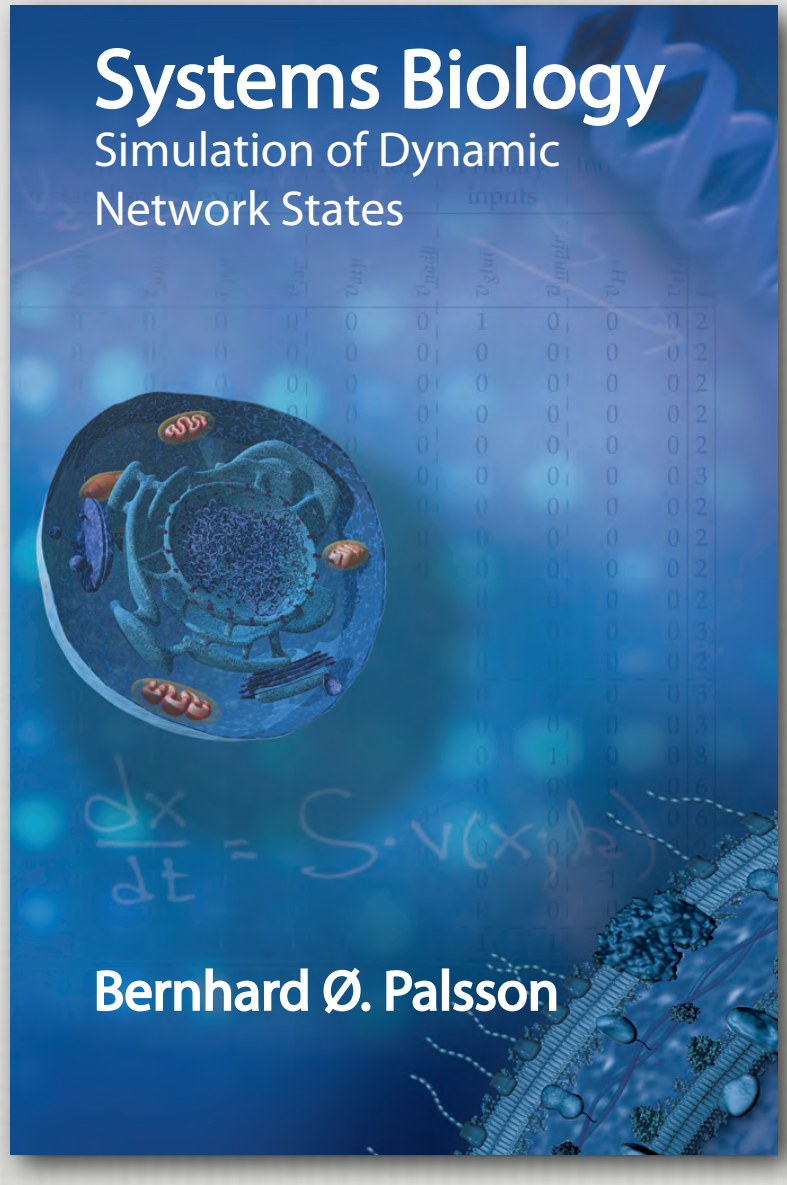


Lecture #3

Simulating
Dynamic Mass
Balances

Outline

- Basic simulation procedure
- Graphical representation
- Post-processing the solution
- Multi-scale representation

BASIC PROCEDURE

Dynamic Simulation

1. Formulate the mass balances: $\frac{dx}{dt} = Sv(x;k)$

2. Specify a) the numerical values for **k** and b) ICs

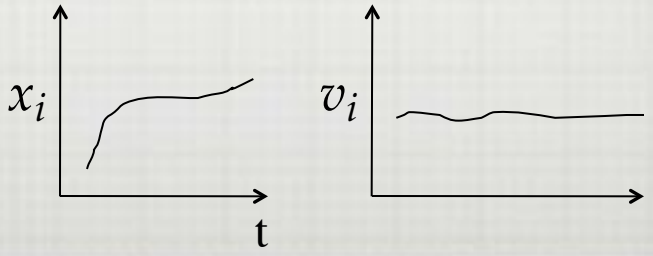
$$k_i = \text{value; I.C. } x_{io} = x_i(t=0)$$

3. Obtain the numerical solution

Run a software package

4. Analyze the results

Graphically:



Typical Workbook:
details to follow

STEP 1

Set up the equations

Specify parameters

STEP 2

STEP 3

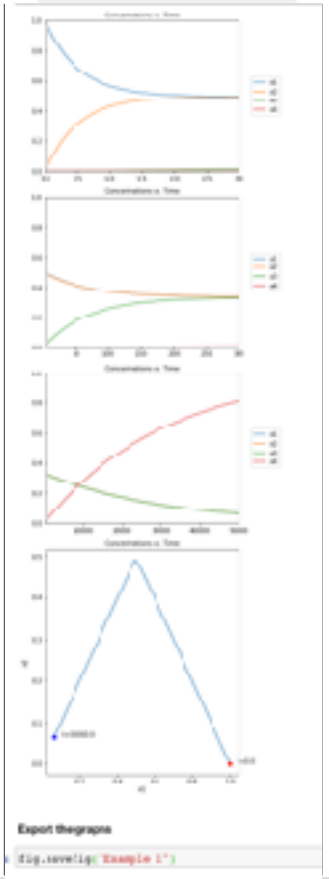
Solve

Specify graphical output

STEP 4

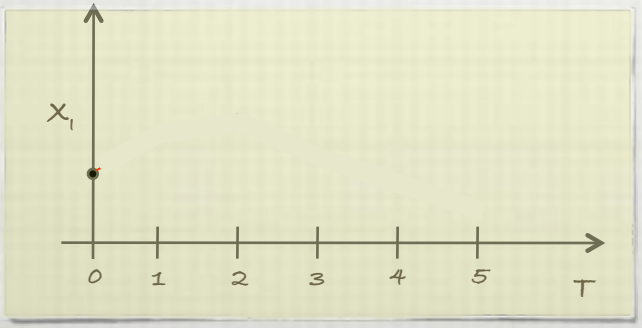
Plot and export

RESULTS



The Results of Simulation: a table of numbers

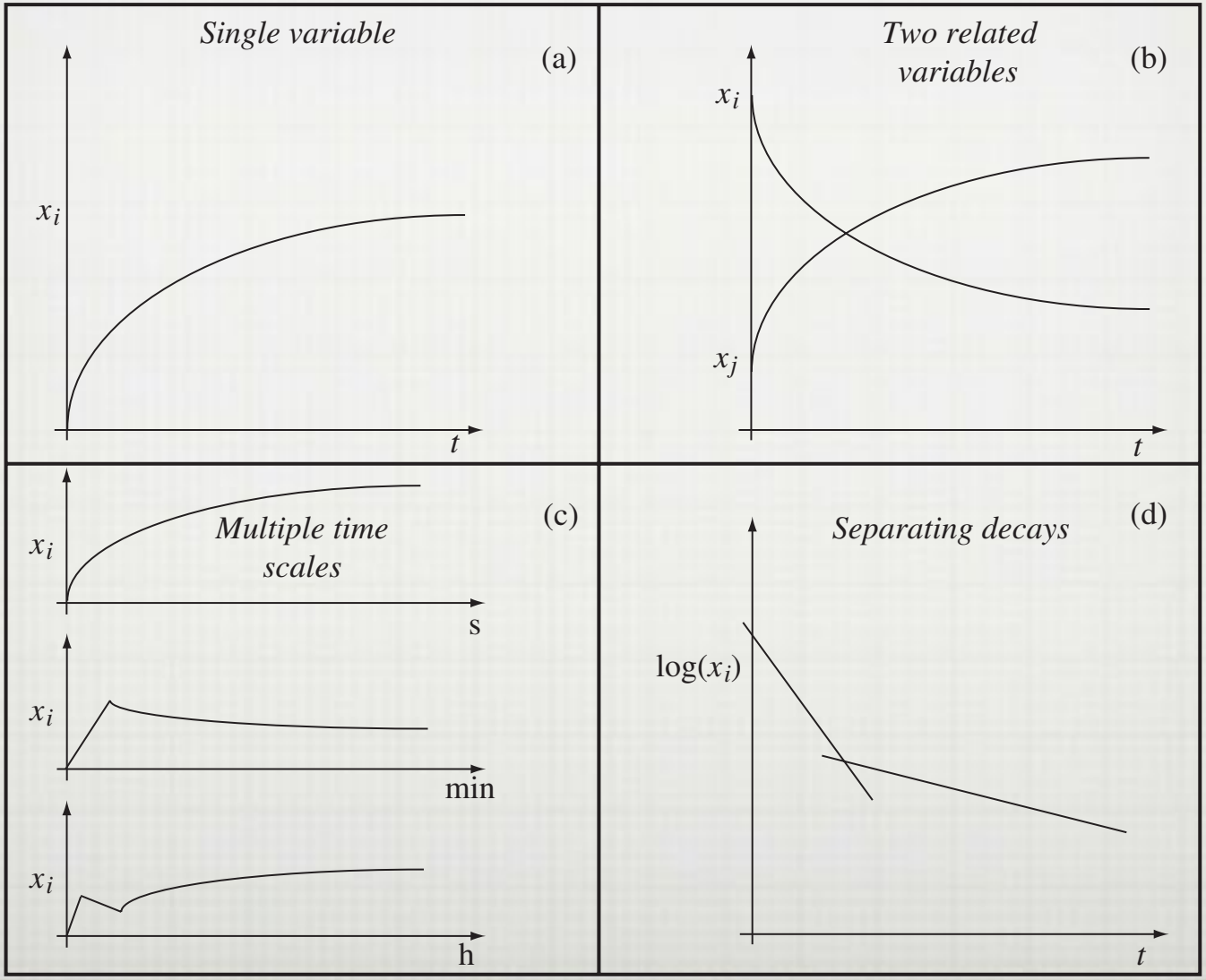
t	x_1	x_2	…	x_m
t_1	$x_1(t_1)$	$x_2(t_1)$		$x_m(t_1)$
t_2	$x_1(t_2)$	$x_2(t_2)$		$x_m(t_2)$
t_3	$x_1(t_3)$	$x_2(t_3)$		$x_m(t_3)$
⋮	⋮			⋮
t_n	$x_1(t_n)$	$x_2(t_n)$		$x_m(t_n)$
t_{n+1}	$x_1(t_{n+1})$	$x_2(t_{n+1})$		$x_m(t_{n+1})$
t_{n+2}	$x_1(t_{n+2})$	$x_2(t_{n+2})$	…	$x_m(t_{n+2})$



Can correlate variables, r^2
- off-set in time
- on a particular time scale

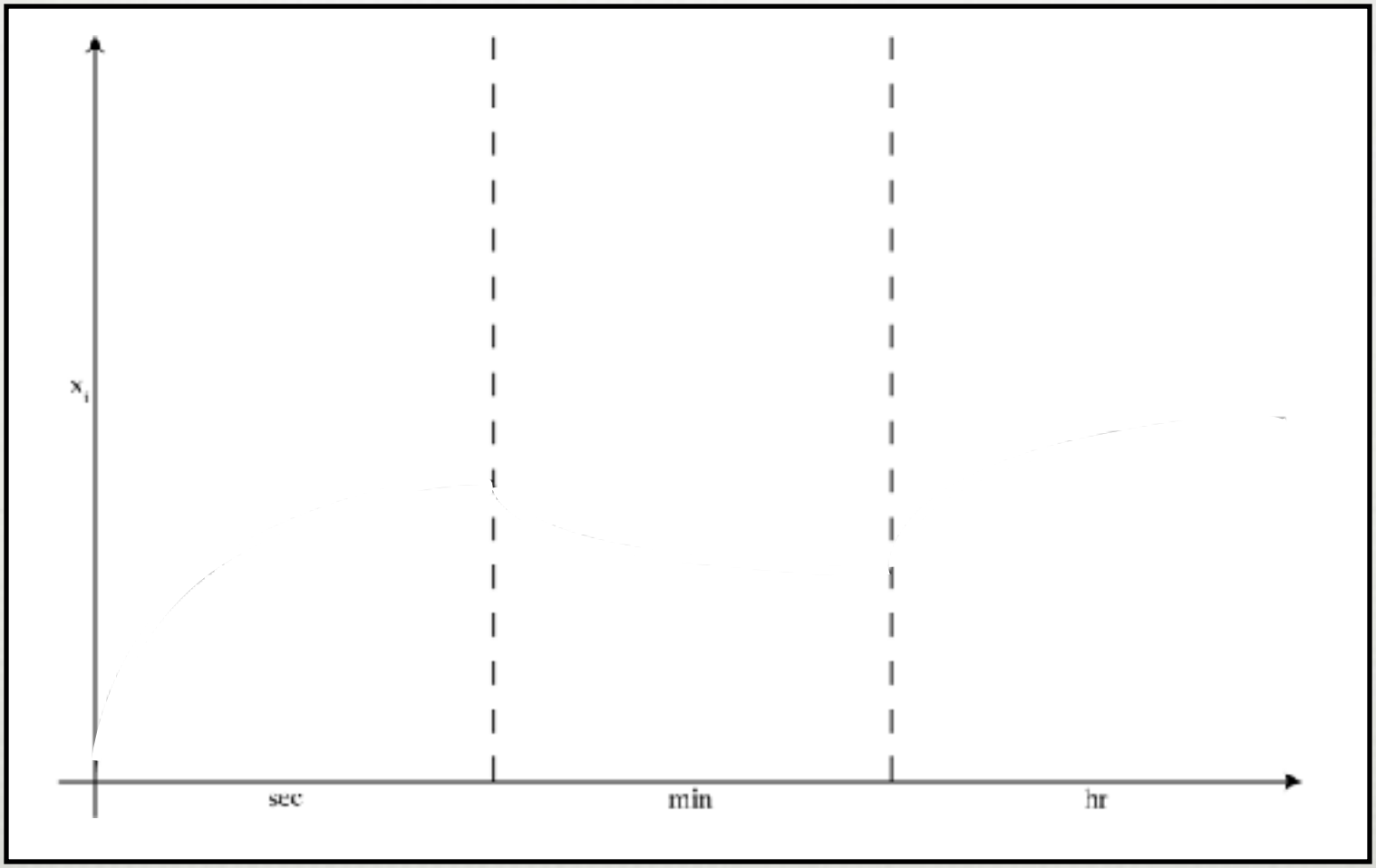
GRAPHICAL REPRESENTATION

Example Graphical Representation of the Solution, **x**(t)

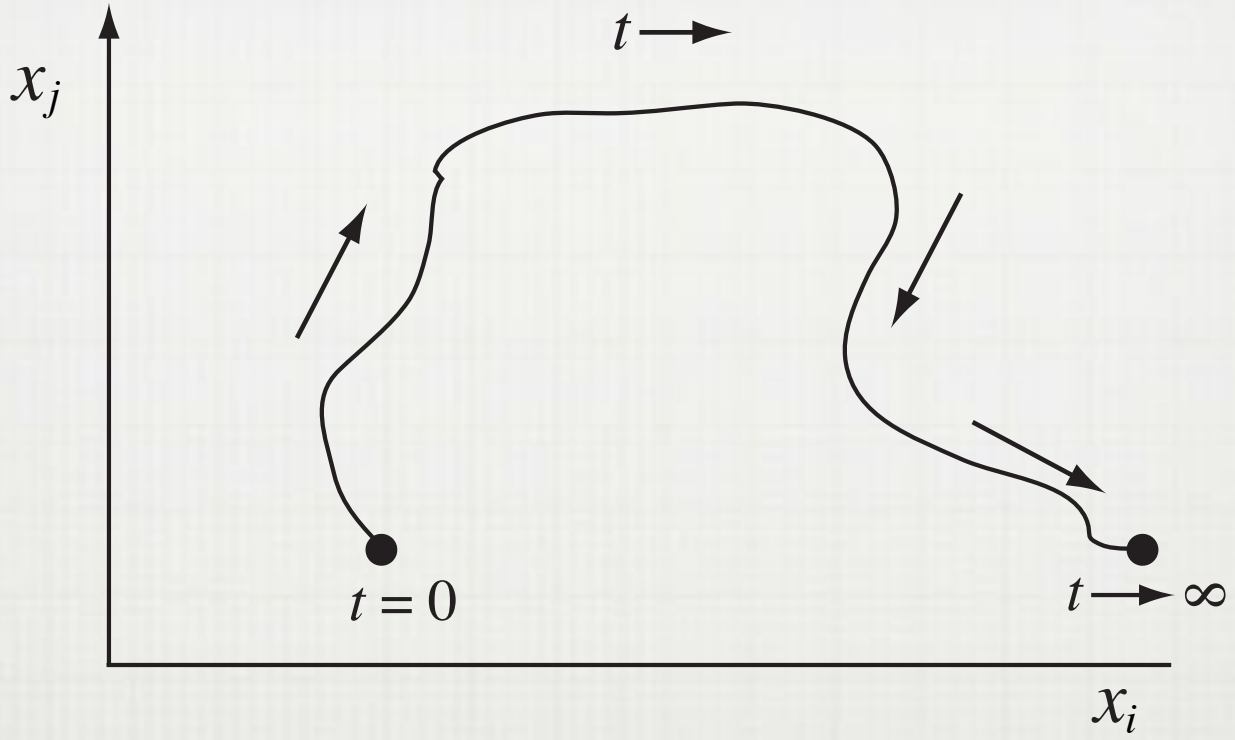


Multi-time scale representation:

segmenting the x-axis

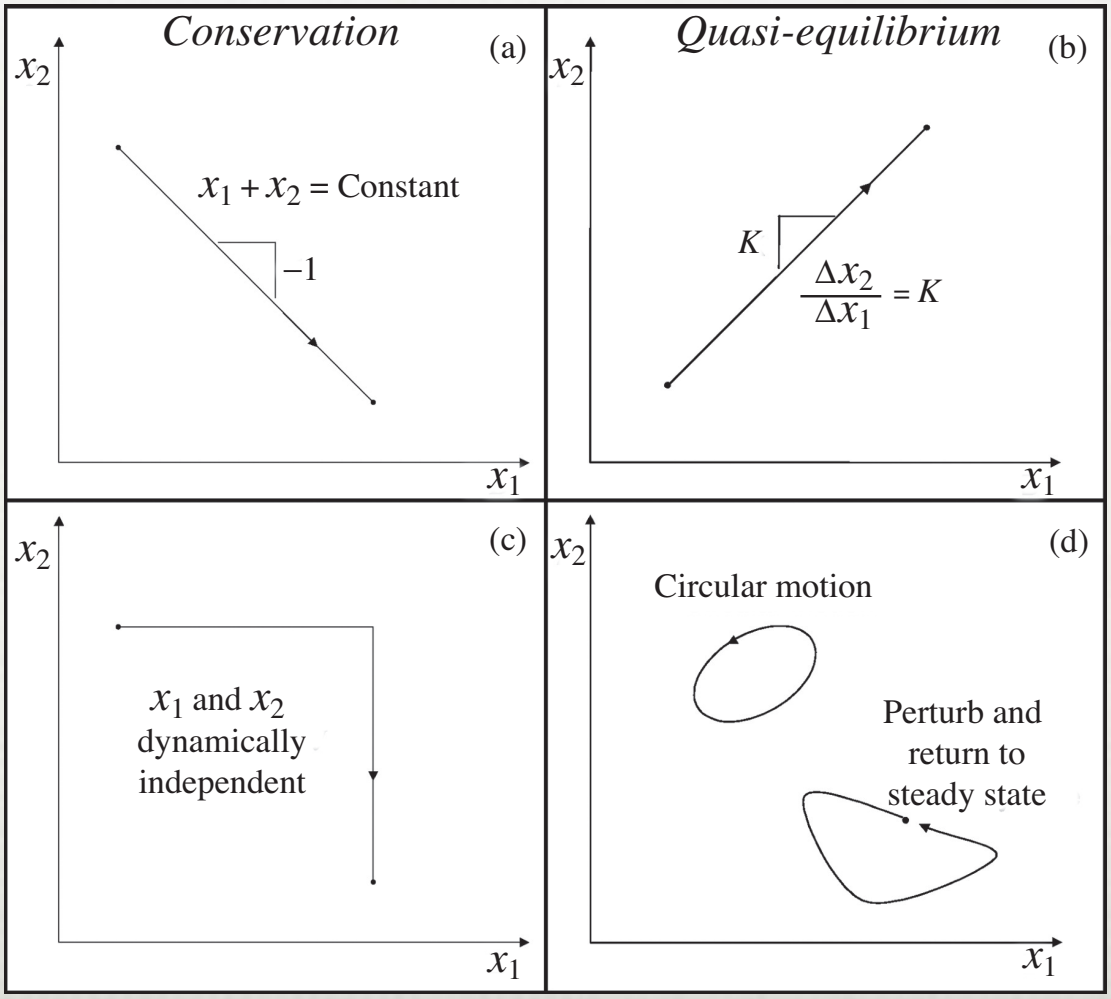


Dynamic Phase Portrait



Can also plot fluxes $v_i(t)$ and pools $p_i(t)$ in the same way; this is a very useful representation of the results

Characteristic "Signatures" in the Phase Portrait

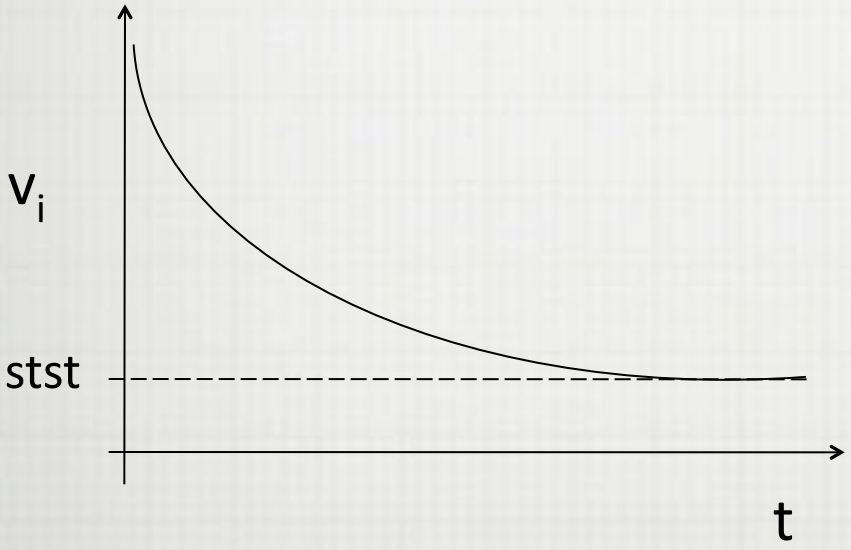


POST-PROCESSING THE SOLUTION

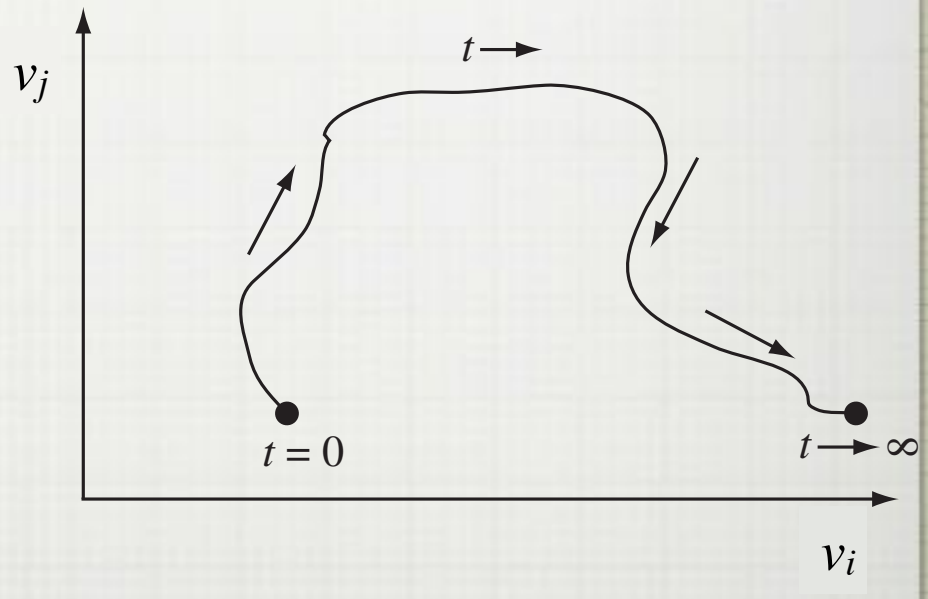
Post-processing the Solution

1. Computing the fluxes

$$v(t) = v(x(t))$$



dynamic response

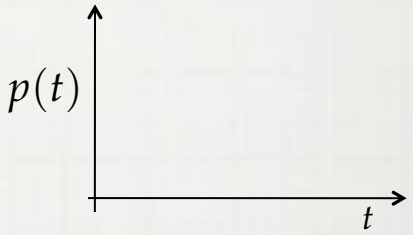


phase portrait

Post-processing the Solution

2. Forming pools; two examples

Compute "pools": $p(t) = Px(t)$ then plot



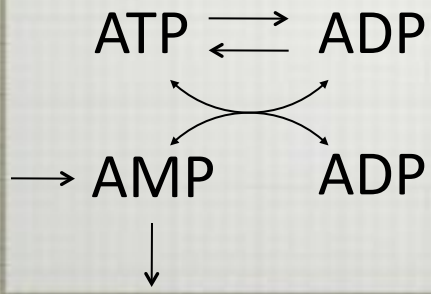
Example #1

$x_1 \rightleftharpoons x_2$

$K_{eq} = 1$

$$p(t) = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1 - x_2 \end{pmatrix} = \begin{pmatrix} \text{total mass} \\ \text{dis-equil} \end{pmatrix}$$

Example #2



$$p(t) = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \text{ATP} \\ \text{ADP} \\ \text{AMP} \end{pmatrix} = \begin{pmatrix} \text{ATP+ADP} \\ \text{2ATP+ADP} \\ \text{ATP+ADP+AMP} \end{pmatrix}$$

Post-processing the Solution

3. Computing auto correlations

Example 1

$$\sigma_{1,m} = \frac{1}{n}\sum_{k=1}^{n}(x_1(k) - \overline{x}_1)(x_m(k) - \overline{x}_m)$$

n=3
first segment

Example 2

$$\sigma_{1,m} = \frac{1}{n}\sum_{k=1}^{n}(x_1(k) - \overline{x}_1)(x_m(k) - \overline{x}_m)$$

n=m
entire range

Example 3

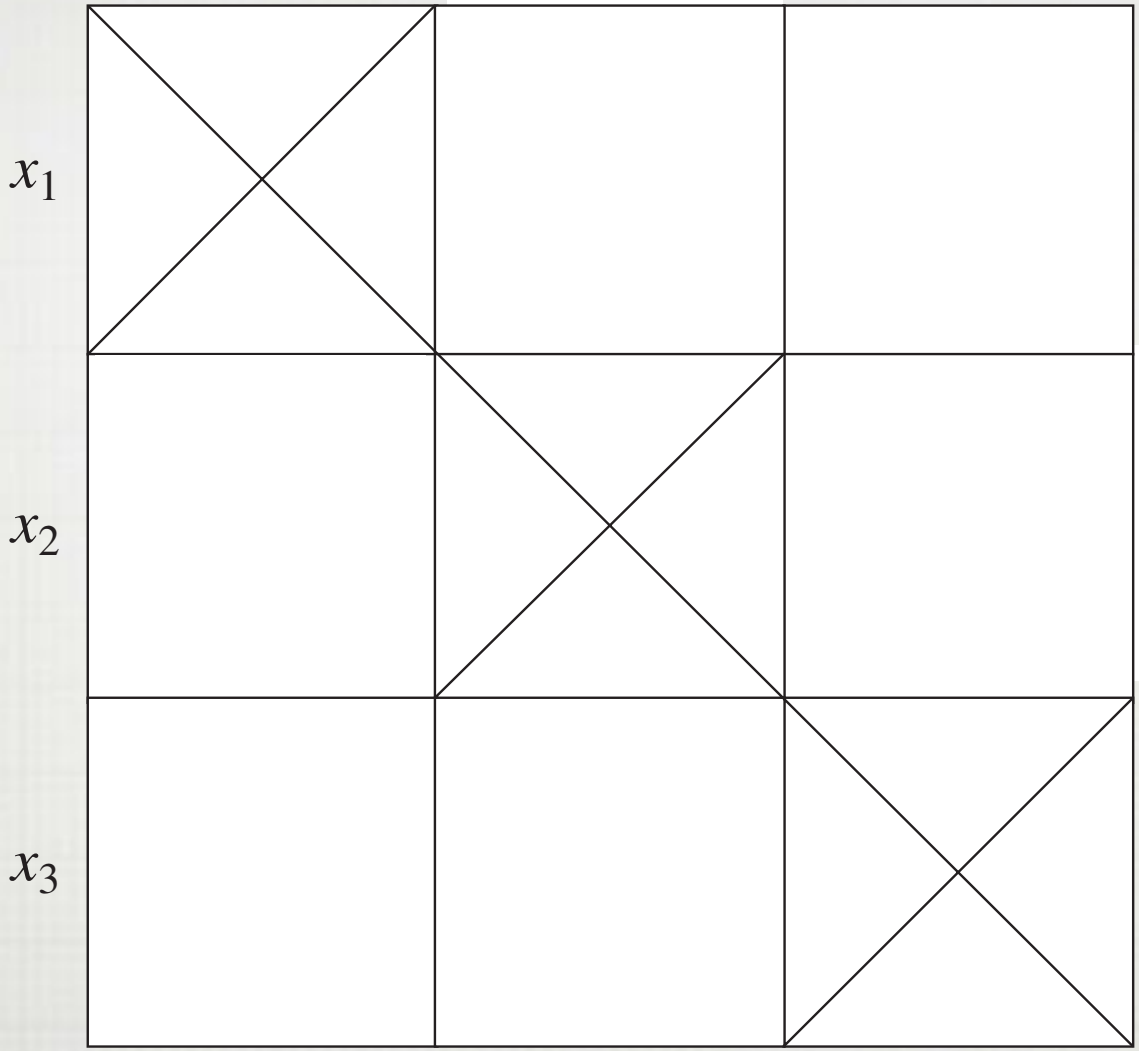
$$\sigma_{1,2} = \frac{1}{n}\sum_{k=\tau}^{n+\tau}(x_1(k) - \overline{x}_1)(x_2(k-\tau) - \overline{x}_2)$$

n=m
off-set in time, τ=2

t	x_1	x_2	...	x_m
t_1	$x_1(t_1)$	$x_2(t_1)$		$x_m(t_1)$
t_2	$x_1(t_2)$	$x_2(t_2)$		$x_m(t_2)$
t_3	$x_1(t_3)$	$x_2(t_3)$		$x_m(t_3)$
⋮	⋮			⋮
t_n	$x_1(t_n)$	$x_2(t_n)$		$x_m(t_n)$
t_{n+1}	$x_1(t_{n+1})$	$x_2(t_{n+1})$		$x_m(t_{n+1})$
t_{n+2}	$x_1(t_{n+2})$	$x_2(t_{n+2})$	...	$x_m(t_{n+2})$

MULTI-SCALE REPRESENTATION

Tiling Phase Portraits



x_1 vs. x_2 same as x_2 vs. x_1
→symmetric array

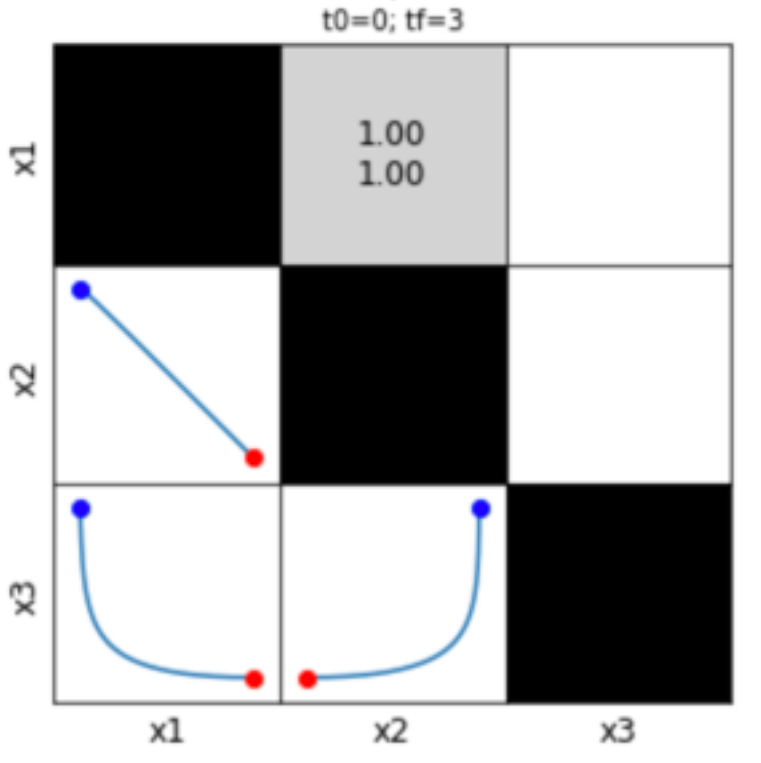
use corresponding locations in array to convey different information
•graph
•statistics

A series of tiled phase portraits can be prepared, one for each time scale of interest

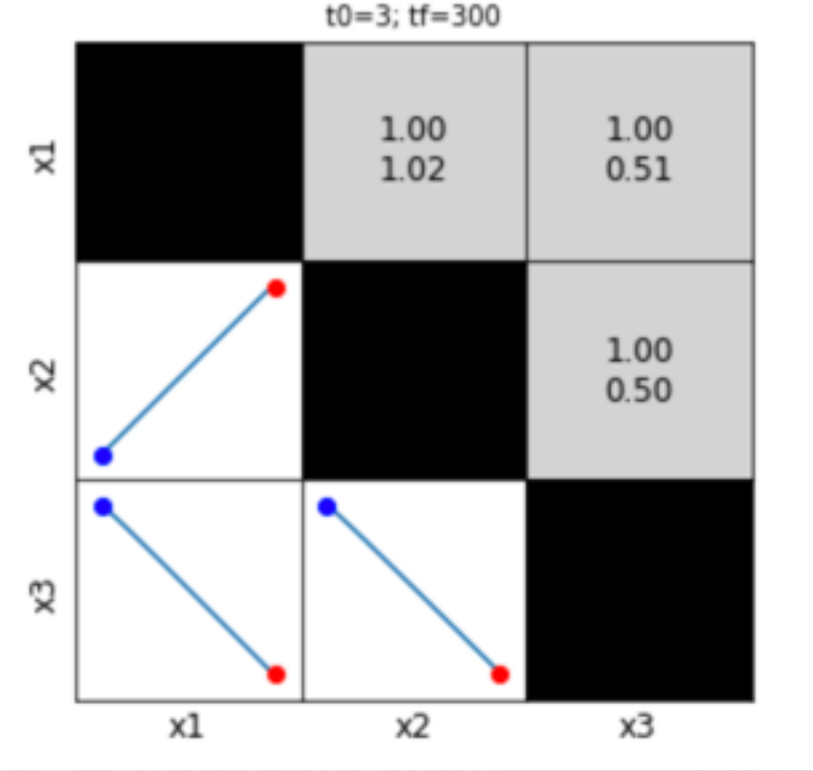
Representing Multiple Time-Scales:

tiling phase portraits on separate time scales

Time: 0 -> 3 sec



Time: 3 -> 300 sec



Red: t0
Blue: tf

Example System:

Fast	Intermediate	Slow
v_1	v_2	v_3
$x_1 \rightleftharpoons x_2$	$\rightleftharpoons$	$x_3 \rightarrow x_4$
v_{-1}	v_{-2}	

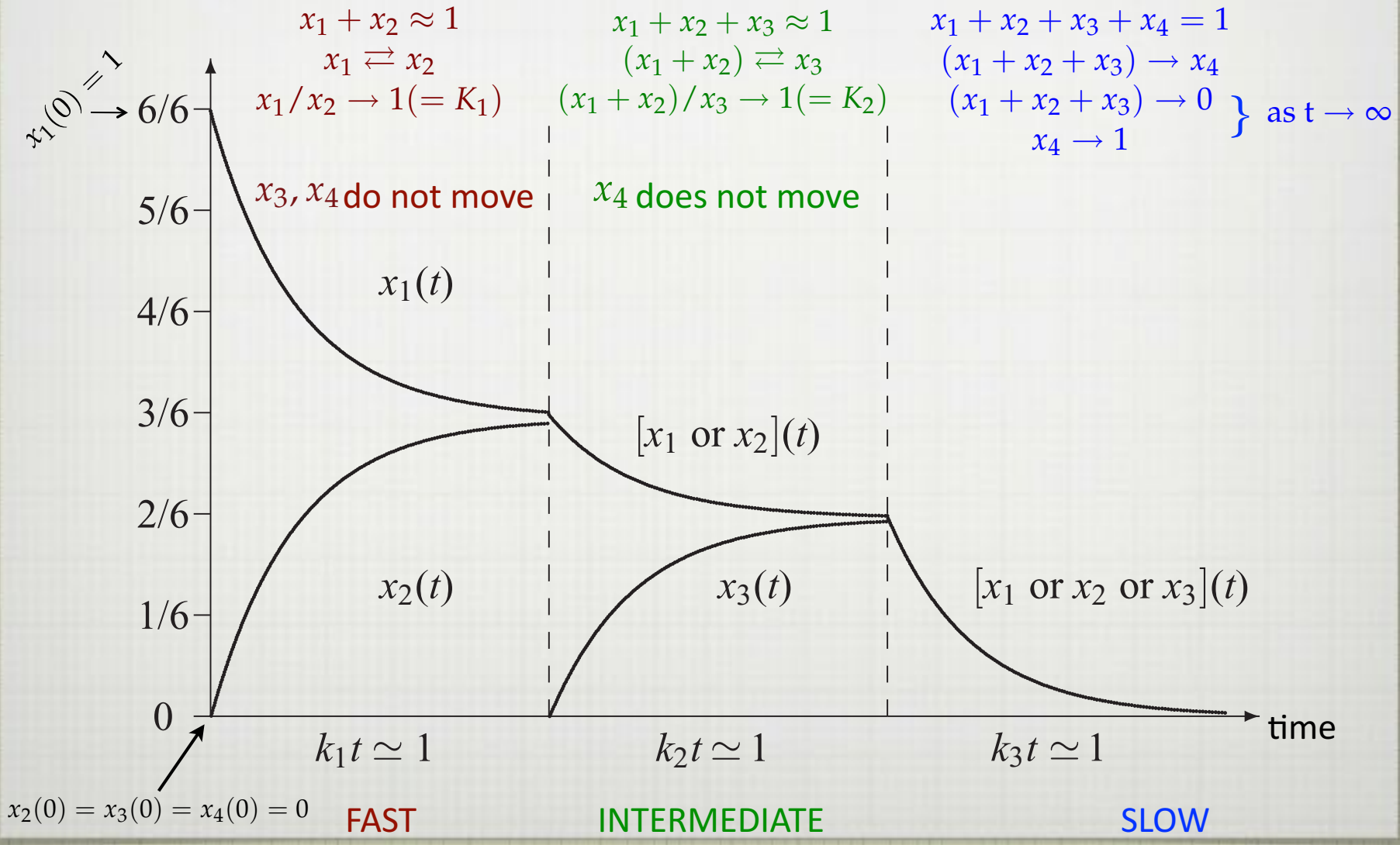
$$\frac{dx_1}{dt} = -v_1 + v_{-1} = -k_1 x_1 + k_{-1} x_2 \qquad x_{10} = x_1(t=0)$$

$$\frac{dx_2}{dt} = v_1 - v_{-1} - v_2 + v_{-2} = etc. \qquad x_{20} = x_2(t=0)$$

$$\frac{dx_3}{dt} = v_2 - v_{-2} - v_3 = etc. \qquad x_{30} = x_3(t=0)$$

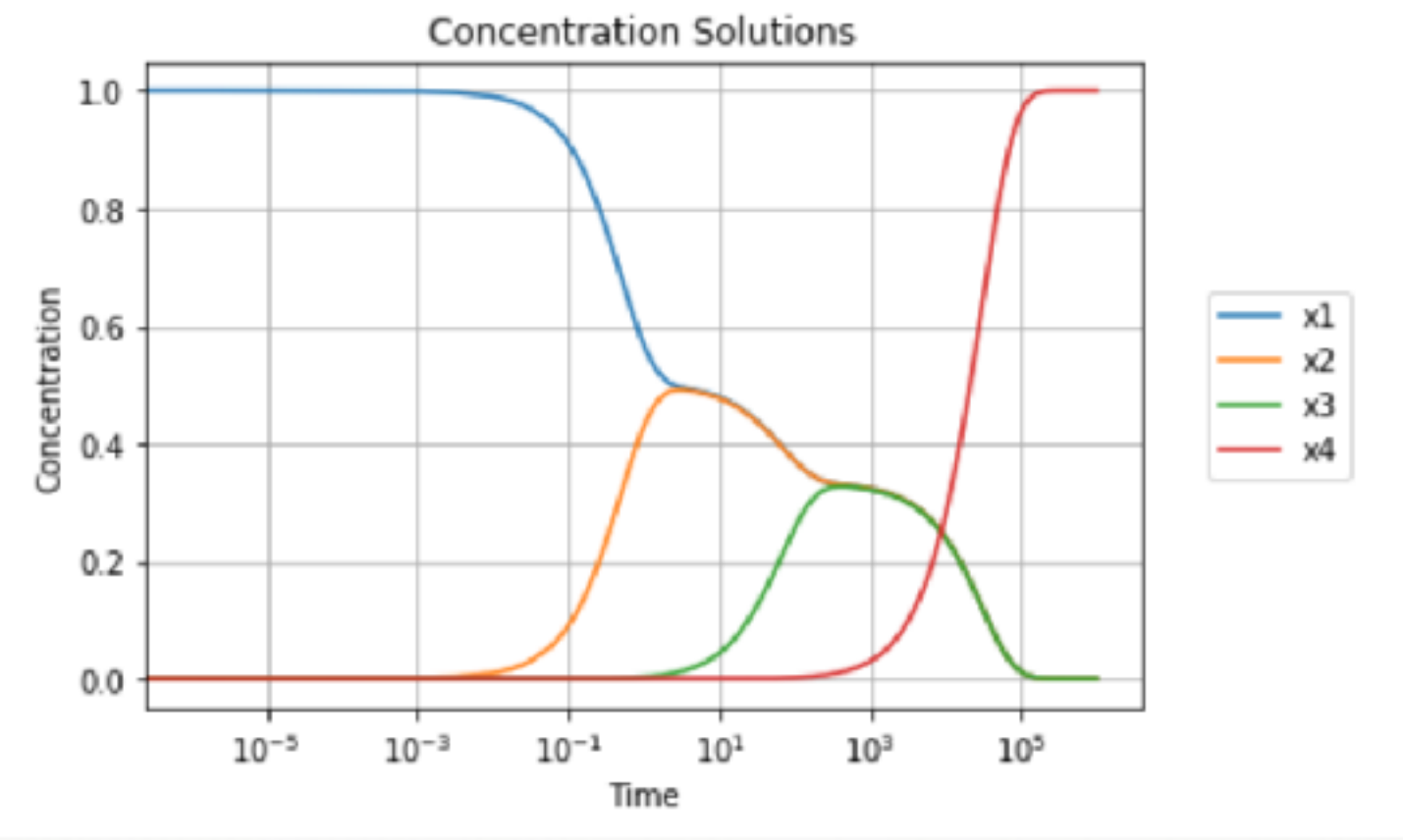
$$\frac{dx_4}{dt} = v_3 = k_3 x_3 \qquad x_{40} = x_4(t=0)$$

Example of Time-Scale Decomposition



Example of Time-Scale Decomposition

Actual Numerical response

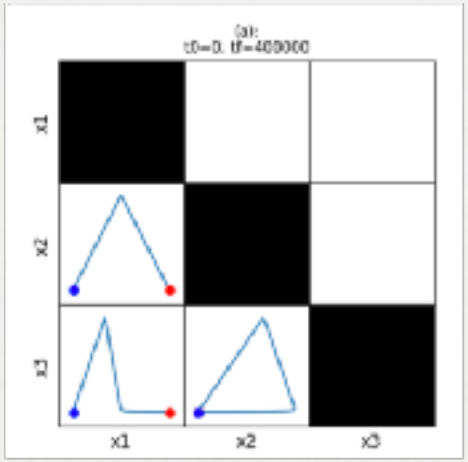


Tiled Phase Portraits:

overall and on each time scale

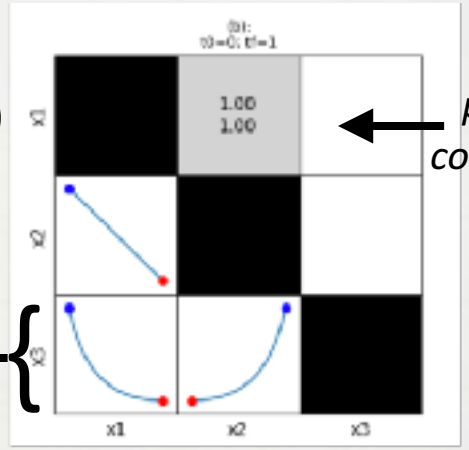
All Times

(a)



Fast

(b)



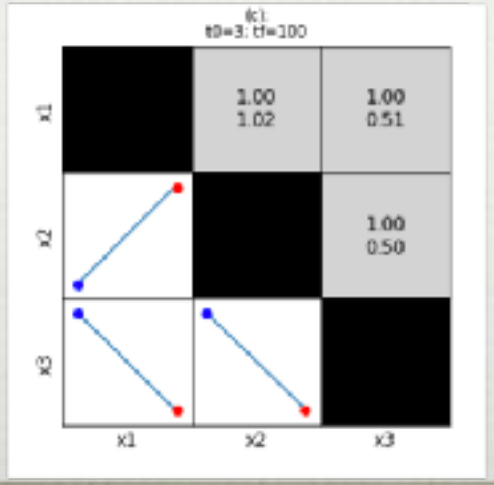
perfect correlation

expanded scale

$x_3 \sim 0$

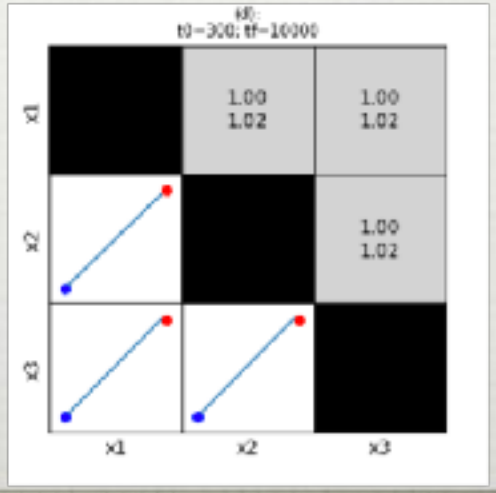
Intermediate

(c)



Slow

(d)



Post-processing the Solution

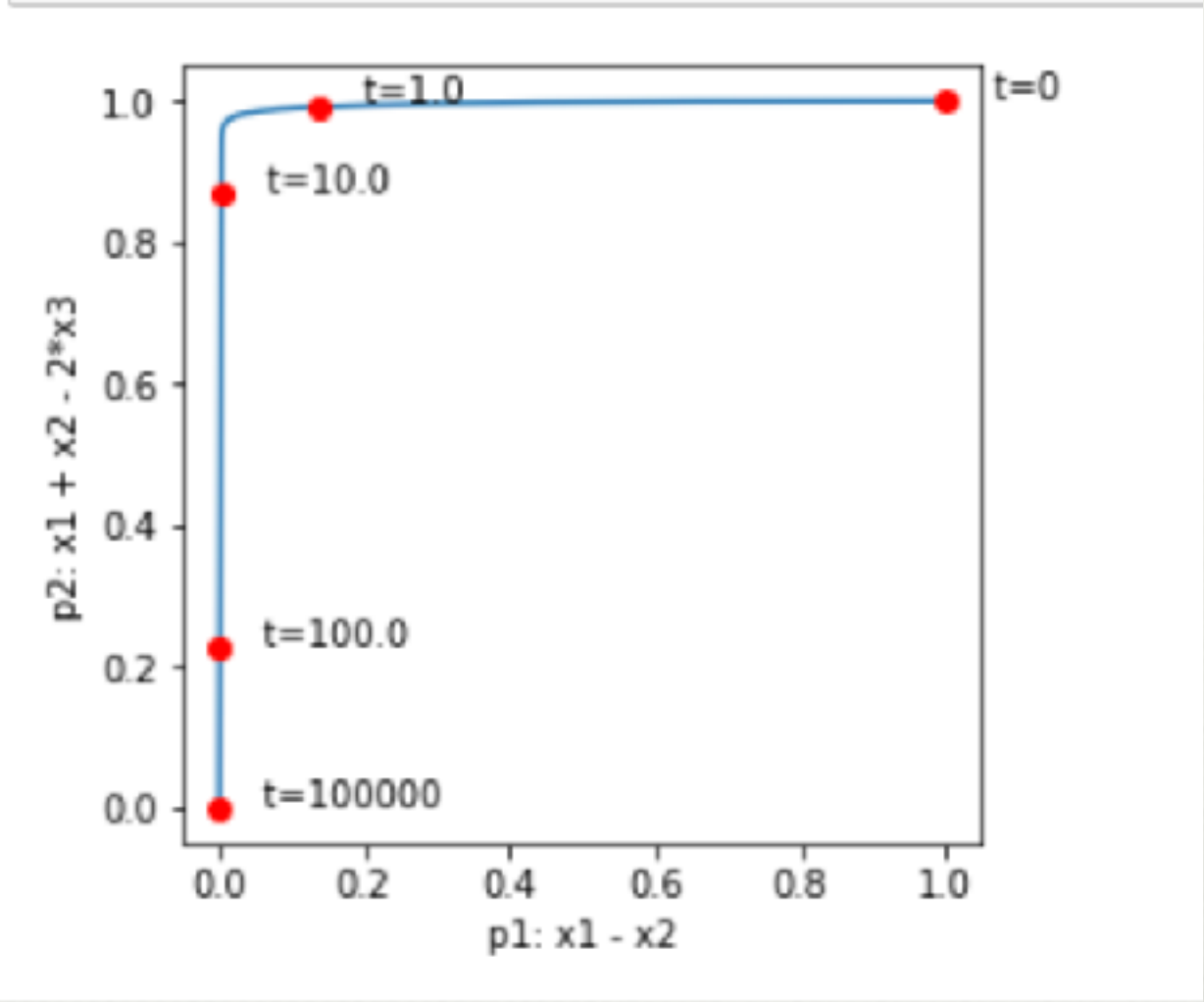
1. Forming pools

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ 1/2 & 1/2 & -1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 - x_2 \\ (x_1 + x_2)/2 - x_3 \\ x_1 + x_2 + x_3 \end{pmatrix}$$

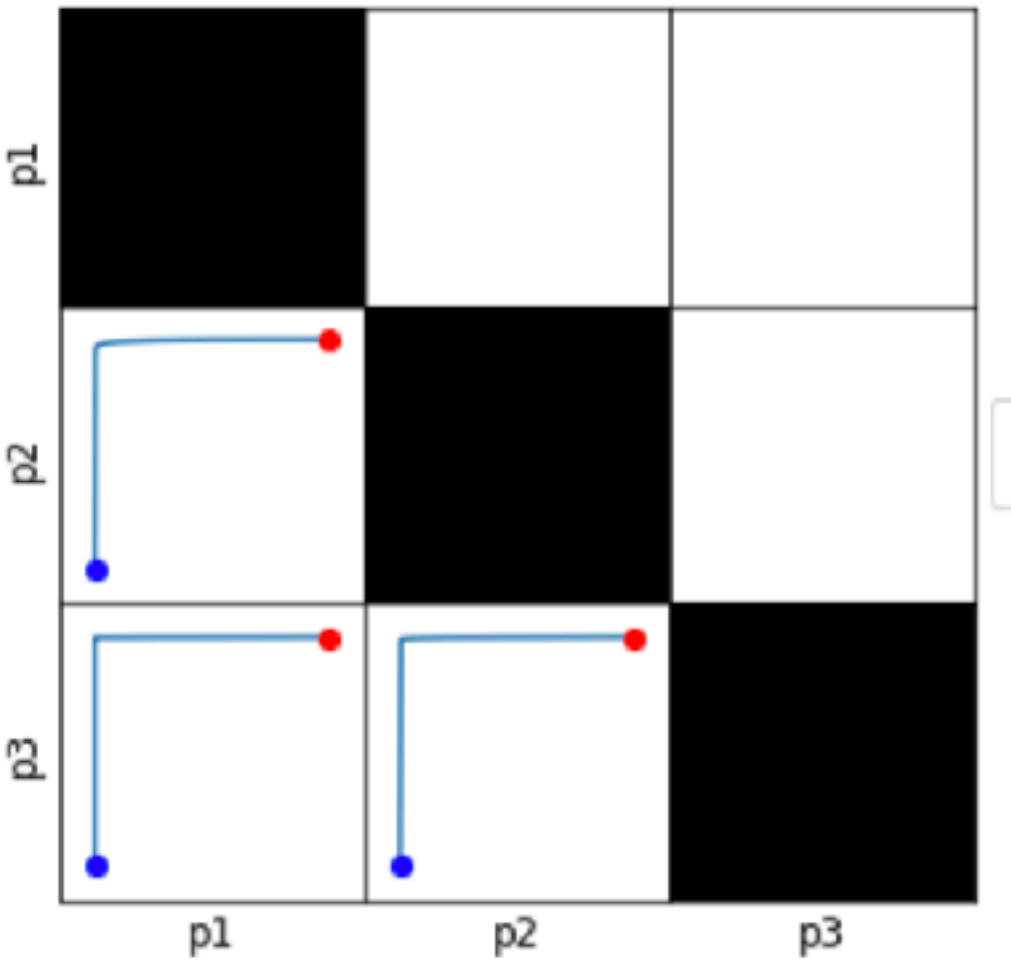
p_1 and p_2 are dis-equilibrium variables

p_3 is a conservation variable

Phase Portrait of Pools: L-shaped



Tiled Phase Portrait of Pools



$p_1 : x_1 - x_2$

$p_2 : x_1 + x_2 - 2x_3$

$p_3 : x_1 + x_2 + x_3$

Summary

- Network dynamics are described by dynamic mass balances $dx/dt = Sv(x;k)$ that are formulated after applying a series of simplifying assumptions
- To simulate the dynamic mass balances we have to specify the numerical values of the kinetic constants $(\mathbf{k})$, the initial conditions $(\mathbf{x})$, and any fixed boundary fluxes.
- The equations with the initial conditions can be integrated numerically using a variety of available software packages. We use MASSpy
- The solution is in a file that contains numerical values for the concentration variables at discrete time points. Often an interpolation function is provided. The solution is graphically displayed, i.e., as concentrations over time, or on a phase portrait.
- The solution can be post-processed following its initial analysis to bring out special dynamic features of the network.
- For the rest of the class, we will apply this simulation procedure to progressively more complicated situations, from the simple to the realistic.

The End