Beyond simulation and Big Data

How informatics and dynamics might merge to shape the future of modeling multi-scale diseases

Robert Clewley
Neuroscience Institute
Georgia State University

NSF CISE/CCF #0829742

… inspired by approaches of Ermentrout & Kopell, PNAS 1998; K. Yip, 1987; F. Zhao, 1994; B. Kuipers, 1993; others …

Henry Markram: *“We can [build a brain] within 10 years”*

Albert Einstein: *”You do not really understand something unless you can explain it to your grandmother”*

Rhetorical Q1. Will we understand multi-scale diseases such as epilepsy once we build Markram’s simulated brain?

Rhetorical Q2. What do we do with all the partial models of epilepsy that we have built?

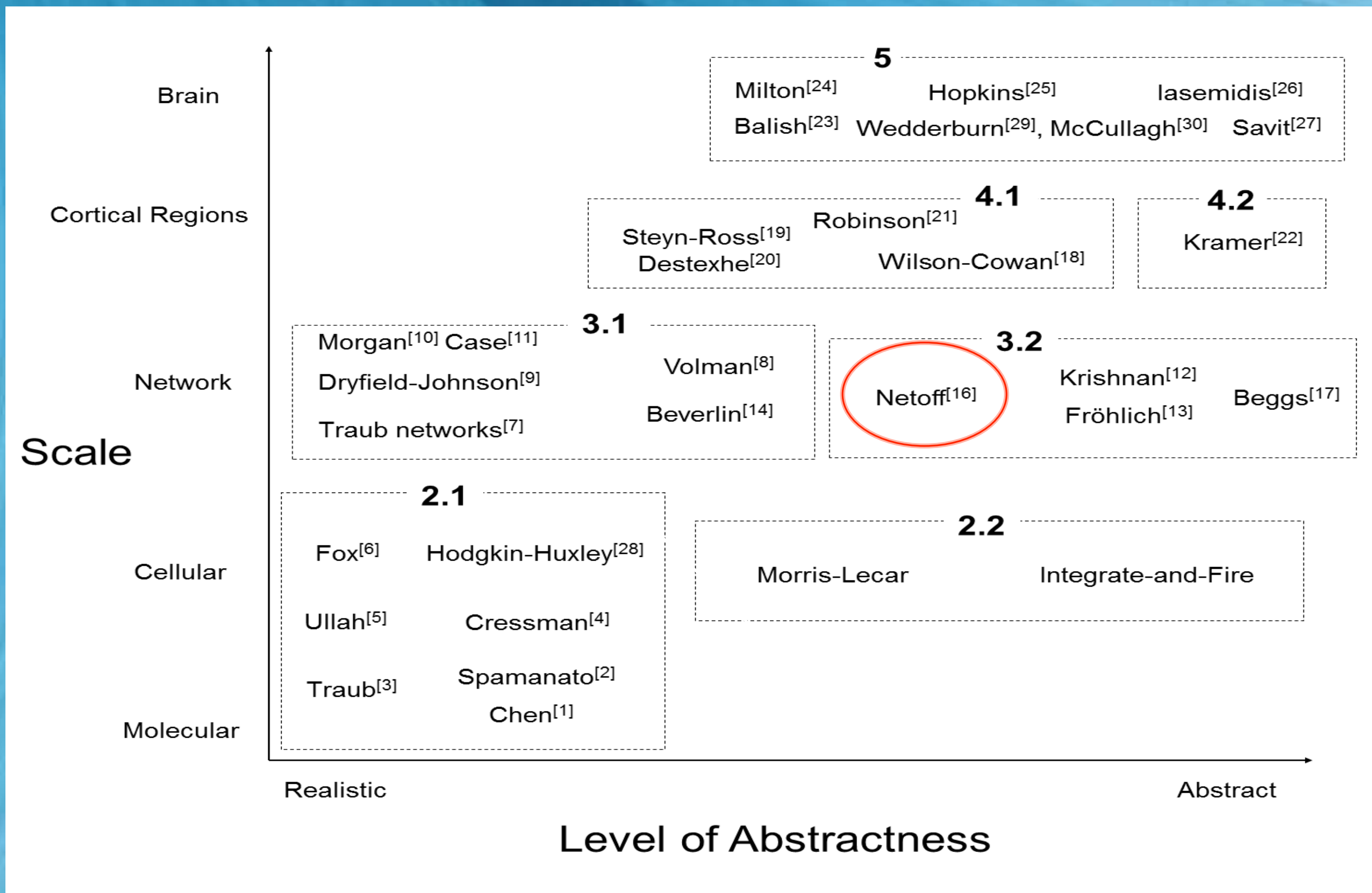
Q1. What’s missing in our methodology?

Q2. How might we better proceed?

What’s missing?

- It’s not lack of modeling effort …

Model zoology: *not a "model state"* ☺



[16] = Netoff, Clewley, et al., J Neurosci 2004

From Holt & Netoff, *in press*.

Model ethnography

- Models are often developed in isolation
  - The process is subjective and esoteric
  - What biological parts and relationships to represent?
  - Much 'curve fitting' is done
  - Assumptions often implicit

- Assumptions are often strong and *a priori*
  - Multiple scales may not be so well separated, or state-dependent
  - Too broad and inclusive (under-constrained)
  - Unconvincing about specifics in real systems
  - How they break helps connect to other models

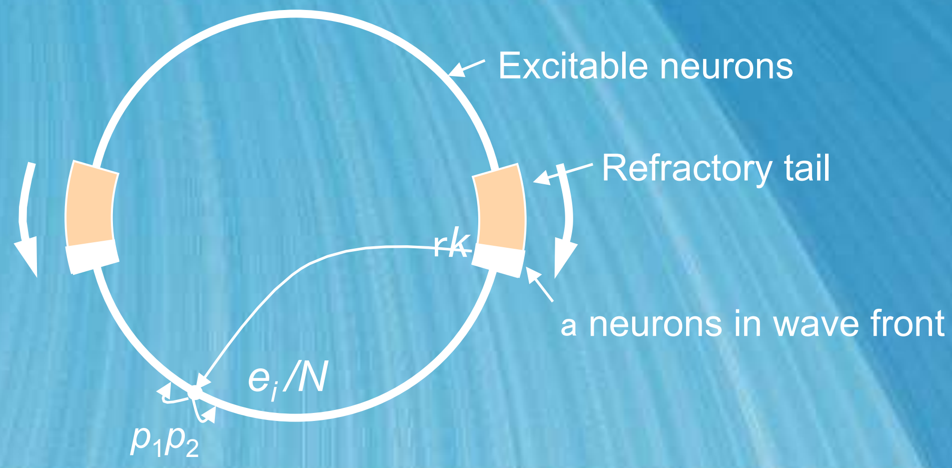
Model inter-operability matters

- Incompatible assumptions between models
  - Related example: General Circulation Models
- Difficult to:
  - reconcile predictions
  - trust / adapt parameter values
  - re-use, compose or unify models
  - explore space of possible models
  - understand underlying causal mechanisms of emergence
- Models are much more fragile than we like to admit
  - (in print)
- *Inter-operability is necessary for an efficient and robust multi-level understanding of brain function*

My (little) work on epilepsy

- Multi-scale representation of network dynamics in Hippocampus (Netoff et al., J Neurosci, 2004)
  - Predicts mechanistic roles for physiological parameters (alone or in combo)
  - Predictions easier to generate/analyze than using large data-driven simulations
  - Encodes basic assumptions
  - Validated against our large-scale (10k neuron) simulations
  - Abstract model derived from detailed model, first principles, and data - no curve fitting!

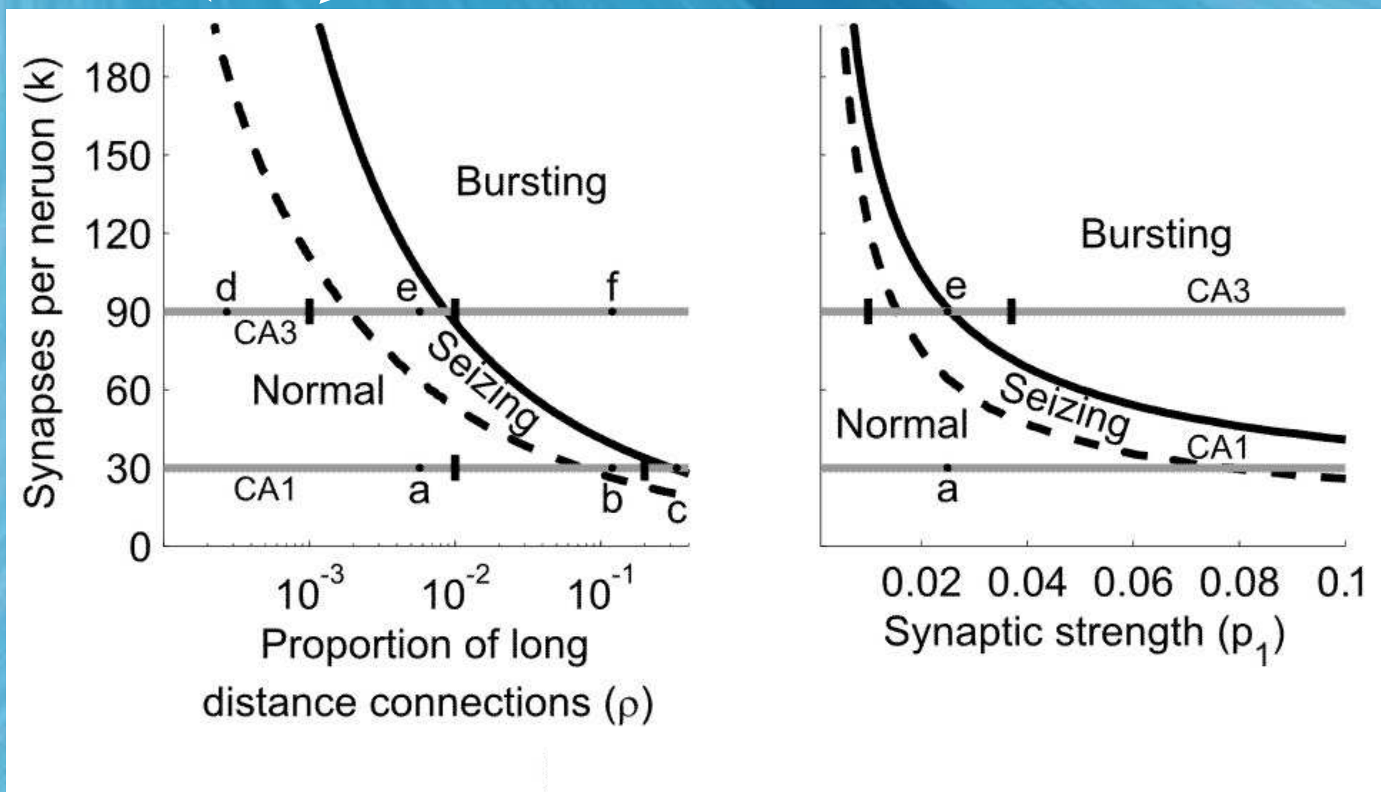
Small-world connectivity (not shown)



My (little) work on epilepsy

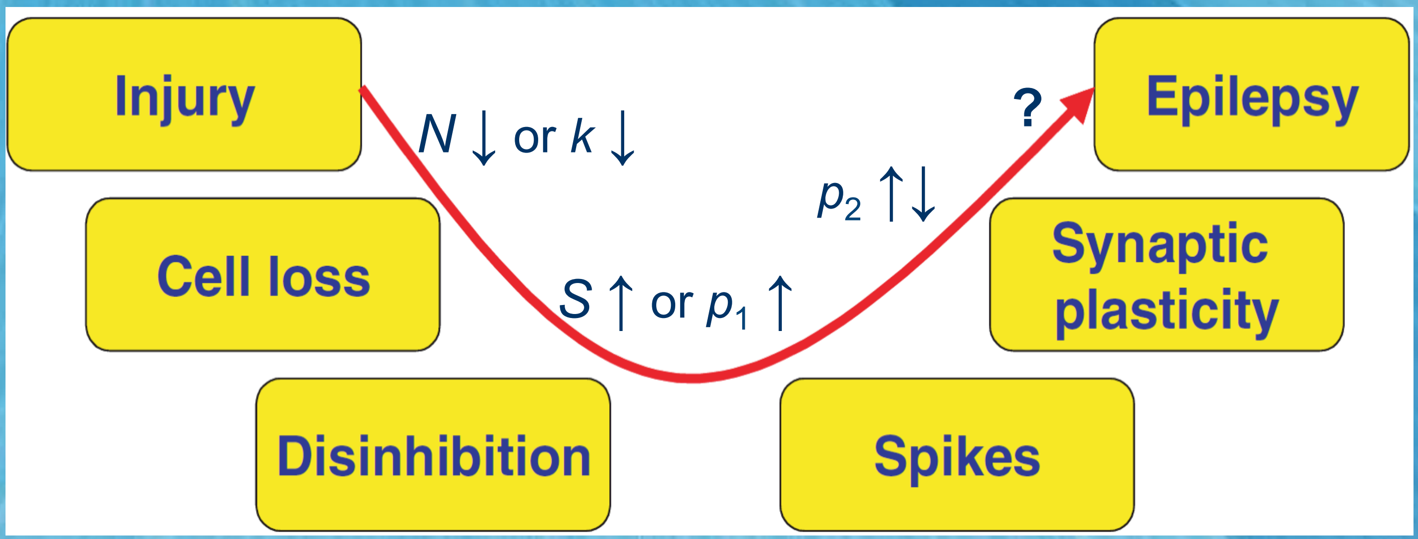
Equilibrium loses stability in (1+*R*)-D map, 1-D map

E.g. mappings correctly predict increasing synaptic efficacy causes network to burst in CA3 before CA1



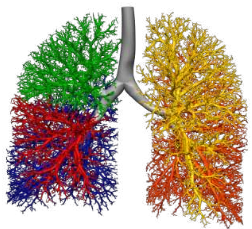
Multi-level mechanisms of epileptogenesis?

- **A prototype for predicting result of parameter changes in mechanistic, computational models across multiple scales**
- **Provides anchors for experimental testing of assumptions and predictions**

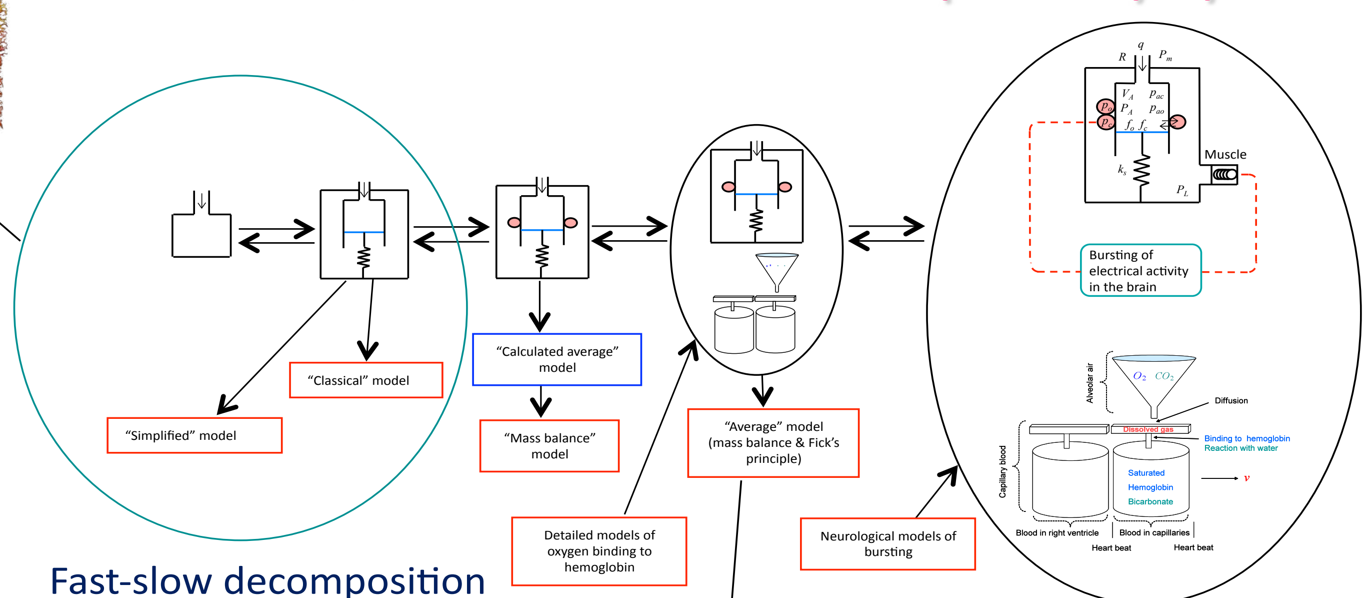


Meta-studies, contextualization

Example from the work of Alona Ben-Tal, Massey University

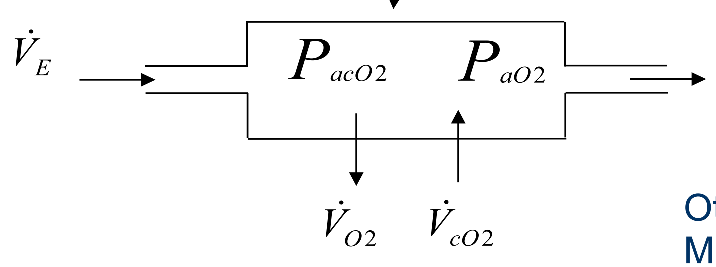


Mathematical models of the respiratory system



Averaging

$$\bar{q}_i = \frac{1}{T}\int_0^T q_i(t)dt$$



A community bias is that this is insufficiently innovative, novel work: "it is just admin"

Other image credits: M. Tawhai, M. Khoo

What's missing?

- It's not lack of modeling effort
- It's not data …

The End of Theory - Wired magazine, June 2008

- "The quest for knowledge used to begin with grand theories. Now it begins with massive amounts of data. Welcome to the Petabyte Age."
  - We're obsessed with greedy **data** collection and list making, from Facebook to gene sequencing
  - We spend much less effort and $$ on *good hypotheses*

- NSF has substantial funds for bioinformatics, but Roger Pennington, head of its OCI admits:
  - "We don't know how to do data-driven science"
  - His view is not being taken seriously

What’s missing?

- It’s not lack of modeling effort
- It’s not data
- It’s not lack of computing power …

High-Performance Computing

- **Raw computing power** is great for collecting, documenting, associating or creating **raw data**
  - It *is* reasonable to exhaust lucrative low-hanging scientific targets with new technology
  - But funding agencies and policy makers *act* as if all we need to do is keep building bigger computers
  - It is certainly a more tangible goal, simplifies funding decisions, looks shiny, and makes $$$
- Brutes tend to get stupid or lazy
  - When all you have is a hammer …
  - And that hammer is increasingly cheap compared to alternatives

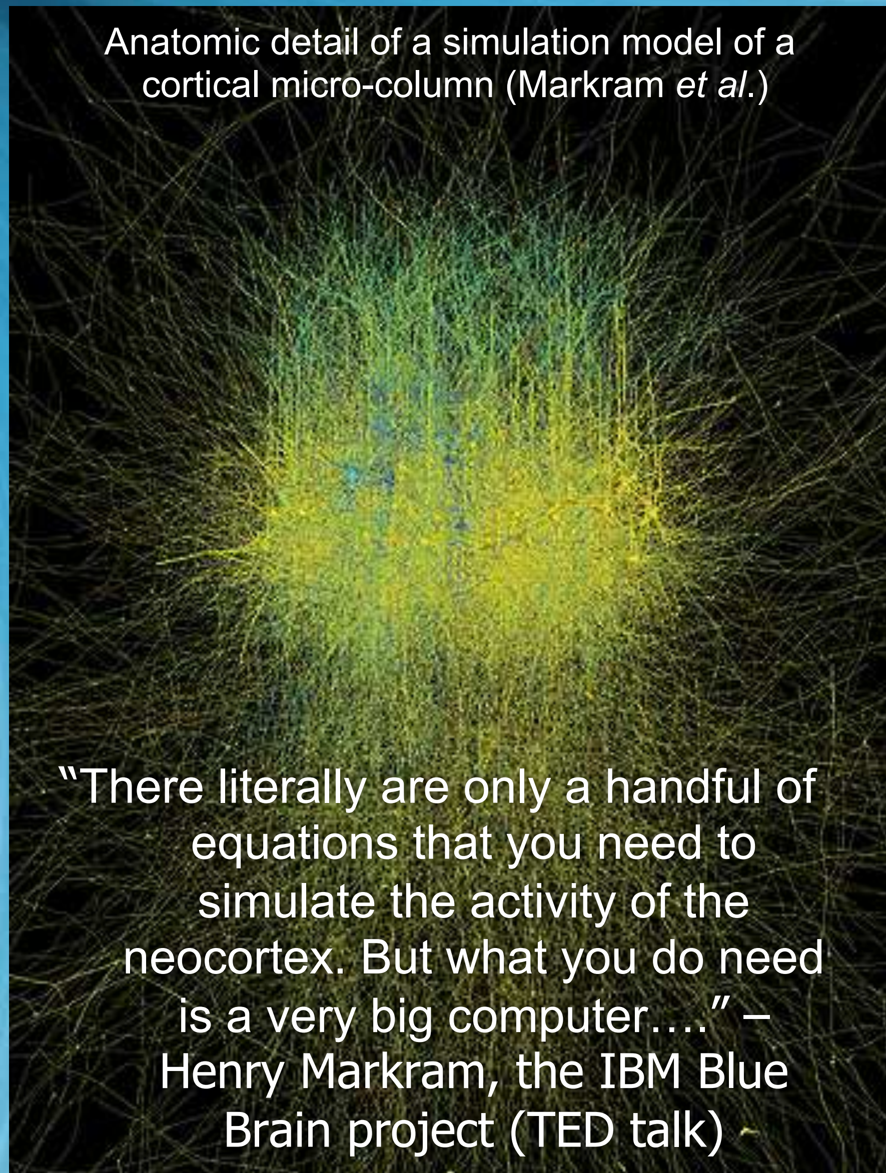
"So, let's switch it on…"

What is the desired function / behavior?

How do we know if it worked?

Are there enough constraints from experiments to justify this detail?

How sensitive is the network to changes in connectivity or parameters?



Anatomic detail of a simulation model of a cortical micro-column (Markram *et al.*)

"There literally are only a handful of equations that you need to simulate the activity of the neocortex. But what you do need is a very big computer…." – Henry Markram, the IBM Blue Brain project (TED talk)

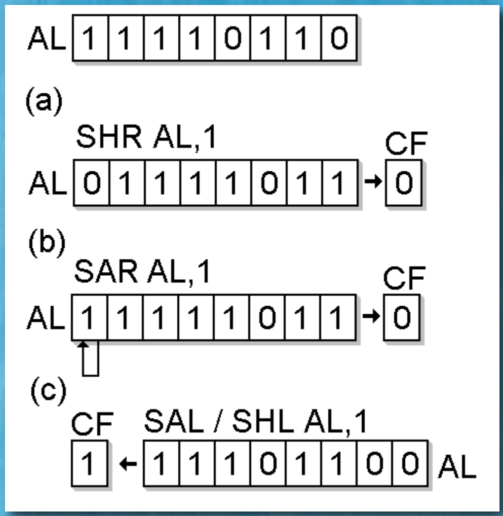
What are the organizing principles at the single-cell / sub-circuit / small network levels?

What are the essential causal mechanisms of emergent dynamics?

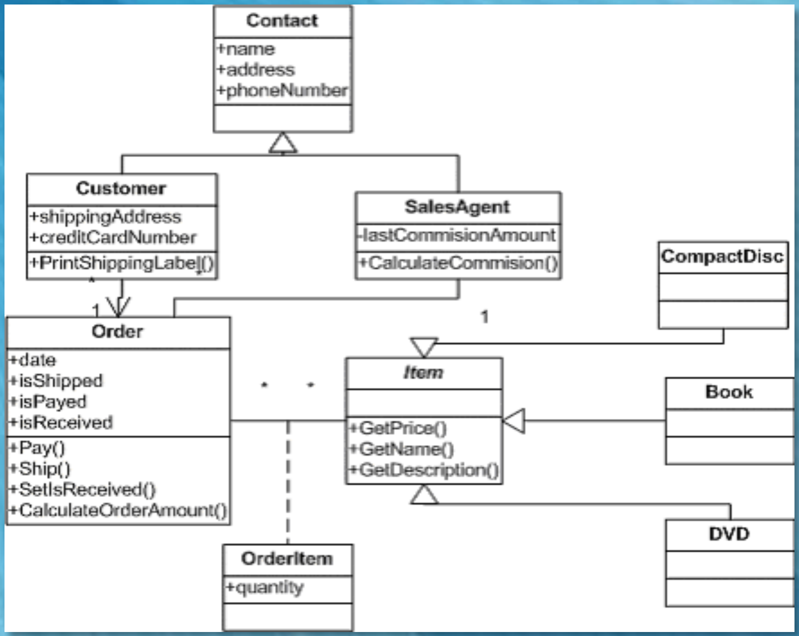
Haven't we replaced the study of one complex system with another?

Heterogeneous Representations

- Would you program Amazon.com's sales transaction system using 1's and 0's and CPU code?



vs.



- No, use high-level languages (e.g., Object Oriented) and appropriate representations for each component

Heterogeneous Representations

- How do we expect to build complex biological models directly from differential equations, parameter values and initial conditions?

$$C_m \frac{dV}{dt} = \sum_{ionic} I_i + I_{axial}$$

$$I_i = g_i m_i^{p_i} h_i^{q_i} (V - E_i)$$

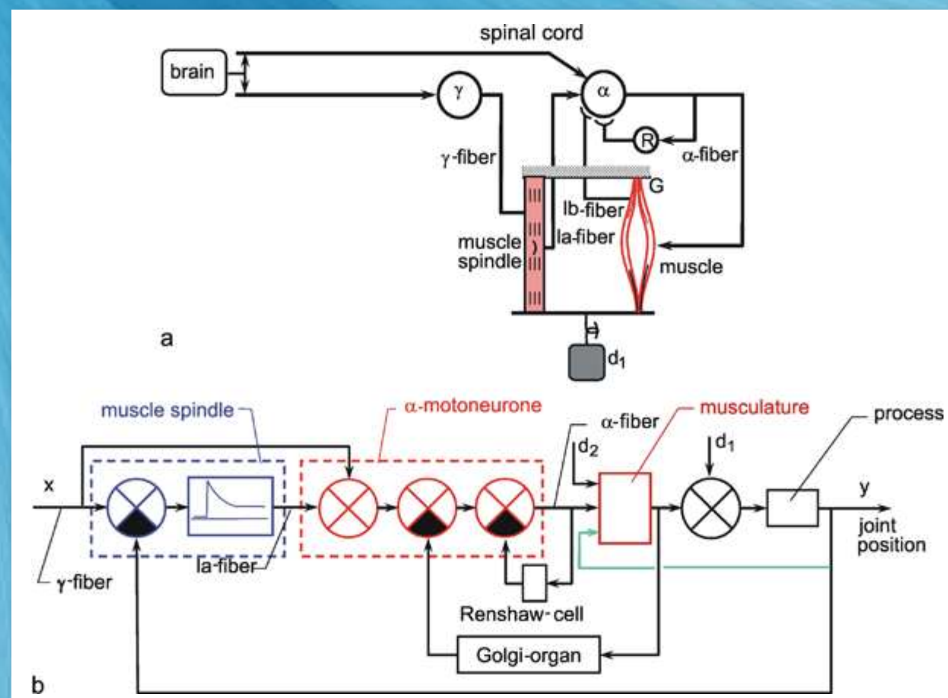
$$I_{axial_{S/N}} = g_{axial}(V - V_{axon}) = -I_{axial_A}$$

$$\tau_s(V)\frac{ds}{dt} = s_\infty(V) - s, \qquad s = m_i, h_i$$

$$\tau_{Ca}\frac{d[Ca^{2+}]}{dt} = -FI_{Ca} - [Ca^{2+}] + C_0$$

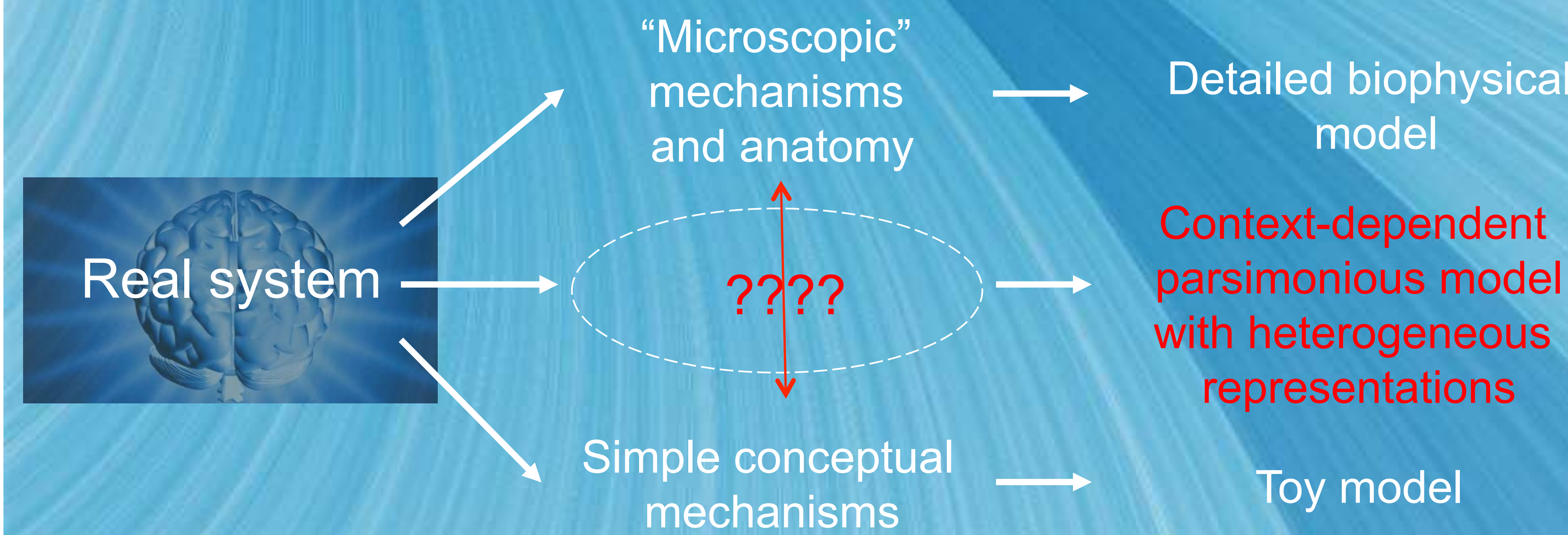
$$= C_\infty(I_{Ca}) - [Ca^{2+}]$$

vs.



So, what *is* missing?

- Understanding dynamics and model assumptions across levels
  - Informatics means creating **information**, not merely data
  - Information from data requires hypotheses and theories
- I suggest that we need an intermediate ground between detailed biophysical models and abstract, toy models
  - Technological (computational tools, CPU power, storage)
  - Mathematical (dynamical systems, networks, statistics, reduction tech)
  - Informatic (inference, heuristics, logic, databases, book-keeping)



A rational plan for progress

Short term:

- Meta-studies
- Transparency
- Diagnostics
- Validation
- Collaborate and share

Medium term:

- Figure out how to do data-driven, multi-level modeling
- Qualitative reasoning
- Computer-assisted management of model building and testing

All of this is possible even without inevitable improvements in mathematics itself

There are a few precedents for all of the medium-term ideas dating back to the 1990s

How to fill in what is missing?

- Qualitative fitting of models to
  - data (analysis)
  - hypothesized mechanisms (synthesis)
- Define a suite of qualitative features to fit
  - Need multi-objective optimization
  - Explore space of hybrid models
- Context-dependent reductions to hybrid models
  - Generate truly modular, inter-operable models
  - That we stand a chance of *understanding*

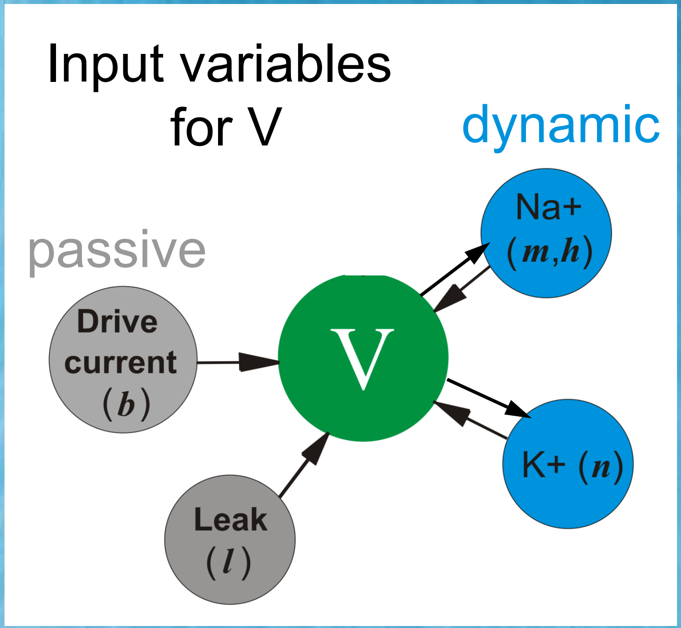
Reverse engineering casual mechanisms

- If you think you know how "it" works:
  - build it
  - stretch it
  - *break it*
  - diagnose it
  - re-build it

- Let's exemplify some of this on a single cell:
  - 1) Show that Hodgkin-Huxley action potential can be represented with a hybrid (piecewise) low-dimensional model
  - 2) Show that the classic Phase Response Curve shape can be understood from relationships between transient ionic channel dynamics

- Addresses how weak is 'weak', far from limit cycle, etc.

Review: dominant scale analysis for Hodgkin-Huxley



(Clewley, Rotstein, Kopell, 2005)

$$V_\infty(t) = \frac{\sum(\text{driving forces})}{\sum(\text{conductances})}$$

$$\tau_V(t) = 1/\sum(\text{conductances})$$

$$\left\{ \tau_x \frac{dx}{dt} = x_\infty - x \right\}_{x \in \{V,m,n,h\}}$$

$V_\infty(t)$ is like a quasi-static fixed point for V

"Influence strengths" for V are like eigenvalues for V_∞

Dominant scale analysis

- Algorithmic form of multiple scale analysis
- Finds dominant variables and fast/slow variables
- Separates dynamics into regimes

Influence Ψ_s and **rate of influence** Ω_s defined along an orbit w.r.t. $s = m, n, h$.

Also, time scale sets *Fast* and *Slow* relative to V.

$$\Psi_s := \left|\frac{\partial V_\infty}{\partial s}\right|, \quad \Omega_s := \left|\frac{\partial V_\infty}{\partial s}\frac{ds}{dt}\right|$$

Each of these are ranked at every time step, then thresholded to determine dominant sets $Acts_\Psi(t)$, $Acts_\Omega(t)$.

Ψ_s measure which currents control local null-surface position (Cf. q.s.f.p.).

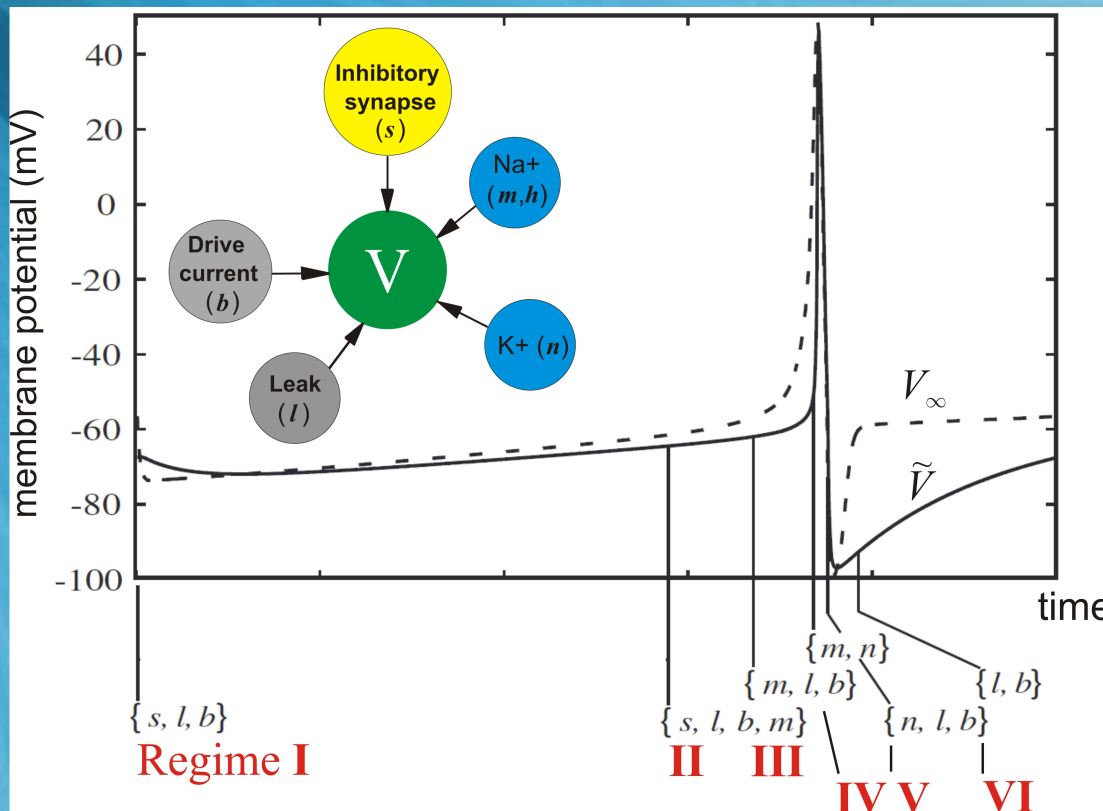
Ω_s determine "local autonomy" of reduced system: "which phase plane is OK".

Clewley, Rotstein, Kopell, *Multiscale Modeling & Simulation 4(3)*, 2005

Software tool: **DSSRT**

Example: Type I interneuron with time-dependent input (*dimension* 4)

Calculate dominance of currents along limit cycle, partitioning into lower-dimensional regimes



Reg.	Time interval	Dynamic	Passive	Bif. par.	Dim.
I	[00.00, 29.04)	s, V	l, b	m	2
II	[29.04, 33.57)	m[F], h, s, V	l, b		3
III	[33.57, 36.87)	m[F], h, V	l, b	n	2
IV	[36.87, 37.62)	m, h[S], n[S], V			2
V	[37.62, 39.21)	n, V	l, b		2
VI	[39.21, 49.98)	V	l, b	s	1

[F] = fast
[S] = slow

Na$^+$/K$^+$ AP mechanistic "template"

- Logical rules for defining
  - regimes / motifs (smooth dynamics)
  - regime transitions (discrete events)
- Encode these to get hybrid model definition
  - A meeting of informatics and dynamics !

Show video!

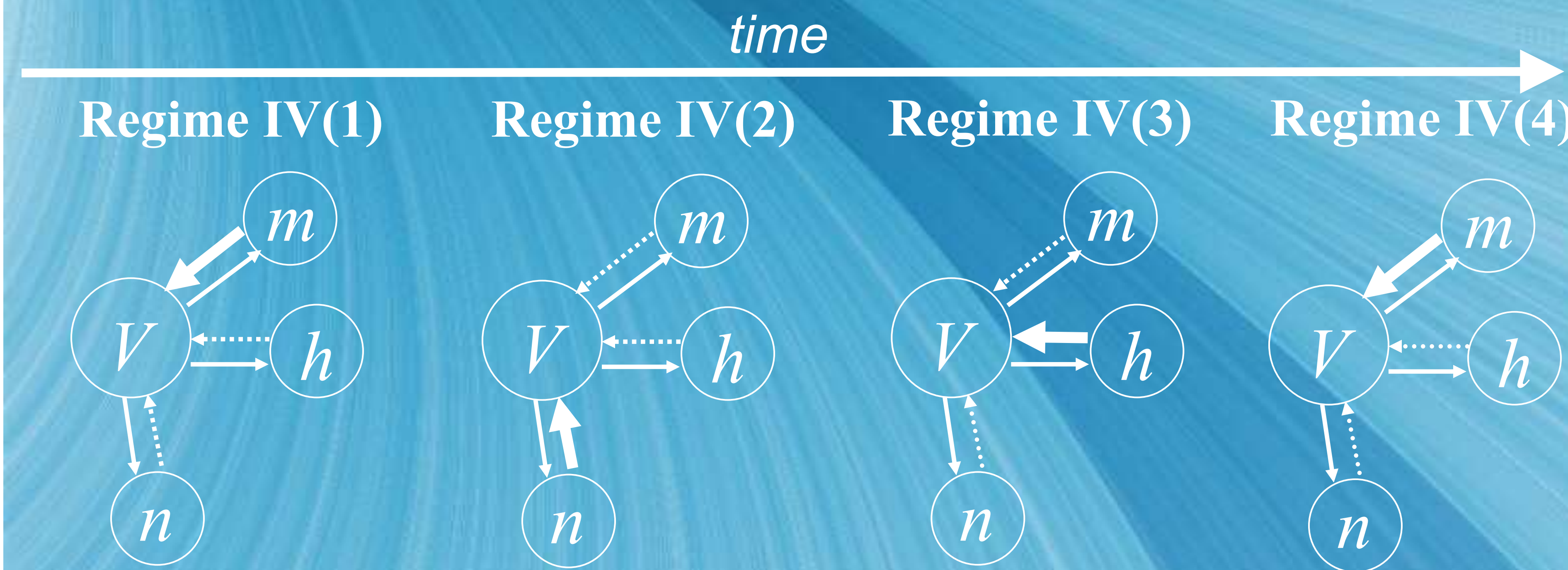
Table 1 Domain and transition motif rules making up the AP template, and the appropriate phase planes in which to view the reduced dynamics.

Motif	Phase plane	Domain rule	Transitional rule
I	(m, V)	$m \in \mathscr{A}_\Psi$	$n = \mathscr{A}_\Omega[1] \wedge n \in \mathscr{S}$
II	(n, V)	$n \in \mathscr{A}_\Psi \wedge m \notin \mathscr{A}_\Psi \wedge h \notin \mathscr{F} \wedge n \notin \mathscr{F}$	$h = \mathscr{A}_\Omega[1]$
III	(h, V)	$(m \in \mathscr{A}_\Psi \vee m \in \mathscr{M}_\Psi) \wedge (n \in \mathscr{A}_\Psi \vee n \in \mathscr{M}_\Psi)$ $\wedge\, h \notin \mathscr{F} \wedge n \notin \mathscr{F}$	$m = \mathscr{A}_\Omega[1]$
IV	(m, V)		$m \notin \mathscr{A}_\Omega$

(R. Clewley, *J. Comput. Neurosci.*, 2010)

Representing an AP with a hybrid model

- AP never before reduced to a self-contained math. description
  - Simulatable as hybrid model
  - Understandable intuitively
  - In traditional fast-slow analysis, *n* & *h* had to be imposed magically



n→*V* & *h*→*V* frozen

m and *V* slower than *n* and *h* (*m*→*V* & *h*→*V* frozen)

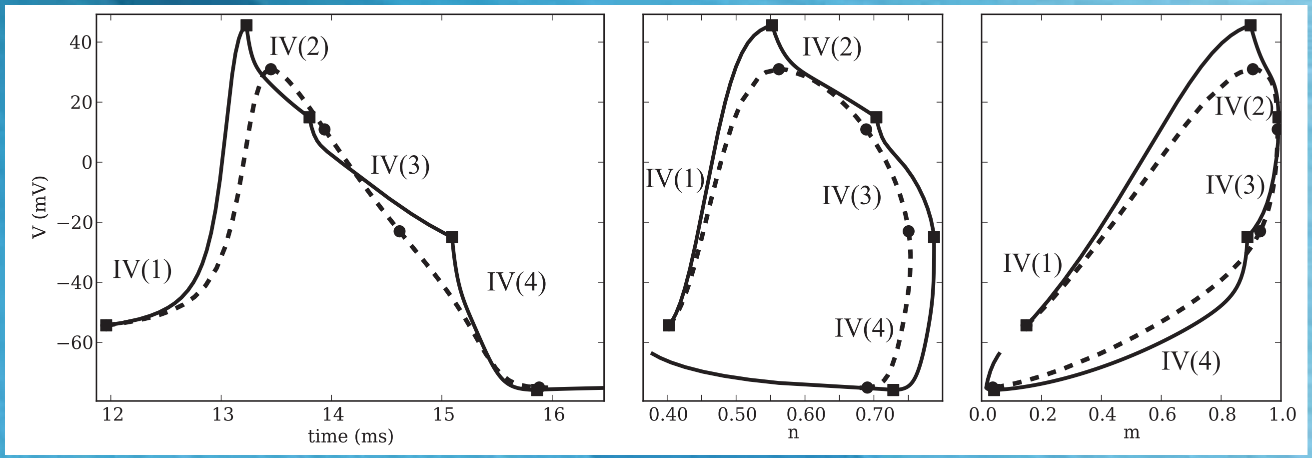
m and *V* still slower (*n*→*V* & m→*V* frozen)

n→*V* & *h*→*V* are frozen

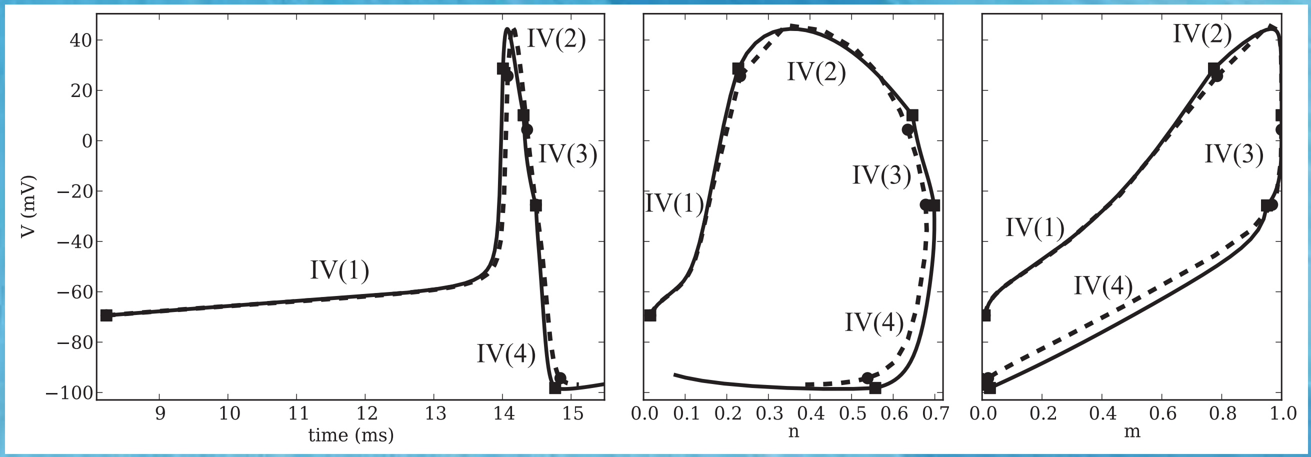
Representing an AP with a hybrid model

(*qualitatively*, as you can see…)

Classic H-H
(Type II params)



Modified H-H
(Type I params
for interneuron)



(R. Clewley, *J. Comput. Neurosci.*, 2010)

Locally-reduced models

- Applies ideas adapted from asymptotics
  - Fast vs. slow currents, large vs. small
  - *Weak vs. strong is most important*
  - Takes advantage of coupling patterns & emergent scales (not *a priori*)

- Explicit domain of validity for reduced models
- Global consistency checks
- Requires automated reasoning,
  book-keeping, data abstraction
- Good for dealing with transients

Phase Response Curve

- Popular way to predict timing relationships in networks of oscillators based on PRC of components
  - e.g. synchrony
- Measured empirically (bio) / numerically (model)
- Cannot be derived analytically for realistic models
- Ionic basis for shape not understood!
  - Would like to know how changing channels affects it
  - Discussing **isochrons** is just the first step

Isochrons

- Isochrons are level sets of phase around a P.O.
  - Usually for stable periodic orbits, but extendable to excitable systems …
  - To study transients, here we will extend to any local neighborhood in terms of exit time (Cf. Day et al., 2009)
- *What ionic properties determine isochrons?*

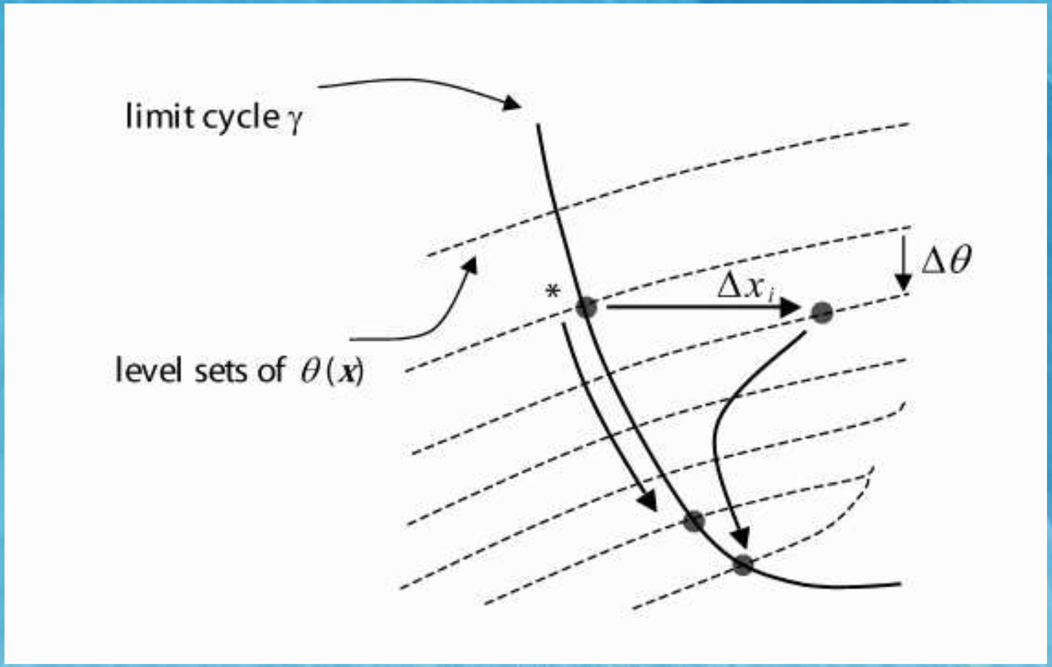
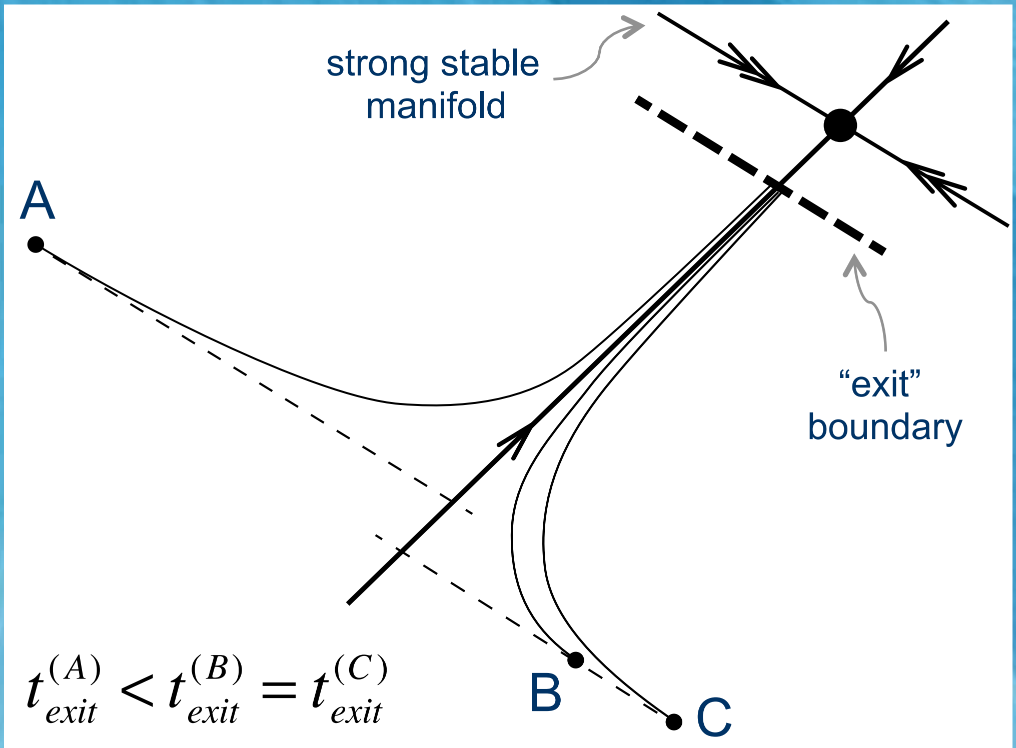


Image credit: *Scholarpedia*

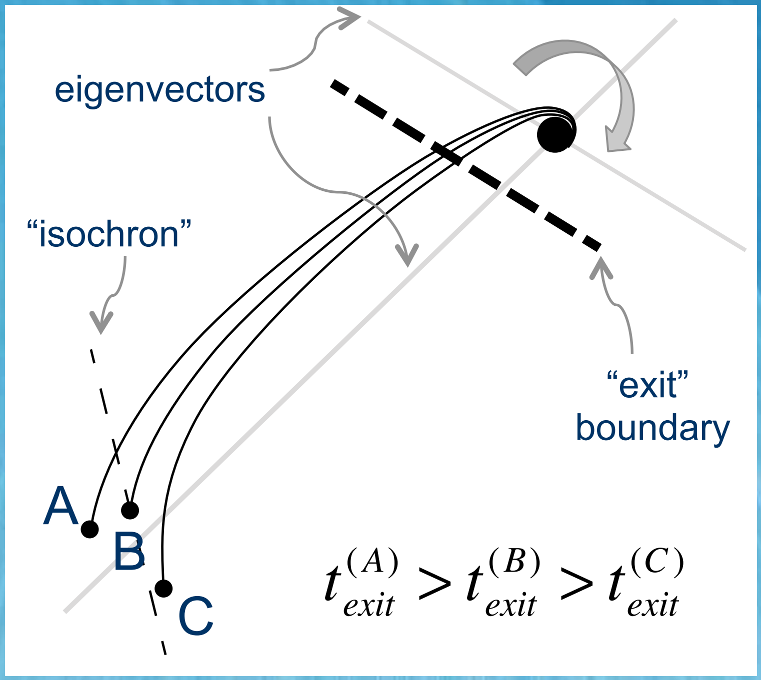
"Isochron" surrogate near node

- Suppose a **linear** stable node with unequal eigenvalues
- Consider time taken to reach an "exit" boundary in the plane
- Boundary is defined to be parallel to the strong stable manifold
- Exit times can be computed analytically
  - defines "relative phase"



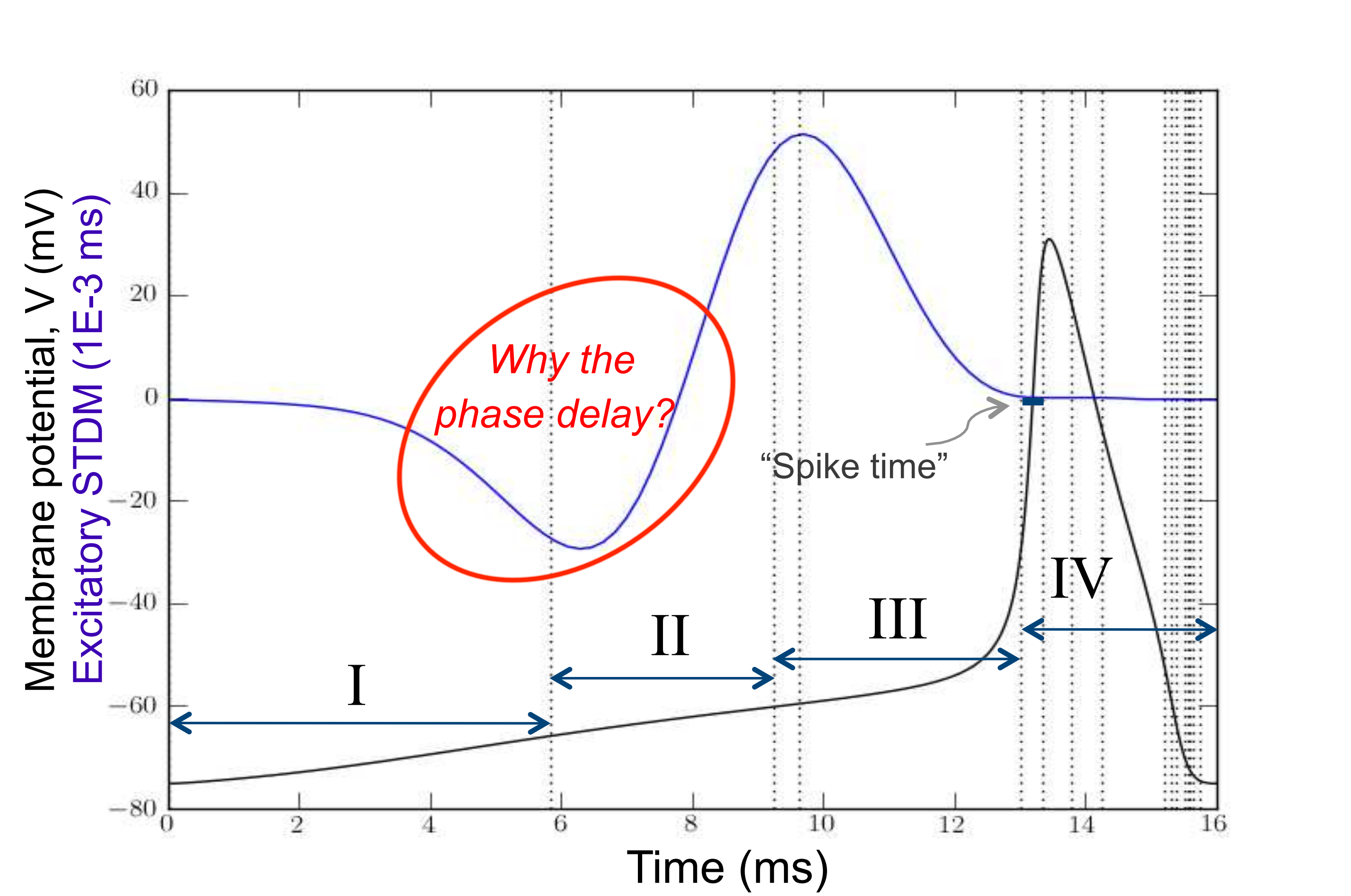
"Isochron" surrogate near focus

- No stable manifolds now (complex eigenvectors)
- Eigenvectors still indicate 'axes' for relative compression of spiral
- For a given position and angle of exit boundary, there is an "isochron" vector at point B dividing phase advance and delay relative to trajectory starting from B ending at boundary



PRC (STDM) for Type II H-H

- Spike Time Difference Map (STDM) in lieu of a PRC
  - we won't need periodicity anyway
  - using a +0.1mV perturbation



Regime I

$\mathscr{A}_\Psi = \{n[S]\}$

Regime II

$\mathscr{A}_\Psi = \{n[S], m[F]\}$

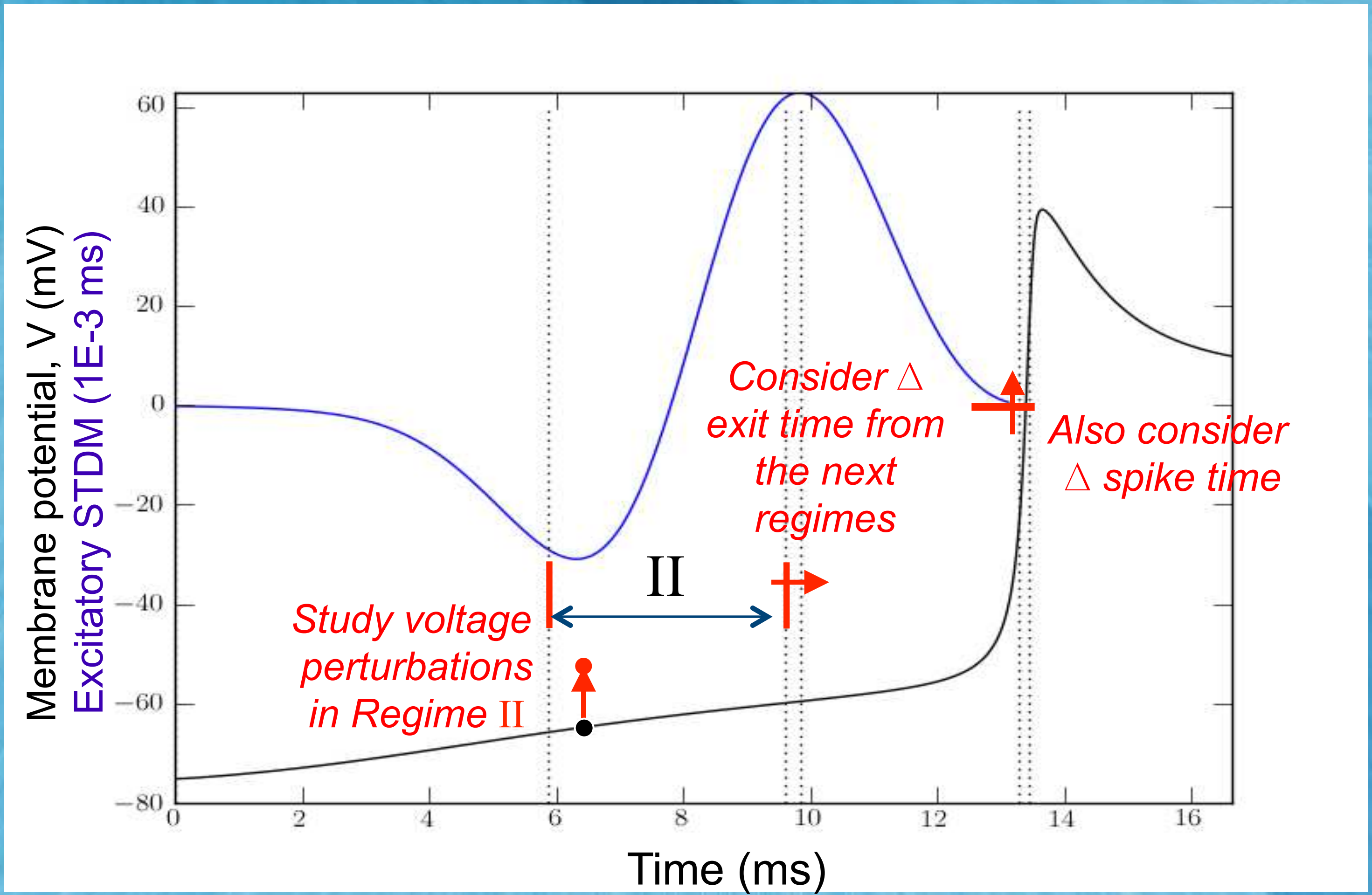
Regime III

$\mathscr{A}_\Psi = \{m, n[S]\}$

Regime IV

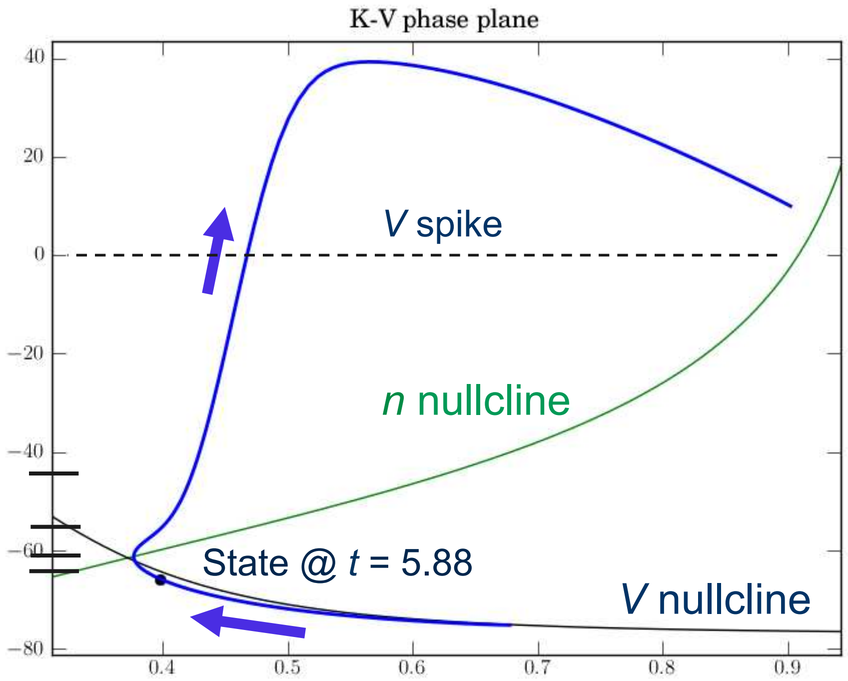
$\mathscr{A}_\Psi = \{n[S], m, h[S]\}$

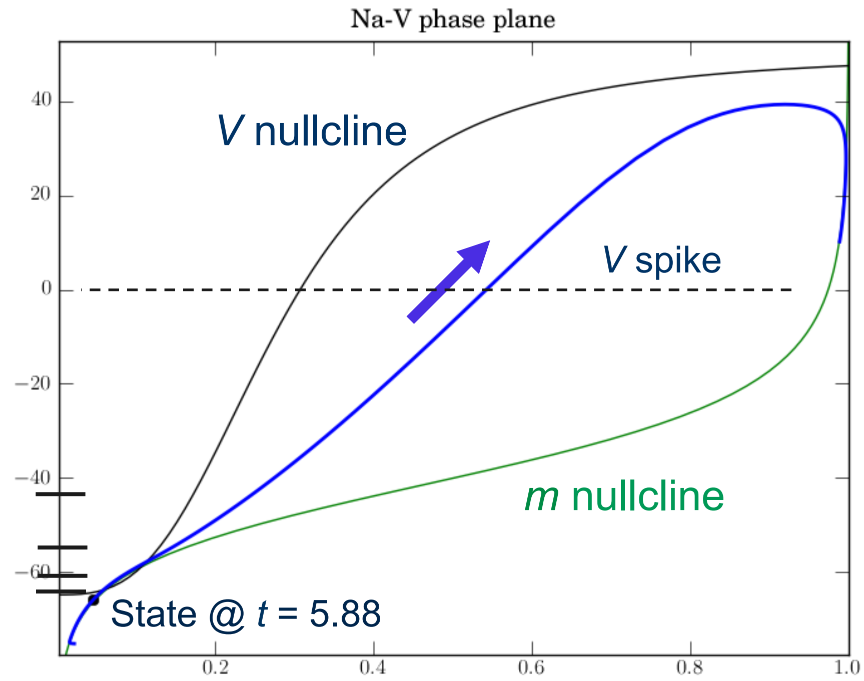
- We can remove Na^+ *h* inactivation (hold it constant) to study STDM mechanism sub-threshold
- Almost no effect on STDM (but model no longer recovers after AP)
- Create a micro-level low-dimensional hybrid reduced model



Initial observations

- At t = 5.88 (start of Regime II, near minimum of PRC)
  - V = -65.8mV
  - Perturb $V \rightarrow V + 0.1$
  - Measure when V crosses thresholds at +2 mV, +5 mV, +10 mV, +20 mV relative to unperturbed orbit



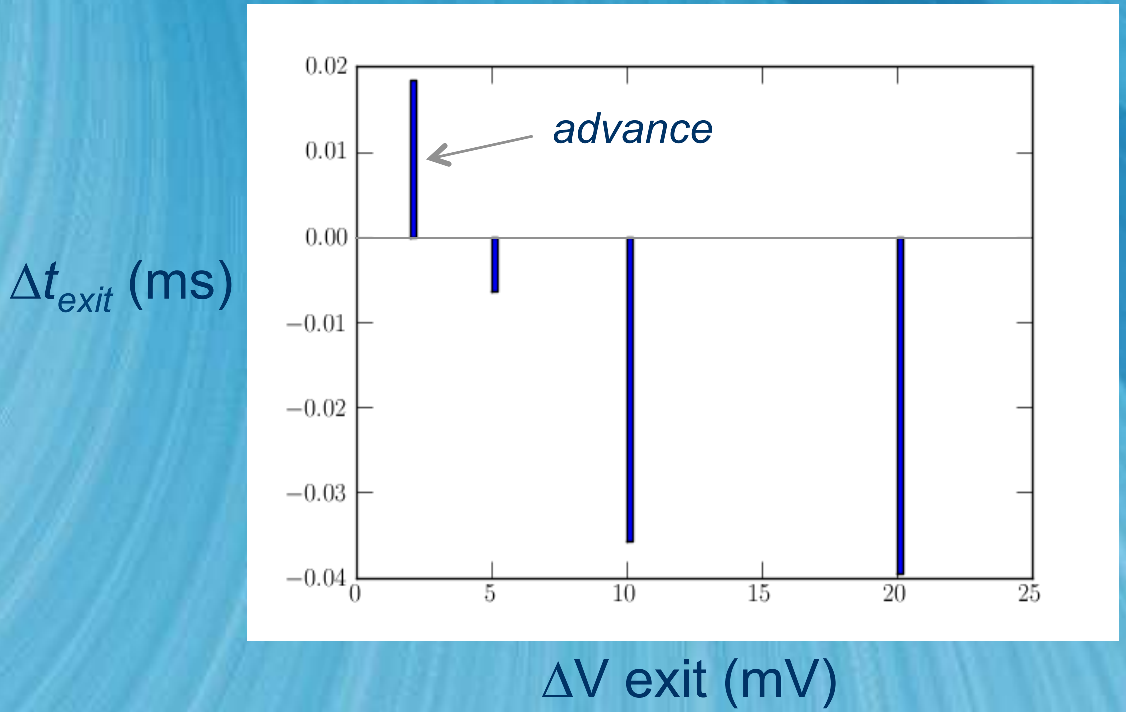


One 'fixed point' in (n, V)

Three 'fixed points' in (m, V)

Initial observations

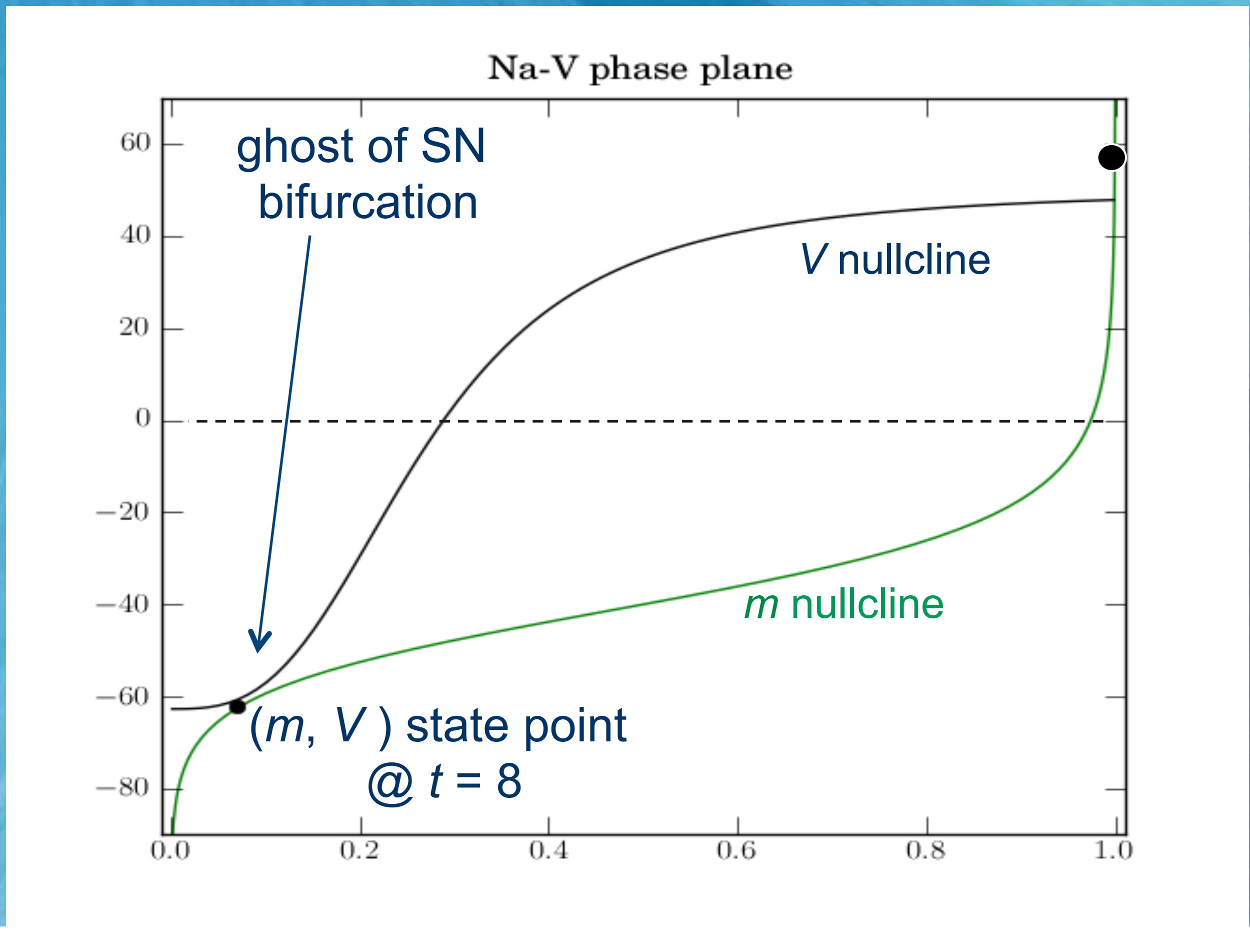
- STDM involves a non-local, non-stationary, non-linear effect:
  - initial phase advance becomes a delay! why?
  - one does not simply look at a variational equation with frozen coefficients around the orbit at t = 5.88   (e.g., see Nick Trefethen's work)



Compare numeric
STDM(5.88) = -0.0397

Local linearization of nullclines

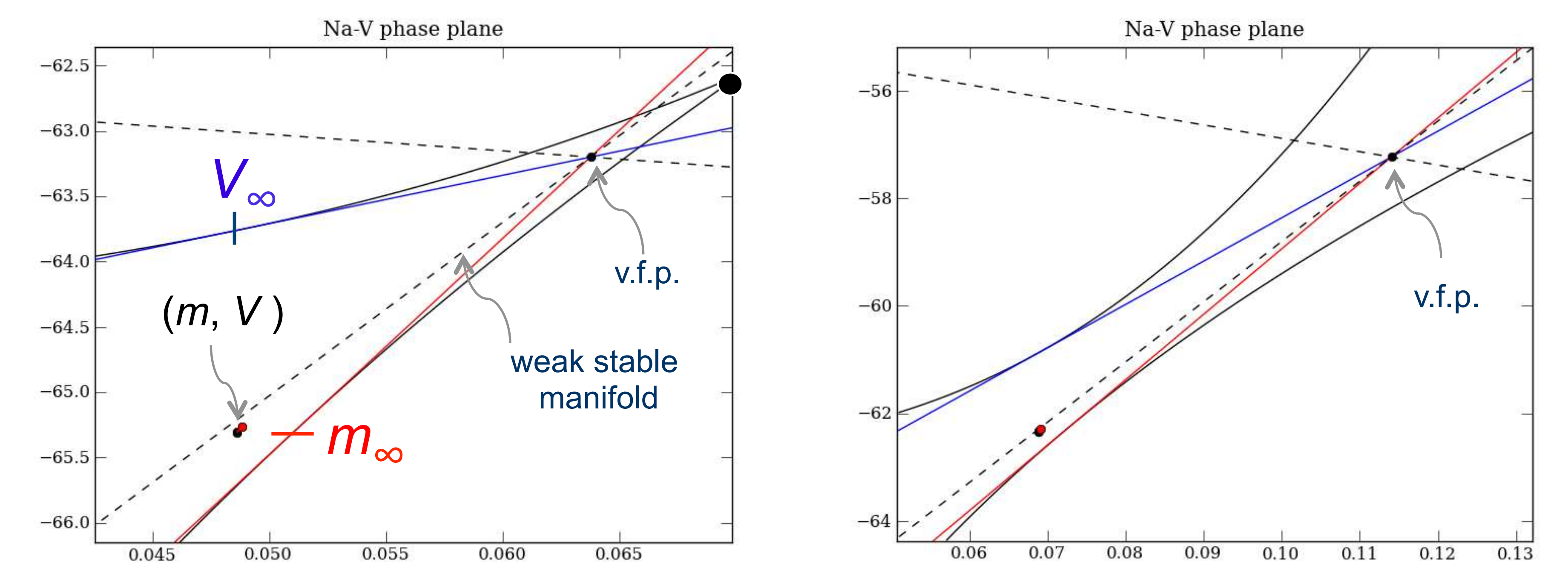
- Create a *virtual* fixed point, valid only while Δ curvature and Δ autonomy remain low
- Step forward in time to new linearization according to smallest relevant time scale



Local linearization of nullclines

- Locally, the weak stable manifold defines the *river* for a virtual node …
  - *e.g., in (m, V) plane*
- $+V$ perturbation has negative feedback effect on decreasing n here => locally phase-delaying

$t \sim 6$ ms $\qquad$ $t \sim 8$ ms

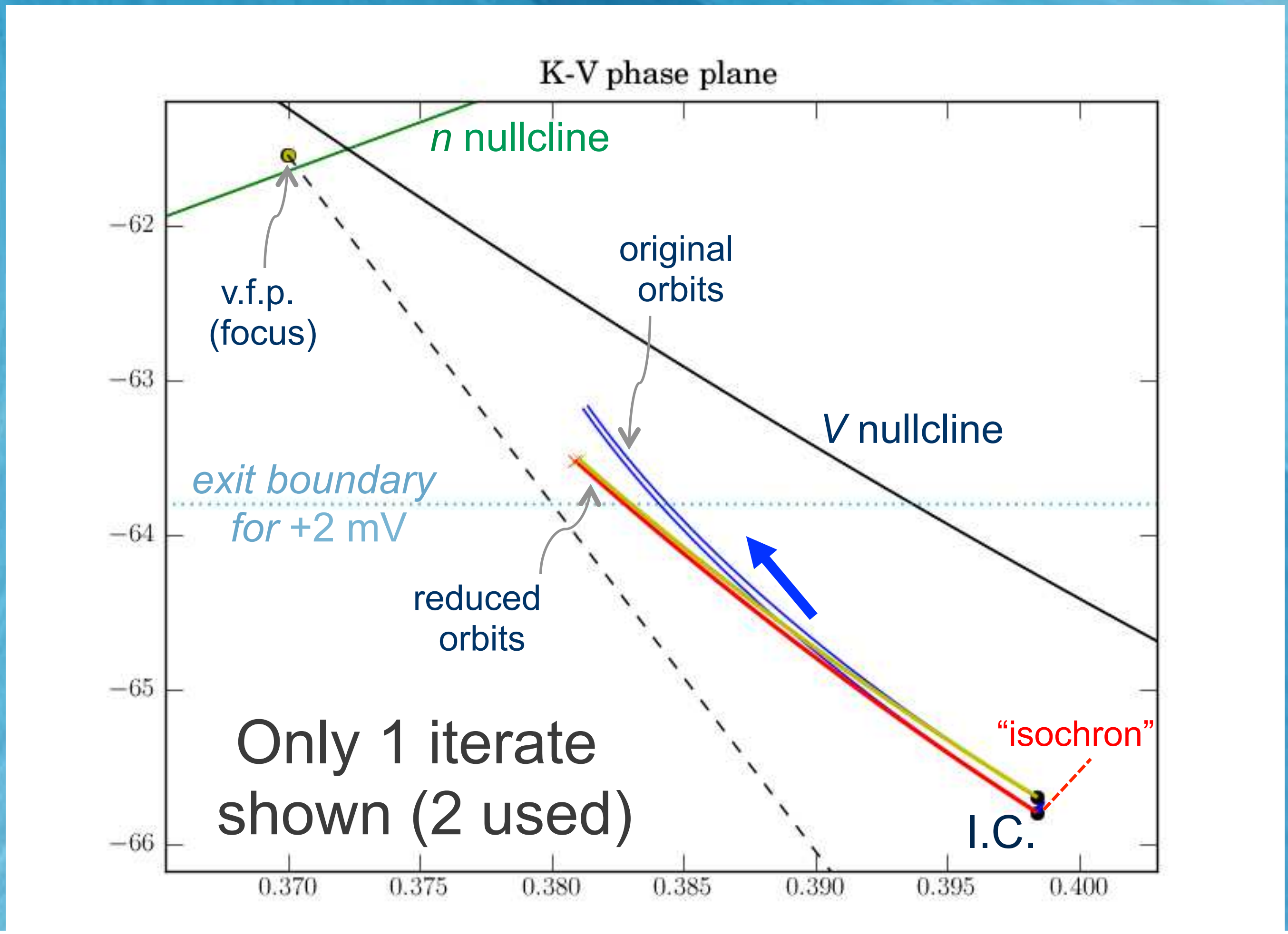


Analytic isochrons for nonlinear systems?

- Nonlinear analysis of isochrons is prohibitively difficult (see Day, Rubin, and Chow)
- Take a piecewise linear approach instead!
  - Must derive on-the-fly for a given system and IC
  - Let the computer bear the burden of tracking self-consistency conditions and numeric details
  - Python + PyDSTool
- We still achieve an intuitive and (approximate) analytic understanding of the mechanisms at work

Coarse-grained estimate part 1

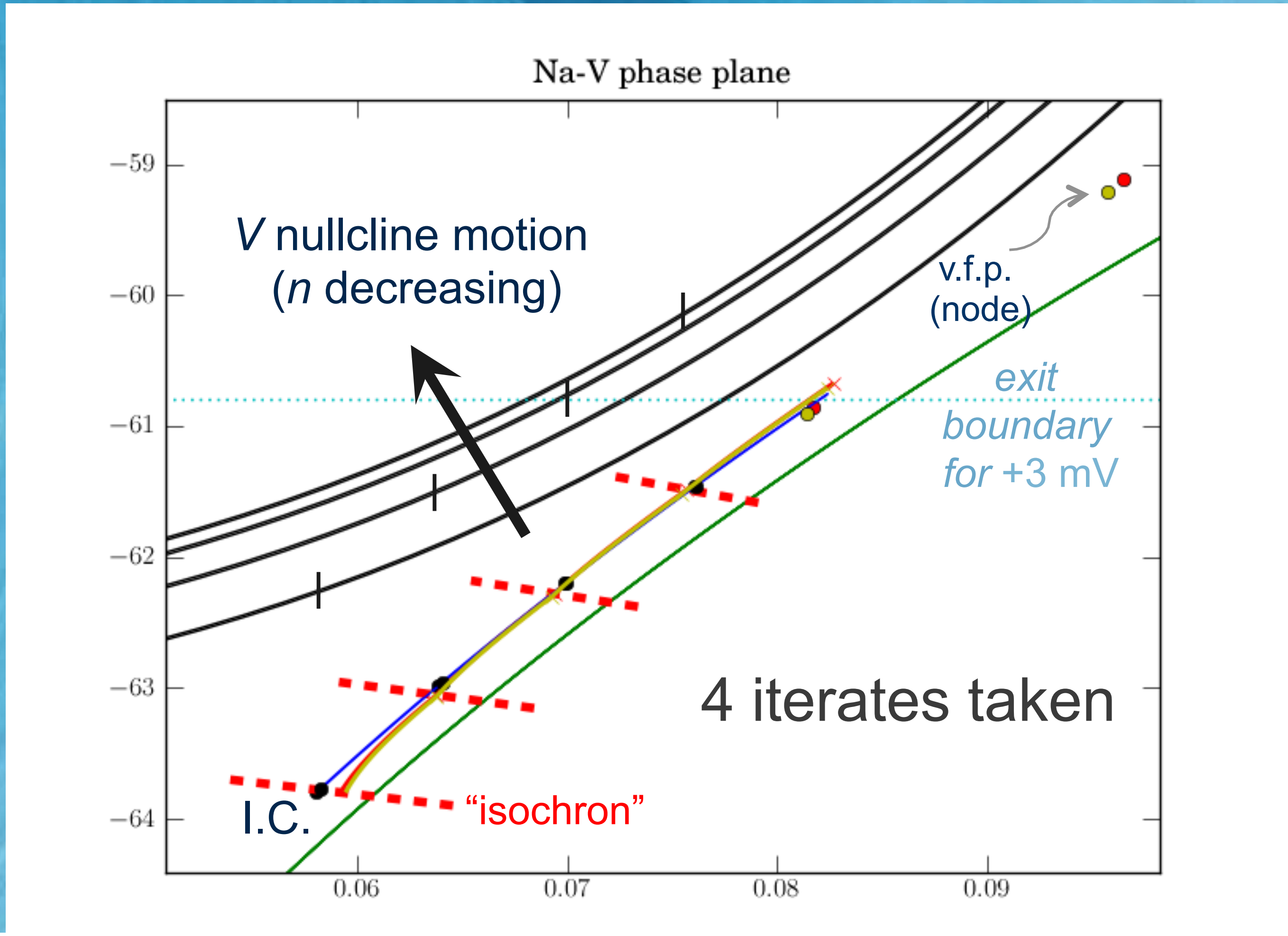
- From t = 5.88 until V increases by 2mV
  - Yellow orbit has + 0.1V initial condition
- (n, V) plane is most autonomous (i.e., most accurate)



Elapsed time ~ 1.2 ms

Coarse-grained estimate part 2

- Continue where left off until *V* increases by 3mV
- (*m*, *V*) plane is most autonomous (most accurate)



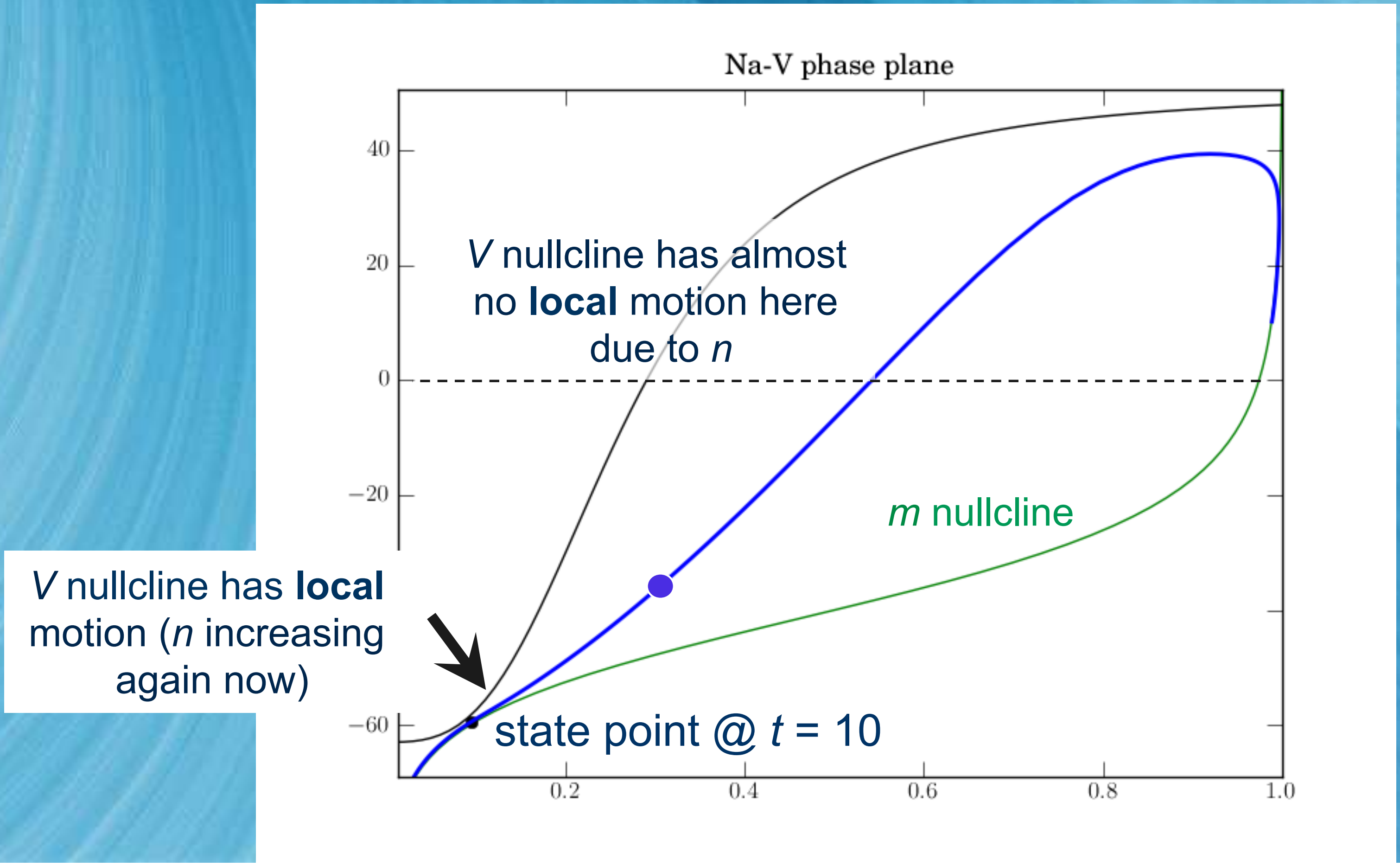
Elapsed time ~ 1.9 ms

Coarse-grained estimate part 3

- Iterating in each sub-regime to exit boundary
  - Error in exit time differences $<$ 0.002 ms
  - Analytically understand origin of this difference
    - micro-to-macro level
- Could continue iterating in expansive part of (m, V)
- Or use "scaffolding principle"
  - Use final iterates as ICs to **original** 4D model
  - See when original model spikes
  - Predicts STDM(5.88) = -0.0355 (vs. -0.0397 actual)
- Other insights
  - Dynamics up to +5mV sets both states to be almost identical
  - Here, difference *only* in n of +1E-5 for perturbed orbit yields STDM = -0.04

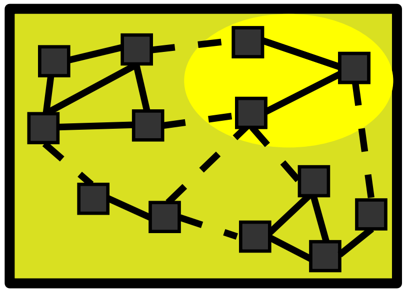
Expansive region – analysis past ghost of saddle-node bifurcation

- +*V* perturbations out here have almost stationary virtual fixed-point *repeller* behind them
- positive feedback effect on increasing *n* is too slow to matter!



Multi-level modeling methodology

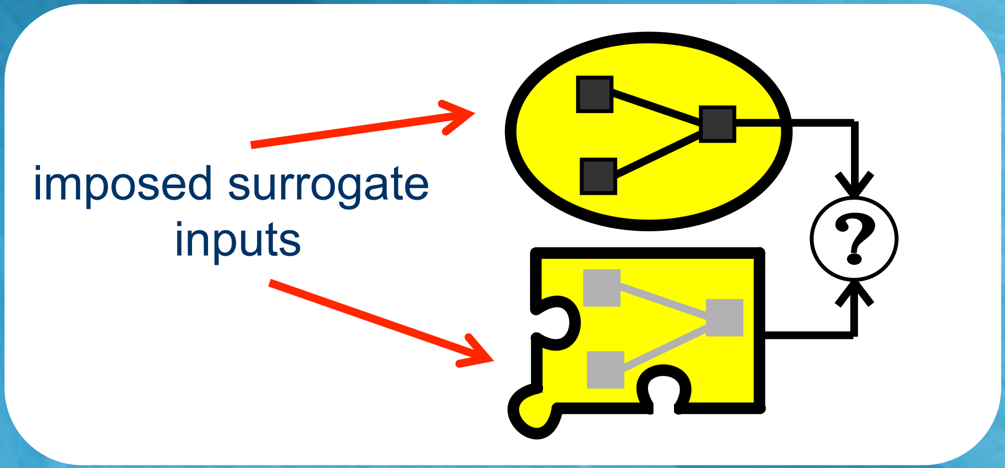
Reverse engineering for optimal parsimony



- Ask a specific question that your model should solve
- Make observations from experimental / simulation data
- Mine data for important features / modules / motifs
  - Sensitivity analysis, bottlenecks, etc.
- Hypothesize high-level conceptual model for them
  - Data-driven constraints
  - Hypothesis-driven constraints

Multi-level modeling methodology

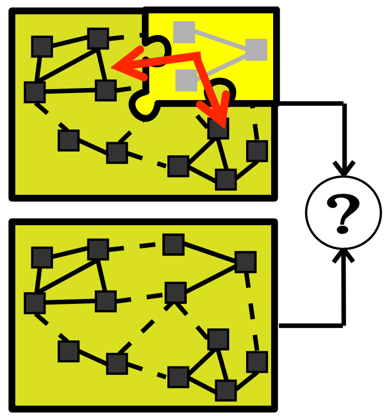
Reverse engineering for optimal parsimony



- Impose the hypotheses to one motif ("open the loop")
- Reduce the motif and determine local validity constraints
  - Open loop configuration
- What worked and what didn't re. original question?
  - Compare *qualitative* features (avoid over-fitting)
  - Diagnostics and optimization to fine-tune

Multi-level modeling methodology

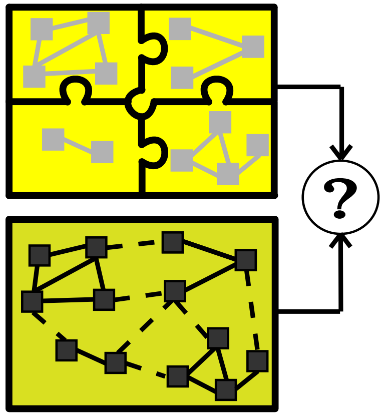
Reverse engineering for optimal parsimony



- Re-contextualize (non-trivial without good software)
- Impose the hypotheses on the rest of the system, by replacing model parts with reduced ones
- What worked, and what didn't re. original question?
  - More diagnostic and optimization steps

Multi-level modeling methodology

Reverse engineering for optimal parsimony



- Continue process and build entire, *hybrid* model
  - Heterogeneous abstractions at multi-levels
  - Many inter-related consistency constraints
  - Re-closes loop
- I call this a “scaffolding” approach

An aspirational methodology

- Seek conceptual models for *causes at all levels*
- Build working micro-theories, test their assumptions and predictions against data *and* functional hypotheses
- Systematic, rational derivation of high-level emergent properties from low-level details in a complex system
  - Semi-analytic, semi-numeric
- Use informatic software tools to
  - manage derivations
    - optimization, data fitting
    - enumerate combinations of possibilities
    - track consistency conditions
    - inference and qualitative reasoning
  - navigate the data produced
  - create more meaningful meta-data about models

PyDSTool dynamical systems software

- Built and simulated hybrid models
  - Fast C-code integrators w/ auto-code generation and event detection
  - Interfaces to AUTO via PyCont
  - Model-building utilities / symbolic expressions
    - E.g. for symbolic Jacobians
  - Templates for conductance-based neural models
- Phase plane analysis objects and tools
- Optimization and inference tools
- Advanced interactive scripting through Python
  - Mathematically-meaningful classes
  - Rich meta-data
- pydstool.sf.net

Overview of PyDSTool

Model building

- Symbolic manipulation
- Macros / constructors
- Hierarchical structures
- Intelligent editing tools

Simulation

- Maps
- ODEs
- DAEs
- Hybrid models
- (Precise events)
- (Arbitrary inputs)

Model analysis

- Bifurcation analysis
- Sensitivity analysis
- Reduction analysis
- Phase plane
- Feature detection etc.
- PRC (adjoint and direct)

Data analysis

- Subspace ID (e.g. PCA)
- Fractal dimension
- Feature detection

Optimization

- Parameter estimation
- Qualitative reasoning
- Inference

PyDSTool dynamical systems software

- Since 2006, hosted at pydstool.sourceforge.net
  - Tutorial, installation instructions, documentation
- Basic features:
  - Entirely scripted / command line (like Matlab)
  - Index-free variables equations
    - Can use indices for macro structure
  - Context heavy (e.g., description/tag/label attributes)
  - Equations easily defined by constructors or directly as strings
  - State/time events, auxiliary "helper" functions, external inputs, Jacobians, special points
  - Fast ODE/DAE solvers + AUTO interface built in
    - C RHS of your vector field is automatically produced!
    - Essentially as fast as native C code!