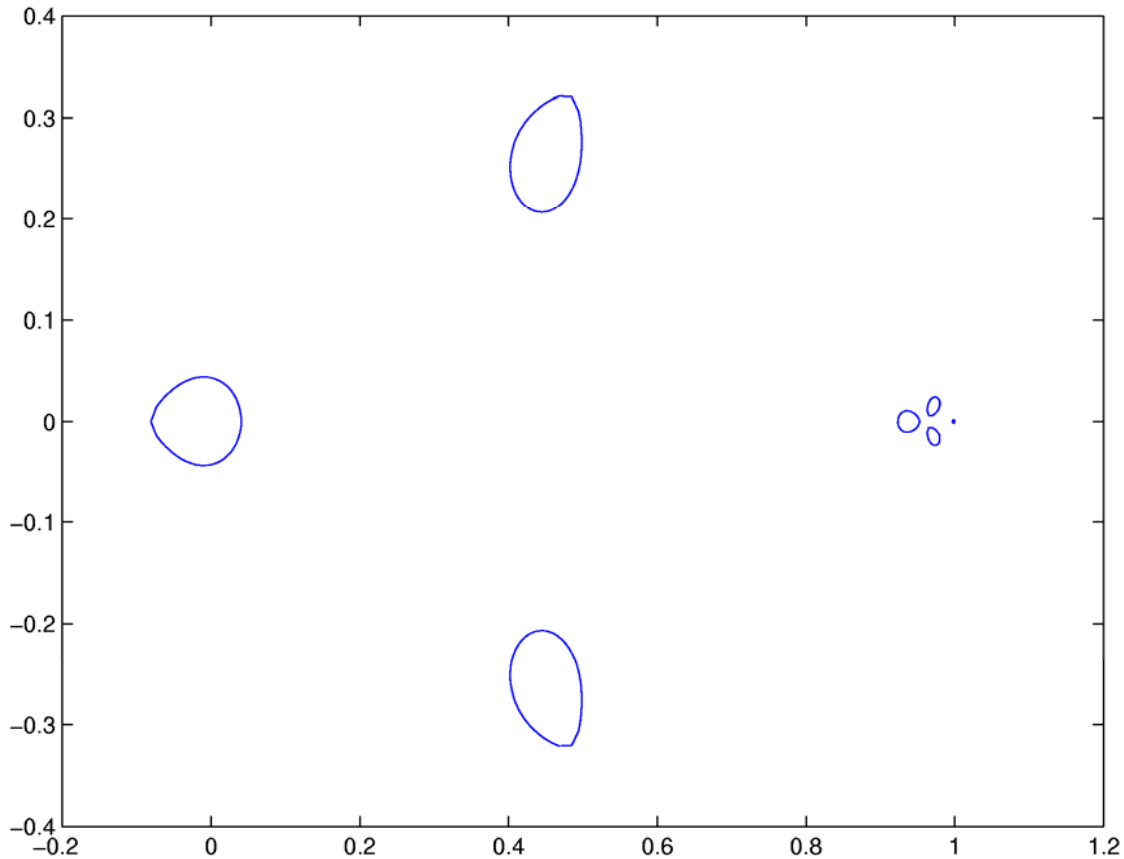
$$y_{k+2+M} = rM^2 y_{k+2} + (M-2rM^2) y_{k+1} - (M-1-rM^2) y_k$$

to achieve a stability polynomial of the form

$$\sigma = \rho^k [1 + M(\rho-1) + rM^2 (\rho-1)^2]$$

This places stability regions around the complex conjugate eigenpair given on the last slide.

With this technique we can place regions of stability around any desired sets of locations in the complex plain, as shown in the next slide.



Stability plot of 4-level projective method in the $\rho = 1+h_0\lambda$ plane. The sizes of steps at each level are h_0, $4h_0$, $20h_0$, $80h_0$, and $880h_0$. The outermost is Projective Forward Euler (k=1, M=9), so the right-most region - a dot in the figure at (0,1) is near the stability region for the forward Euler with step size $880h_0$. The first and third levels used the method on the last slide with r=0.3 and r=0.35 respectively (with k=1, M=3) to place stability regions around $0.44\pm0.25i$, and $0.976\pm0.015i$ respectively. The second level was PFE with k=1, M=3 placing a region near 0.95. Larger values of k would make the regions (except for the rightmost) larger.