

# Advanced Computational Methods for Biological Systems

Faranak Rajabi



Department of Mechanical Engineering,  
University of California, Santa Barbara

Candidacy Talk  
December 5th, 2024

# Presentation Overview

## 1 Optimal Control for Stochastic Neural Oscillators

- Motivation and Background
- Optimal Control Approach
- Model
- Optimization Problem
- Numerical Methods
- Results
- Robustness to Network Properties
- Conclusion
- Ongoing and Future Work

## 2 Protein Aggregation

- Motivation and Background
- Protein Aggregation Physics
- Continuum Model
- Primarily Results
- Ongoing and Future Work

# Acknowledgments



Dr. Frederic Gibou  
UCSB



Dr. Jeff Moehlis  
UCSB

Based on work in

F. Rajabi, F. Gibou, and J. Moehlis. [Optimal control for neural oscillators](#).

*Journal of Biological Cybernetics*, 2024, under review

# Motivation and Background

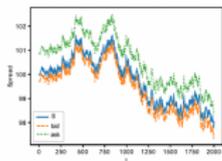
## Stochastic-PDEs

**Most Engineering problems Can be framed as mathematical modeling challenges**

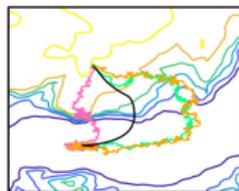
**PDEs that lack analytical solutions  
Numerical methods to the rescue**

Stochastic Hamilton-Jacobi Equations are applied to:

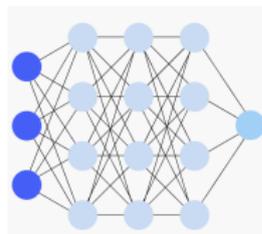
- Optimal control for robotics and autonomous systems
- Financial modeling under uncertainty
- Enhancing reinforcement learning and neural networks
- Modeling biological oscillators and dynamics under stochastic influence



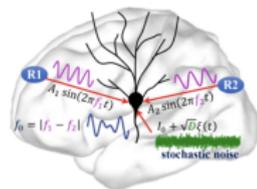
Finance



Path Planning



Reinforcement Learning



Biological Systems

# Motivation and Background

## Parkinson's Disease

### Hypothesis

Pathological synchronization of neuronal activity is linked to neurological disorders like Parkinson's disease, essential tremor, and epilepsy.

#### **Idea:**

Modulate this synchronization is key for effective treatments like Deep Brain Stimulation (DBS).

#### **Challenges:**

- Diminishing efficacy over time
- Stimulation-induced side effects
- High energy consumption
- Static parameters vs. dynamic symptoms

# Motivation and Background

## Advances in Control Theory for DBS

- Control theory addresses DBS limitations like diminishing efficacy and side effects.
- Closed-loop feedback systems adjust stimulation based on real-time signals, improving symptom control.
- Adaptive DBS (aDBS) personalizes treatment by modulating stimulation in response to symptom fluctuations.
- Model Predictive Control (MPC) and Reinforcement Learning (RL) optimize stimulation, improving energy efficiency and therapeutic efficacy.

# Proposed Approach

## Stochastic HJB for Desynchronization

### **Our Approach:**

Event-based optimal control based on phase resetting technique to desynchronize neurons, incorporating stochastic elements to manage neural fluctuations.

- Designs optimal control based on the stochastic Hamilton-Jacobi-Bellman (HJB) Equation
- Extends Nabi et al.'s optimal control strategy, integrating randomness in neural dynamics
- Uses BSDEs and viscosity solutions for robust optimization.
- Event-based control strategy
- Computational framework is adaptable to other stochastic control problems.

A. Nabi, M. Mirzadeh, F. Gibou, and J. Moehlis. [Minimum energy desynchronizing control for coupled neurons.](#)

*Journal of Computational Neuroscience*, 34:259–271, 2013

# Model Overview

## Neuronal Dynamics and Control

### Reduced 2D Hodgkin-Huxley Model

The neuronal model is described by:

$$\dot{V}_i = f_V(V_i, n_i) + \eta_i(t) + \frac{1}{N} \sum_{j=1}^N \alpha_{ij} (V_j - V_i) + u(t),$$

$$\dot{n}_i = f_n(V_i, n_i).$$

- $i = 1, \dots, N$ : Index of neurons in a network of  $N$  neurons.
- $u(t)$ : Control input applied uniformly to the network.
- $V_i, n_i$ : Membrane voltage and gating variable of neuron  $i$ .
- $f_V, f_n$ : Dynamics in the absence of noise, coupling, or control.
- $\eta_i(t)$ : Gaussian white noise,  $\sqrt{2D}\mathcal{N}(0, 1)$ , modeling neurons intrinsic noise.

# Model Overview

## State-space Diagram for Deterministic System

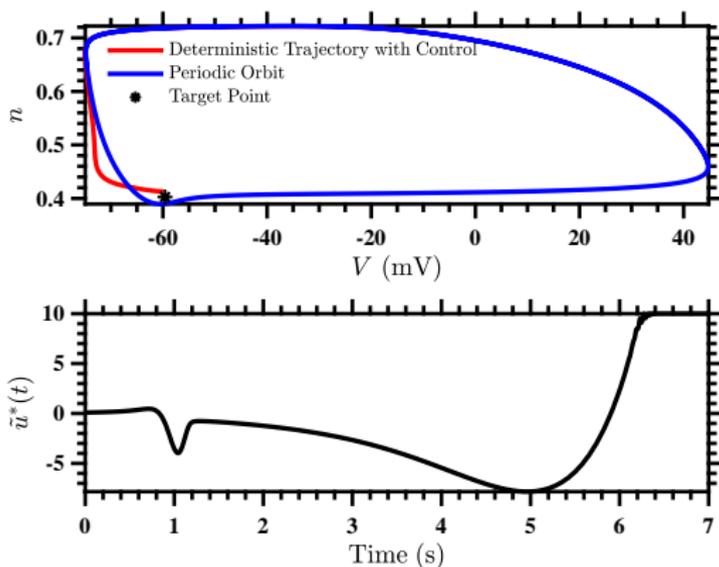


Figure: State-space diagram and control for the deterministic system

A. Nabi, M. Mirzadeh, F. Gibou, and J. Moehlis. [Minimum energy desynchronizing control for coupled neurons.](#)

*Journal of Computational Neuroscience*, 34:259–271, 2013

# Model Overview

## State-space Diagram for Stochastic System

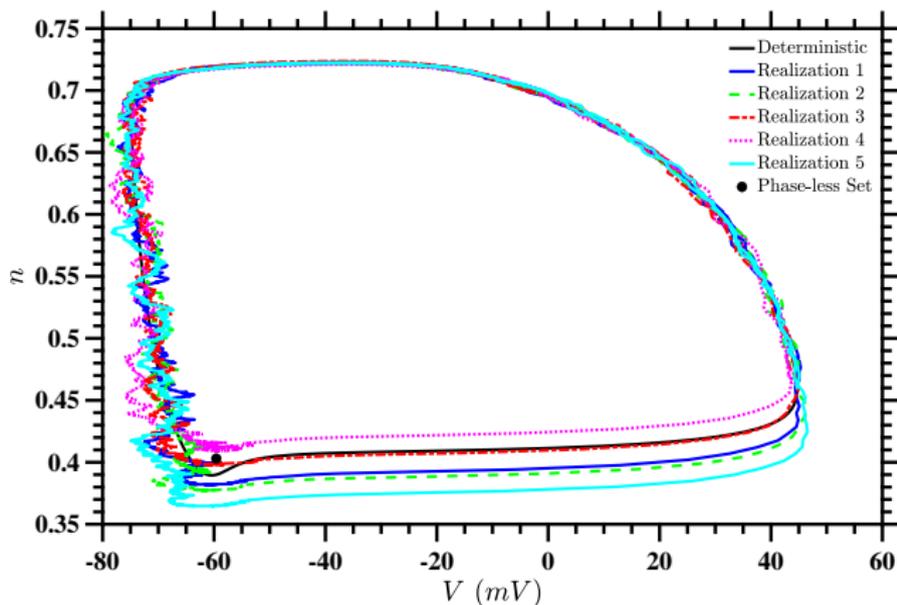


Figure: State space diagram for  $D = 15$  without control input

# Optimization Problem

## Control Objective

### **Control Objective:**

Find the energy-optimal control law to steer the system to  $(V_{pl}, n_{pl})$  within time  $[0, T_{end}]$  while minimizing the following cost function:

### Cost Function

$$J(z, u(t)) = \int_0^{T_{end}} u^2(t) dt + \gamma q(z(T_{end})),$$

where:

- $u(t)$ : Control input, bounded by  $|u| \leq u_{\max}$ .
- $\gamma q(z(T_{end}))$ : Penalizes the deviation from the desired final state, representing the final-time penalty for being far from the phaseless set.

# Optimization Problem

## Hamilton-Jacobi-Bellman Equation

The stochastic cost-to-go function,  $\mathcal{V}(z, \tau)$ , represents the minimum expected cost to reach the target state while satisfying control constraints:

### Cost-to-Go Function

$$\mathcal{V}(z, \tau) = \min_{u(\cdot)} \mathbb{E} \left[ \int_{\tau}^{T_f} u^2(t) dt + \gamma q(z(T_f)) \right],$$

subject to the control constraint  $|u(t)| \leq u_{\max}$ .

This function satisfies the stochastic Hamilton-Jacobi-Bellman equation:

$$-\frac{\partial \mathcal{V}}{\partial \tau} = \min_{|u| \leq u_{\max}} \mathcal{H}(z, \nabla \mathcal{V}) + D\Delta \mathcal{V},$$

where:

- $\mathcal{H}(z, \nabla \mathcal{V})$  is the deterministic Hamiltonian.
- $D\Delta \mathcal{V}$  represents the diffusion term, with  $\Delta$  being the Laplacian operator.

# Optimization Problem

## Hamiltonian and Optimal Control

### Hamiltonian

The Hamiltonian governing the system is given by:

$$\mathcal{H}(z, \nabla \mathcal{V}, u) = u^2 + \nabla \mathcal{V}(z(t), t) (F(z(t)) + Bu(t)),$$

where  $F(z(t))$  and  $B$  represent the system dynamics and control input coefficients, respectively.

The optimal control law is derived from the Hamiltonian and is given by:

$$u^*(t) = \begin{cases} -\frac{1}{2K} \mathcal{V}_x & |\mathcal{V}_x| \leq 2Ku_{\max} \\ -\text{sign}(\mathcal{V}_x)u_{\max} & |\mathcal{V}_x| > 2Ku_{\max} \end{cases}$$

where  $\mathcal{V}_x = \frac{\partial \mathcal{V}}{\partial x}$ .

# Numerical Methods

## Stochastic HJB Equation

To solve the optimal control problem, we address the stochastic HJB equation:

### Stochastic HJB Equation

$$\frac{\partial \mathcal{V}}{\partial t} + \min_{|u| \leq u_{max}} \mathcal{H}(z, \nabla \mathcal{V}, u) + \frac{D}{K^2} \frac{\partial^2 \mathcal{V}}{\partial x^2} = 0,$$

with a terminal cost defined as:

### Terminal Condition

$$\mathcal{V}(z(T_{end}), T_{end}) = \gamma \left( 1 - e^{-\left( \frac{(x-x_{pl})^2}{\sigma_x^2} + \frac{(y-y_{pl})^2}{\sigma_y^2} \right)} \right),$$

### Key considerations:

- The problem is a terminal value problem provided by the end-point cost  $\mathcal{V}(z(T_{end}), T_{end})$ .
- The equation must be solved backward in time from  $T_{end}$  to  $t = 0$ .

# Numerical Methods

## Convergence Criteria for BSPDEs

### Theorem (Barles-Souganidis Framework)

*For a Backward Stochastic Partial Differential Equation (BSPDE), numerical methods must satisfy the following properties to guarantee convergence to the viscosity solution:*

- 1 **Monotonicity:** *Ensures a discrete maximum principle, preventing instability.*
- 2 **Consistency:** *The numerical method must accurately approximate the BSPDE as the grid spacing tends to zero.*
- 3 **Stability:** *Maintains bounded numerical solutions over time.*

# Numerical Methods

## Proposed Method

We employ Hamilton-Jacobi (HJ) solvers leveraging advanced numerical techniques.

### **First-order Term:**

- Weighted Essentially Non-Oscillatory (WENO) for  $\nabla\mathcal{V}$
- Local Lax-Friedrichs (LLF) for *numerical Hamiltonian* ( $\hat{\mathcal{H}}$ ) which is needed for nonlinear  $\mathcal{H}$

### **Second-order Term:**

- Backward Time Central Space (BTCS) for  $\nabla^2\mathcal{V}$

### **Time Integration:**

- Total Variation Diminishing Runge-Kutta (TVD-RK) for time integration

# Numerical Methods

## Sequential Operator Splitting Method

**Operator Splitting Concept:** For a differential equation where the local time derivative equals the sum of several operators, the operator splitting method allows us to treat each spatial operator separately, solving each corresponding sub-problem individually.

### Abstract Cauchy Problem

Consider the Cauchy problem on Banach space  $X$ :

$$\begin{cases} \frac{d}{dt}u(t) = (A + B)u(t), & t > 0 \\ u(0) = u_0 \end{cases}$$

where  $A, B$  are densely defined, linear, closed operators with  $D(A) \cap D(B)$  dense in  $X$

# Numerical Methods

## Sequential Operator Splitting Method

### Sub-Problem Structure

For  $t \in [(k-1)h, kh]$  with  $k \in \mathbb{N}$  and  $u_{spl,h}(0) = u_0$ , solve:

$$\begin{aligned} \frac{d}{dt} u_A(t) &= Au_A(t), & u_A((k-1)h) &= u_{spl,h}((k-1)h) \\ \frac{d}{dt} u_B(t) &= Bu_B(t), & u_B((k-1)h) &= u_A(kh) \end{aligned}$$

Set  $u_{spl,h}(kh) = u_B(kh)$

### Split Solution Form

The numerical solution takes the form:

$$u_{spl,h}(t) = (F(h))^n u_0$$

for all  $u_0 \in D$  and  $n \in \mathbb{N}$  with  $h = \frac{t}{n}$

# Numerical Methods

## Sequential Operator Splitting Method

**IMEX Methods:** Suitable combinations of implicit and explicit schemes for PDEs with terms of different nature.

### PDE Setting

Start with PDE:

$$\begin{cases} u_t(x, t) = f(x, u(x, t)) \\ u(x, 0) = u_0(x) \end{cases}$$

Split into  $f = f_1 + f_2$ , yielding system  $\frac{d}{dt}U = F_1U + F_2U$

### Numerical Approximations

For each equation  $u_t = f_k(u)$ ,  $k = 1, 2$ :

Explicit Euler:  $u^{n+1} = \psi_k^e(u^n) \equiv u^n + hf_k(u^n)$

Implicit Euler:  $u^{n+1} = \psi_k^i(u^n) \equiv u^n + hf_k(u^{n+1})$

# Application to PDEs

## IMEX Implementation

### Semi-Discrete System

After splitting  $f = f_1 + f_2$ :

- $f_1$ : diffusion term (requires implicit treatment)
- $f_2$ : nonlinear term (suitable for explicit integration)

Examples: convection-diffusion, reaction-diffusion problems

### IMEX Scheme

Linear one-step scheme combining methods:

$$u^{n+1} = \varphi_3^h(u^n) = \psi_1^i \circ \psi_2^e(u^n) = u^n + h(f_1(u^{n+1}) + f_2(u^n))$$

For convergence analysis, see Ascher et al. (1997)

# Numerical Methods

## Algorithm Overview

### Algorithm Steps

- 1 Initialize with the terminal cost  $\mathcal{V}(z, T_{end})$
- 2 Solve the Hamiltonian term:
  - Compute gradients using WENO scheme
  - Evaluate the Hamiltonian with LLF method
  - Output is  $\mathcal{V}^*(z, T_{end} - \Delta t)$
- 3 Solve the diffusion term:
  - Use the computed  $\mathcal{V}^*(z, T_{end} - \Delta t)$
  - Apply BTCS implicit scheme
  - Output is  $\mathcal{V}(z, T_{end} - \Delta t)$
- 4 Time integration stage:
  - Repeat steps 2 and 3 for each stage of the TVDRK
- 5 Repeat steps 2, 3 and 4 until  $t = 0$

# Numerical Methods

## Model Setup

### Single Neuron Model

After computing the value function, we integrate the stochastic ODE:

$$\begin{aligned}\dot{V} &= f_V(V, n) + \eta(t) + u(t) \\ \dot{n} &= f_n(V, n)\end{aligned}$$

where noise term  $\eta(t) = \sqrt{2D}\mathcal{N}(0, 1)$

### Simulation Parameters

- Time horizon:  $T_{end} = 7$  ms
- Spatial grid:  $320 \times 320$  uniform grid
- Control bound:  $u_{max} = 10 \mu\text{A}/\mu\text{F}$
- Honeycutt's second-order method for SDE integration

# Numerical Methods

## Model Setup

### Single Neuron Model

After computing the value function, we integrate the stochastic ODE:

$$\begin{aligned}\dot{V} &= f_V(V, n) + \eta(t) + u(t) \\ \dot{n} &= f_n(V, n)\end{aligned}$$

where noise term  $\eta(t) = \sqrt{2D}\mathcal{N}(0, 1)$

### **Simulation Parameters:**

- Time horizon:  $T_{end} = 7$  ms
- Spatial grid:  $320 \times 320$  uniform grid
- Control bound:  $u_{max} = 10 \mu A / \mu F$
- SDE Integration: Honeycutt's second-order method

# Numerical Methods

## Optimal Control Computation

### *Control Computation Strategies:*

$u^*(t)$ :

**Stochastic value function on stochastic trajectories**

Energy-optimal for specific noise realization

$\tilde{u}^*(t)$ :

**Stochastic value function on deterministic trajectories**

Approximates ensemble average behavior

$u_0^*(t)$ :

**Deterministic value function on stochastic trajectories**

Applies noise-free control to noisy system

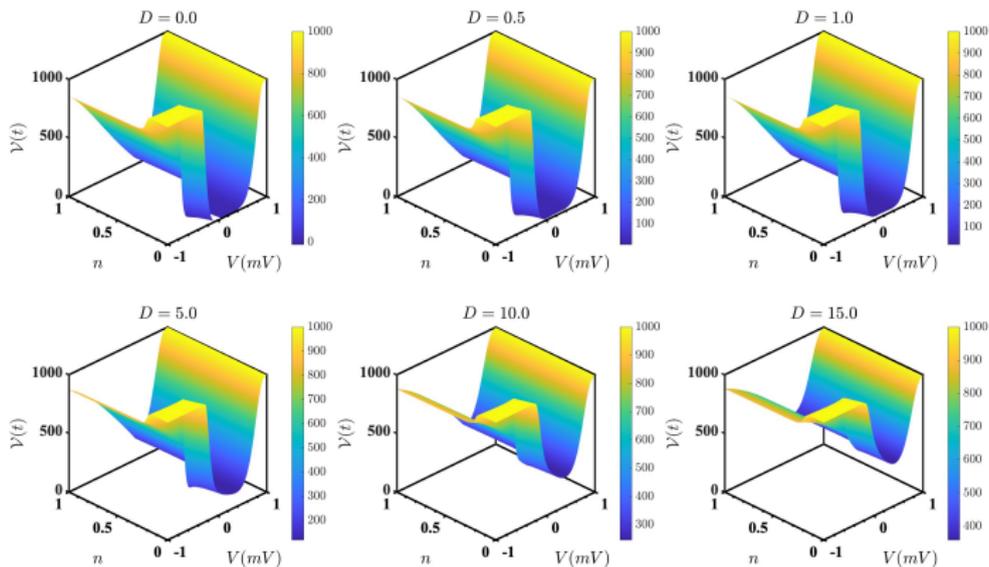
$u_d^*(t)$ :

**Deterministic value function on deterministic trajectories**

Classical approach (Nabi et al., 2013)

# Results for Single-Neuron Level

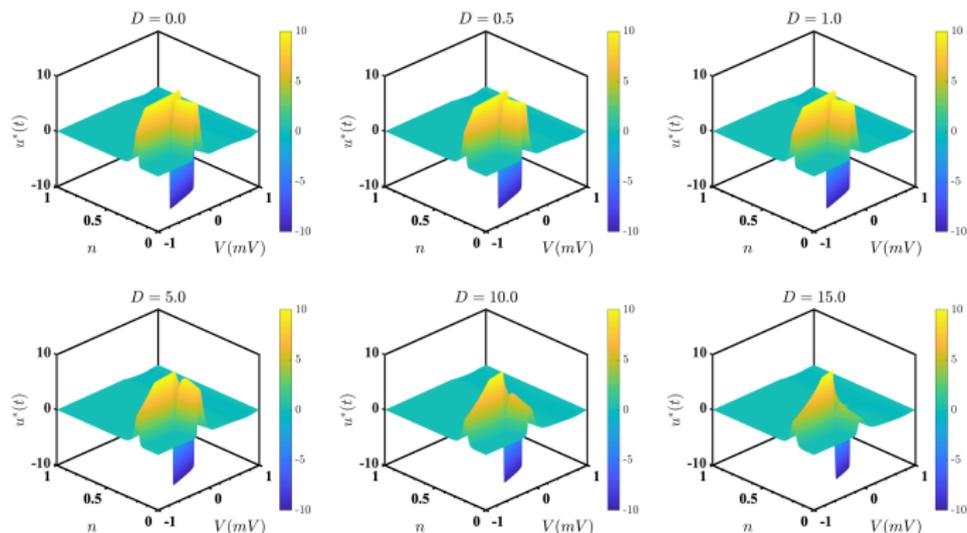
## Effects of Noise on System Dynamics



**Figure:** Cost-to-go function  $\mathcal{V}$  for various noise levels  $D$ . For small  $D$ ,  $\mathcal{V}$  is steeper and approaches zero, while as  $D$  increases,  $\mathcal{V}$  rises with a less steep slope, indicating noise influence on the dynamics

# Results for Single-Neuron Level

## Effects of Noise on System Dynamics



**Figure:** Optimal control input  $u^*$  for various noise levels  $D$ . The control input becomes more pronounced as  $D$  increases, reflecting system dynamics under varying noise.

# Results for Single-Neuron Level

## Single Realization of Noise

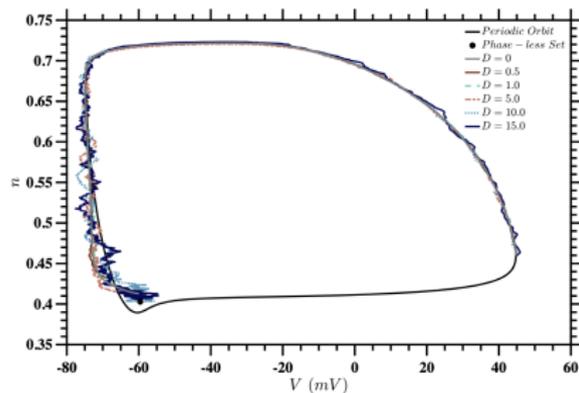
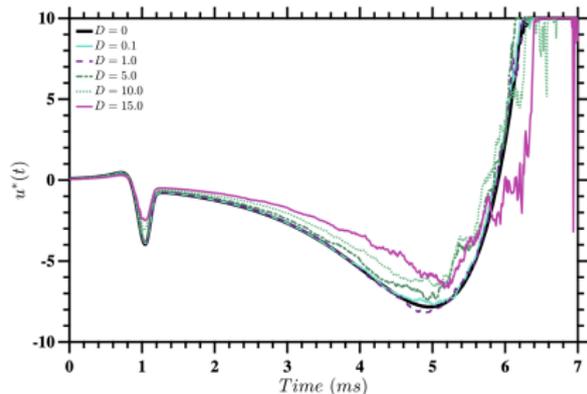
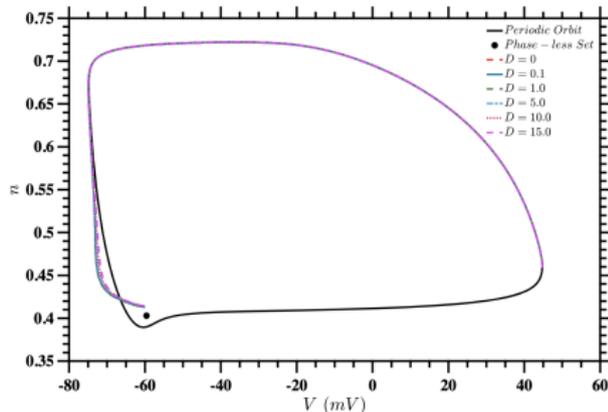
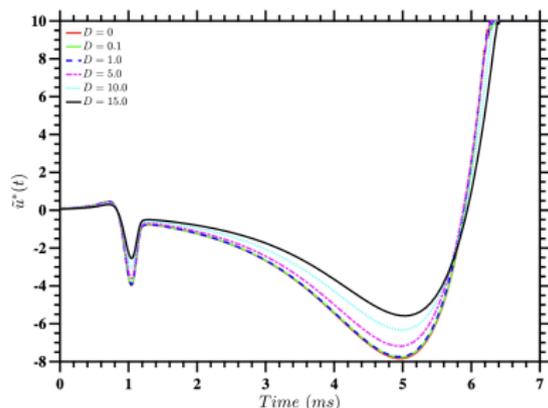


Figure: Stochastic control inputs and stochastic trajectories

# Results for Single-Neuron Level

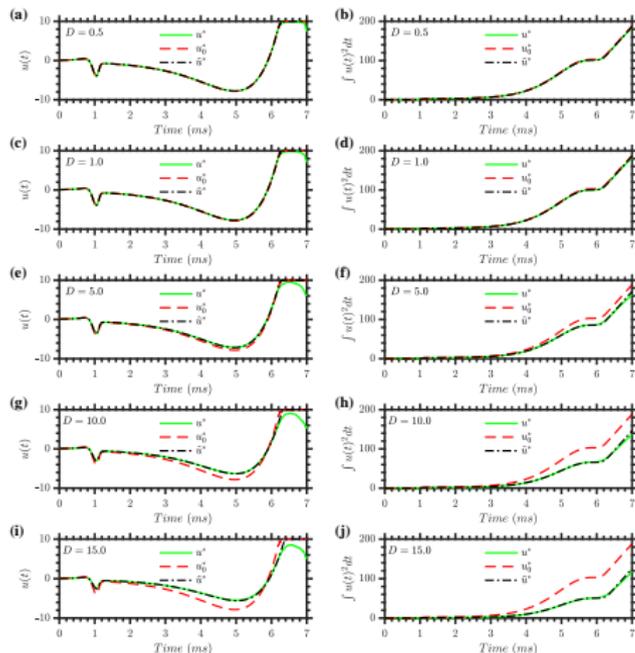
## Stochastic Value Function on Deterministic Trajectories



**Figure:** Control inputs derived from the stochastic value function on deterministic trajectories.

# Results for Single-Neuron Level

## Comparison of Energy Consumption Across Different Noise Levels



**Figure:** Control inputs and energy consumption with expected value integrals,  $\mathbb{E}[\int u^2(t)]$ , based on 10,000 Monte Carlo simulations.

# Results for Population Level

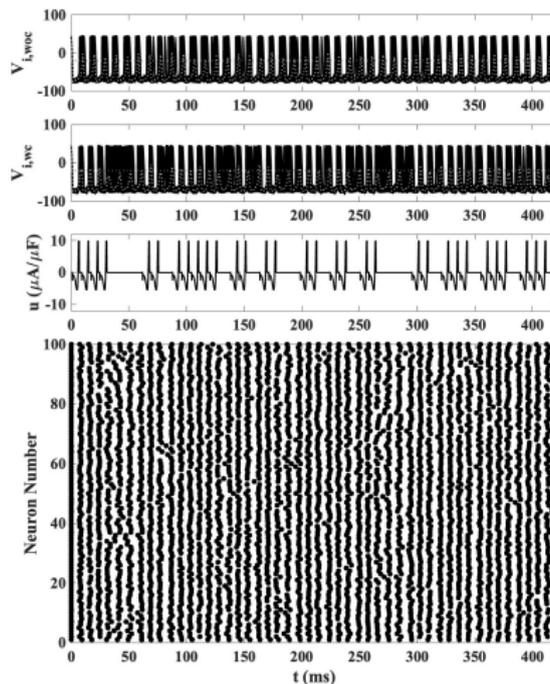
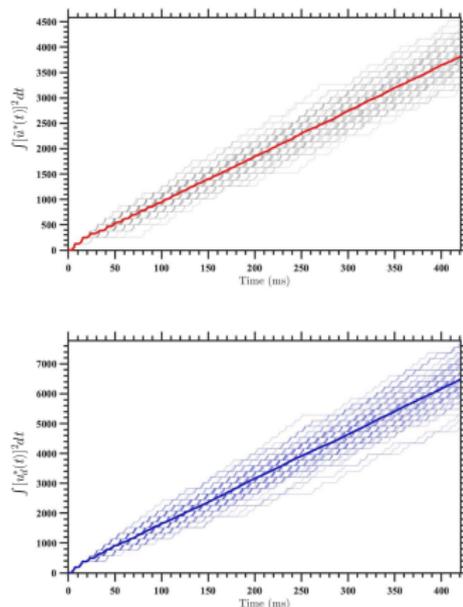


Figure: Stochastic control for  $D = 15.0$ ,  $\alpha = 0.25$ ,  $N = 100$  neurons: (top) without control, (second) with control, (third) input, (bottom) spike raster.

# Results for Population Level

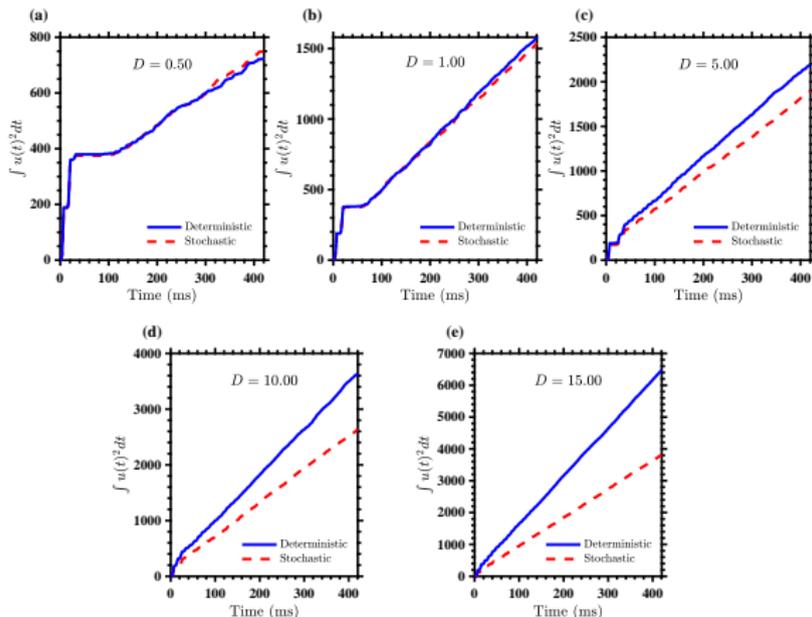
Realizations for  $D = 15$ ,  $\alpha_{ij} = 0.25$



**Figure:** Individual realizations and mean trajectories for stochastic (top) and deterministic (bottom) optimal control at  $D = 15$ .

# Results for Population Level

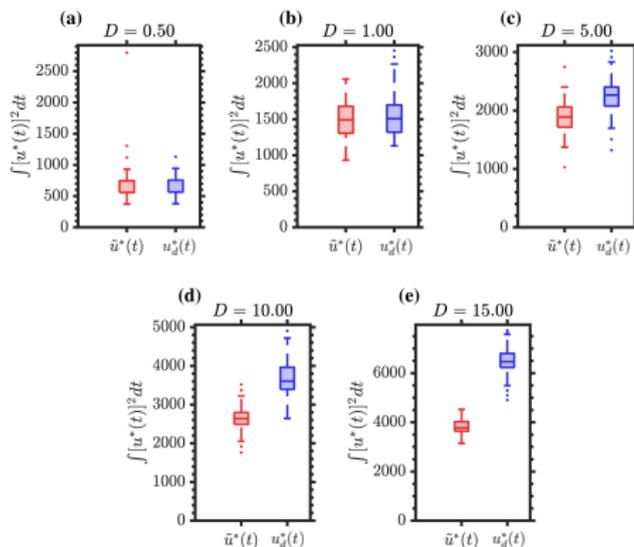
## Cumulative Energy Expenditure Comparison



**Figure:** Cumulative energy expenditure for deterministic (blue) and stochastic (red) control across different noise intensities  $D$  and coupling strengths  $\alpha$ .

# Robustness to Network Properties

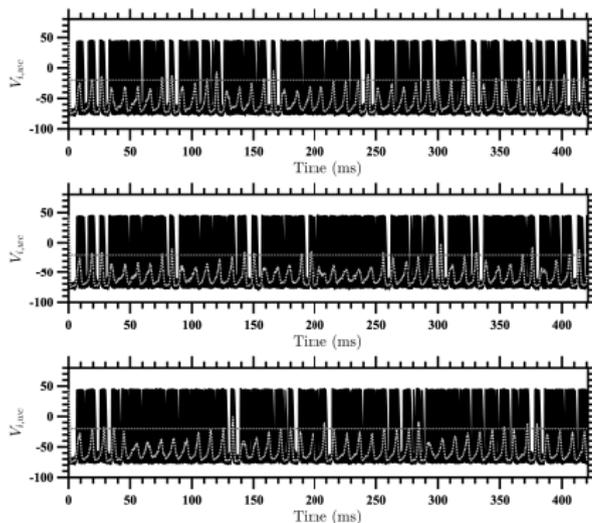
## Energy Distribution for Control Strategies



**Figure:** Accumulated energy expenditure for deterministic (blue) and stochastic (red) control across different noise levels  $D$  and coupling strengths  $\alpha$ . The boxes show the distribution of  $\int_0^{425} [u^*(t)]^2 dt$  for 100 noise realizations.

# Robustness to Network Properties

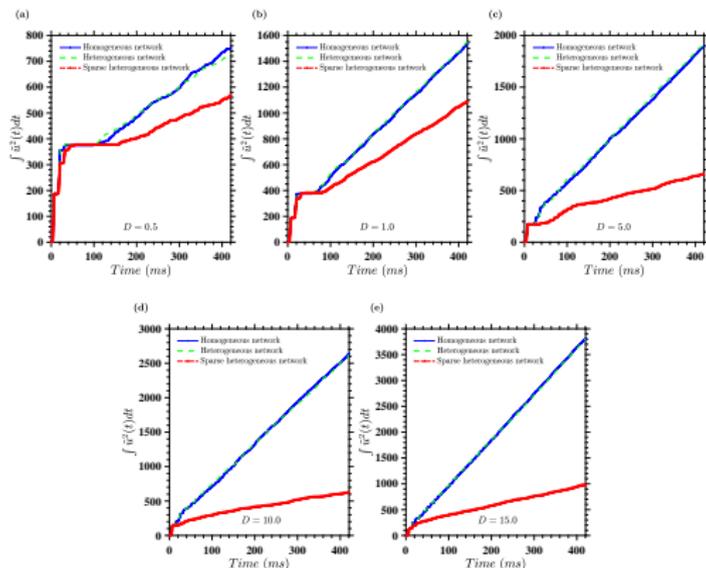
## Neural Network Behavior Under Varying Coupling Strength



**Figure:** Network behavior with active event-based control for  $D = 10$  and  $N = 100$  neurons. Panels show different coupling strength conditions: (1) uniform  $\alpha = 0.20$ , (2) random distribution, and (3) 20% zeroed coupling strengths. Dotted gray traces represent mean voltage, with control activation indicated by horizontal dotted lines.

# Robustness to Network Properties

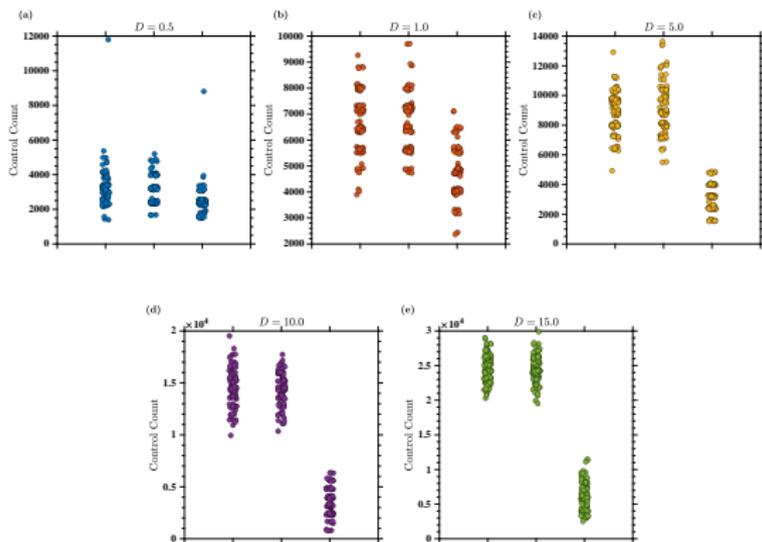
## Cumulative Energy Expenditure



**Figure:** Energy expenditure across coupling scenarios and noise levels. Plots show  $\int [u^*(t)]^2 dt$  averaged over 100 realizations for varying  $D$  and  $\bar{\alpha}$ . Blue: Homogeneous; green: Heterogeneous; red: Sparse Heterogeneous

# Robustness to Network Properties

## Robustness of Control Duration Across Coupling Scenarios



**Figure:** Control duration distribution across coupling scenarios and noise levels. Plot shows control time durations over 100 realizations for varying  $D$  and  $\bar{\alpha}$ . Blue: Homogeneous; green: Heterogeneous; red: Sparse Heterogeneous

# Robustness to Network Properties

## Control Duration Across Coupling Scenarios

Homogeneous			Heterogeneous		Sparse Heterogeneous	
$(D, \bar{\alpha})$	$\mathbb{E}[\int \tilde{u}^2(t)dt]$	Control Time (ms)	$\mathbb{E}[\int \tilde{u}^2(t)dt]$	Control Time (ms)	$\mathbb{E}[\int \tilde{u}^2(t)dt]$	Control Time (ms)
(0.5, 0.05)	752.97	28.35	729.17	27.54	546.85	20.58
(1.0, 0.10)	1540.44	58.00	1548.44	58.26	1048.92	39.52
(5.0, 0.15)	1895.18	77.64	1918.11	78.78	754.88	30.94
(10.0, 0.20)	2636.17	126.76	2619.58	125.64	646.08	31.00
(15.0, 0.25)	3815.53	213.98	3813.70	213.66	1012.22	56.52

**Table:** Energy consumption  $\mathbb{E}[\int \tilde{u}^2(t)dt]$  and control duration for  $N = 100$  coupled neurons with event-based control over 425 ms. Network types: Homogeneous, Heterogeneous, Sparse Heterogeneous. Averaged over 100 noise realizations.

# Conclusion

## Summary of Findings and Key Results

### **Summary of Findings:**

- Energy-optimal control for phase resetting of stochastic neural oscillators, generalizing earlier work on deterministic oscillators.
- Development of a nonlinear second-order monotone scheme solver for BSPDEs applied to the stochastic HJB equation using level set methods.
- Event-based feedback control effectively desynchronizes neural oscillators, activated when voltage exceeds a threshold.

### **Key Results:**

- Stochastic control outperforms deterministic methods in energy efficiency, particularly in high-noise environments.
- Achieved energy savings of up to 32% for higher noise levels ( $D = 15$ ).
- Demonstrated robustness to variations in neuronal coupling strengths, ensuring effectiveness in non-uniform neural networks.

# Conclusion

## Implications and Future Directions

### ***Implications and Broader Applications:***

- Potential to extend battery life of DBS devices for Parkinson's disease treatment.
- Improved energy efficiency and robustness in heterogeneous neural networks.
- Applicability to other fields, such as robotics, aerospace, and finance.
- Revisiting studies on seizure-like bursting and cardiac arrhythmias with noise integration.

### ***Future Research Directions:***

- Developing adaptive DBS protocols that account for stochastic neural dynamics.
- Investigating long-term effects on neural plasticity and network reorganization.
- Conducting comparative clinical studies to validate real-world benefits of stochastic control strategies.

# Ongoing and Future Work

## Proposed Research Directions

### **Proposed Research Directions:**

- Preparing an *in-proceedings* paper for the *Journal of Computer Physics Communications*, featuring a software manual for a 2D nonlinear PDE solver. This solver addresses a variety of deterministic and stochastic Hamilton-Jacobi equations and aims to serve the broader research community.
- Developing a machine learning-based, data-driven approach for adaptive Deep Brain Stimulation (DBS) to enhance therapeutic outcomes in neurological disorders.
- Expanding the software package to solve higher-dimensional Hamilton-Jacobi equations, up to 4D, with robust capabilities for both stochastic and deterministic cases.
- Modeling a 4D coupled neuron system to explore complex neural dynamics under stochastic influences and their control mechanisms.

## *Connecting the Dots*

# From Neural Oscillators to Protein Aggregation

### Previous Work: Stochastic PDEs

- Focus on nonlinear optimization and control
- Developed robust numerical methods
- Solved complex stochastic equations
- Fixed computational domains

### New Challenge: Free Boundary Problem

- Moving boundaries (protein aggregates)
- Level-set representation
- Multi-scale physics
- Complex geometries

## Building on Our Foundation

Our computational expertise in:

- Numerical methods for nonlinear PDEs
- High-performance computing
- Multi-scale modeling

provides the perfect foundation for tackling free boundary problems in protein aggregation.

# Acknowledgments



Dr. Frederic Gibou  
UCSB



Dr. Poura A. Mistani  
Synopsis

Based on work in

P. A. Mistani and F. Gibou. [Continuum dynamics protein aggregation model](#).

*insert NSF link or project URL here, if available*, 2024, Proposal submitted for NSF funding

# Motivation and Background

## Protein Aggregation in Biotherapeutics

### *Protein-based Therapeutics*

- Monoclonal antibodies (mAb) are a major class of biotherapeutics.
- Challenges arise at high concentrations:
  - Multi-body interactions and crowding effects
  - Increased viscosity and instability
  - Conformational and colloidal instability
- Stable high-concentration formulations are crucial for improving efficacy and patient outcomes.

### *Understanding Aggregation and Morphology*

- Protein aggregation accelerates at high concentrations, leading to:
  - Irreversible aggregation, limiting shelf life
  - Increased viscosity complicating formulation and drug delivery
  - Immune responses due to high molecular weight aggregates
- Understanding the aggregation-morphology relationship is key to predicting long-term stability.

# Protein Aggregation

## Computational Models

### **Why Computational Models?**

They are advantageous for the early development of biotherapeutics.

- Limited material availability
- Need to explore many formulation conditions

### **Main Challenge?**

- Long-term stability prediction arising in high concentration formulations
- Multi-scale nature of aggregation:
  - Spatial scales: monomers to large aggregates
  - Time scales: microseconds to months

# Protein Aggregation

## Existing Computational Models

### Kinetic Rate Models

Describes protein aggregation through population balance equations, tracking the evolution of mass fractions over time.

#### **Key Features:**

- Rate equations for:
  - Nucleation
  - Growth
  - Condensation
- Statistical averaging of bulk properties
- Efficient for long timescales

### Molecular Dynamics

Simulates protein aggregation by solving Newton's equations of motion for individual particles at atomic resolution.

#### **Key Features:**

- Force fields include:
  - Bonded interactions
  - Non-bonded forces
  - Solvent effects
- Detailed molecular trajectories
- Limited to short timescales

# Motivation and Background

## Limitations of Existing Computational Models

### Kinetic Models

#### Advantages:

- Long timescale coverage
- Large systems
- Computationally efficient

#### Limitations:

- Lack spatial resolution
- Miss morphological details
- No local interactions

### Our Approach

#### *Bridging the Two Models!*

#### Key Features:

- Multi-scale coupling
- Level-set representation
- Spatial resolution
- Morphological details
- Long-range physics
- Efficient computation

### Molecular Dynamics

#### Advantages:

- High spatial resolution
- Atomic precision
- Captures solvent effects

#### Limitations:

- Limited timescales
- High computational cost
- Small system sizes

# Protein Aggregation

## Protein Aggregation Process

The protein aggregation process has two main phases:

- 1 **Nucleation**
- 2 **Growth**
  - Monomer addition
  - Polymerization
  - Condensation

These processes are described by rate equations developed by Roberts et al (2009).

Y. Li and C. J. Roberts. Lumry-eyring nucleated-polymerization model of protein aggregation kinetics. 2. competing growth via condensation and chain polymerization.

*The Journal of Physical Chemistry B*, 113(19):7020–7032, 2009

# Protein Aggregation

## Kinetic Model for Protein Aggregation

The kinetic model describes the following three main components:

### 1 Monomer fraction ( $m$ )

$$m = \frac{[N] + [I] + [U]}{C_0}$$

- $C_0$ : Initial monomer concentration.
- $N$ : Native monomers.
- $I$ : Intermediate state monomers.
- $U$ : Unfolded state monomers.

### 2 Aggregate fraction of size $x$ ( $a_x$ ).

### 3 Aggregate fraction of larger sizes ( $a_i$ ):

- Where  $x < i < n^*$ .
- $n^*$ : Size at which precipitation occurs (insoluble aggregate).

# Protein Aggregation

## Kinetic Model

### 1. Monomer Depletion

$$\frac{dm}{d\theta} = -xm^x - \delta\beta_{gn}m^\delta\sigma$$

- $-xm^x$ : Nucleation phase.
- $-\delta\beta_{gn}m^\delta\sigma$ : Growth phase (monomer addition and polymerization).

### 2. Formation of Aggregates of Size $x$

$$\frac{da_x}{d\theta} = m^x - \beta_{gn}a_xm^\delta - \beta_{cg}\beta_{gn}a_x\left(\kappa_{x,x}a_x + \sum_{j=x}^{n^*-1} \kappa_{x,j}a_j\right)$$

- $m^x$ : Formation due to nucleation.
- $-\beta_{gn}a_xm^\delta$ : Growth phase (monomer addition).
- $-\beta_{cg}\beta_{gn}$ : Condensation mechanism.

### 3. Formation of Larger Aggregates ( $x < i < n^*$ )

$$\frac{da_i}{d\theta} = \beta_{gn}(a_{i-\delta} - a_i)m^\delta$$

 $n^* - 1$

# Protein Aggregation

Kinetic Model: Understanding  $\kappa_{i,j}$

$\kappa_{i,j}$  represents how easily/quickly aggregates combine

## **Collision Rates**

How often aggregates bump into each other:

- Transport mechanisms (diffusion)
- Inter-molecular interactions

## **Capture Rates**

Likelihood of sticking after collision:

- Collision cross section
- Binding affinities

# Protein Aggregation

Kinetic Model: Understanding  $\kappa_{i,j}$

## **Collision Forces**

Long-range:

- Electrostatic interactions
- Hydrodynamic interactions

Short-range:

- Van der Waals forces
- Brownian motion

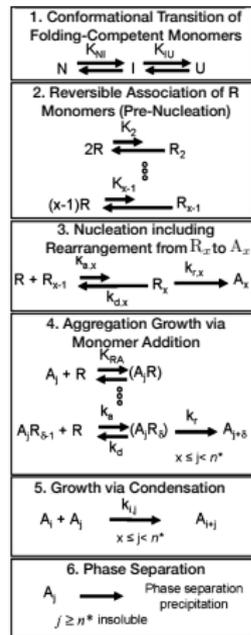
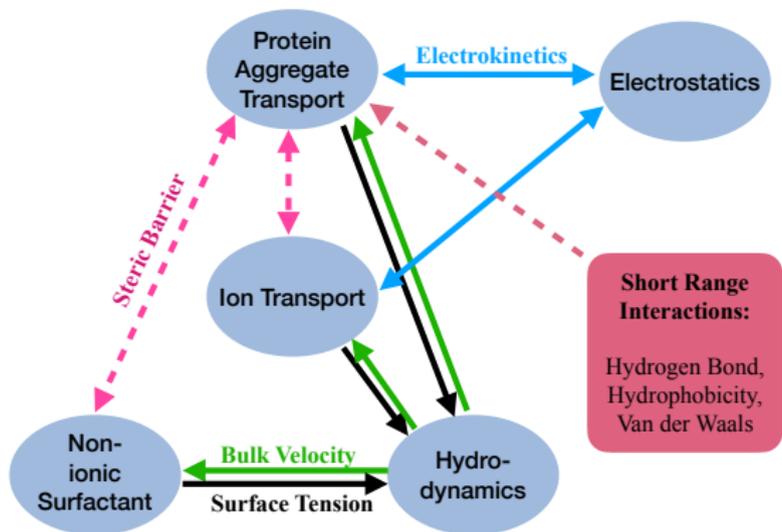
## **Capture Forces**

Short-range only:

- Hydrogen bonding
- Hydrophobic interactions
- Binding site interactions
- Van der Waals forces
- Short-range electrostatic

# Protein Aggregation

## Conceptual Schematic



**Figure:** (Left) Physical interactions during protein aggregation. (Right) The different stages of protein aggregation driven by all the physical interactions in the solution

# Continuum Model

The Uniform Density Model for Epitaxy

## *What is Epitaxial Growth?*

Epitaxial growth involves depositing adatoms (diffusing atoms) onto a substrate, where they diffuse, nucleate, and aggregate to form ordered atomic layers.

## *The Uniform Density Model:*

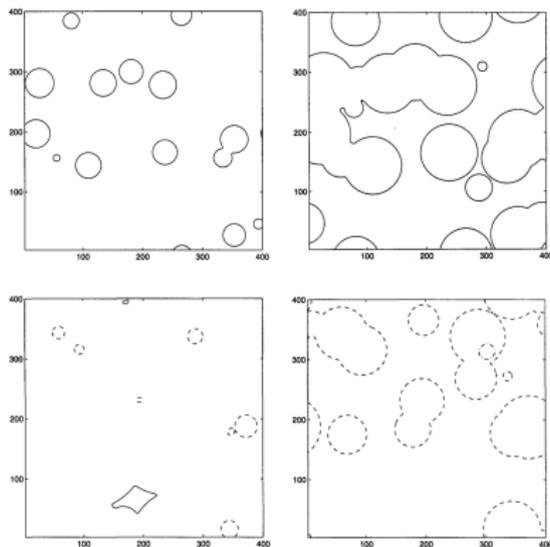
Caflich et al. introduced this model to simulate epitaxial growth, focusing on the interaction between nucleation, adatom density, and morphology evolution.

## *Key Concepts:*

- Islands form as circular regions and grow until coalescence begins.
- Island boundaries evolve based on adatom density  $p(t)$  and deposition coverage.
- Steady-state shapes depend on the attachment rate  $A(\theta)$ .

# Continuum Model

## The Uniform Density Model for Epitaxy



**Figure:** *Island morphology at various coverages: (a) 10%, (b) 50%, (c) 100%, and (d) 130%.*

R. E. Caflisch, M. F. Gyure, N. Papanicolaou, W. E. Prachtka, and D. D. Vvedensky. *Kinetic monte carlo simulation of epitaxial growth.*

*Physical Review E*, 59(6):6879–6887, 1999

# Continuum Model

Inspired by the Island Dynamics Model for Epitaxy

## ***Model Relevance:***

Our approach is inspired by Caflisch et al.'s Island Dynamics Model (IDM) for epitaxial growth Caflisch et al. (1999), which simulates island formation, growth, and coalescence based on adatom density.

## ***Key Insights from IDM:***

- Models island growth as a free boundary problem.
- Bridges microscopic atomic movement and macroscopic elastic effects.
- Influences our modeling of aggregation and morphology evolution.

# Continuum Model

## Model Overview

**Core Concept:** Adding spatial resolution to protein aggregation kinetic models via a continuum representation of species and to include the two-way coupling of long and short range interactions through the design of effective boundary conditions

### Level-set Representation $\phi(\mathbf{x}, t)$ :

- Tracks aggregate boundaries as a free surface problem
- Evolves with monomer flux at the surface

### Domains:

- $\Omega^-$ : Aggregates
- $\Omega^+$ : Outside aggregates
- $\Gamma_j$ : Aggregate surface

### Density Evolution:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D(\mathbf{x}) \nabla \rho) - x \frac{dn_x}{dt}$$

# Continuum Model

## Physical Processes

### 1. Nucleation:

- Stochastic or deterministic nucleation sites.
- Rate:

$$\frac{dn_x}{dt} = \left( \frac{\tau_g}{\tau_n} \right) \frac{\langle \rho^x(t) \rangle}{\langle \rho(0) \rangle^{x-1}}$$

### 2. Growth Mechanisms:

- Surface flux (monomer addition):

$$D \nabla \rho \cdot \mathbf{n} = -\delta \frac{\rho^\delta - \rho_{eq}^\delta}{\langle \rho(0) \rangle^{\delta-1}}$$

- Robin boundary condition with surface energy:

$$\nabla \rho \cdot \mathbf{n} + \alpha \rho^\delta = \alpha \rho_{eq}^\delta$$

### 3. Level-set Evolution:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = \sum_n \delta(t - t_n) \delta(\mathbf{x} - \mathbf{x}_n)$$

# Continuum Model

## Multi-Physics Integration

### 1. Hydrodynamics:

- Incompressible Stokes flow equations:

$$\nabla \cdot \mathbf{v} = 0$$

$$\mu \nabla^2 \mathbf{v} - \nabla P + \rho_l \mathbf{f} = 0$$

### 2. Electrostatics:

- Poisson-Boltzmann equation:

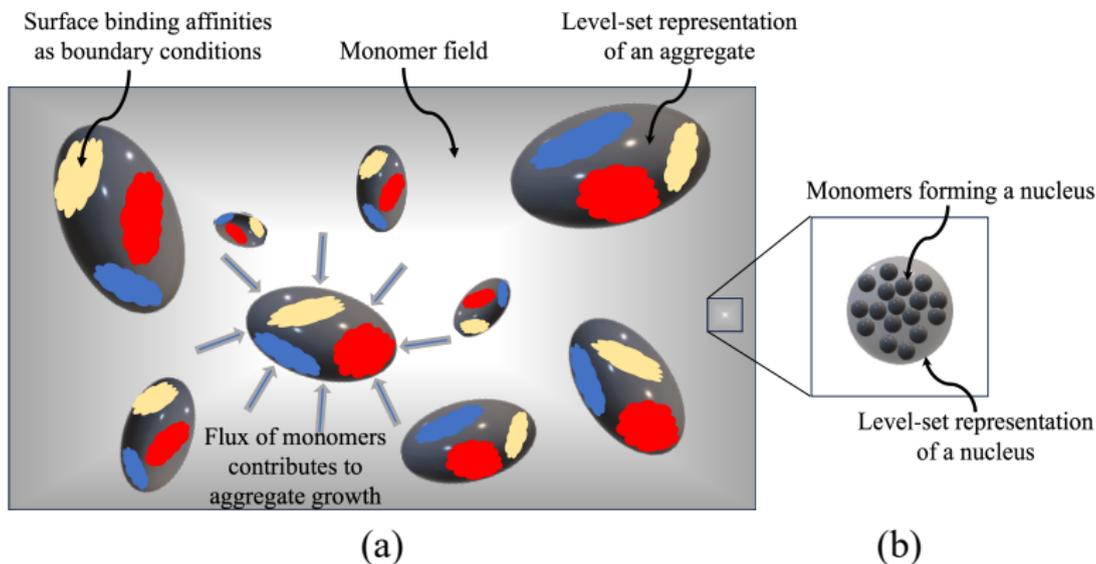
$$\nabla \cdot (\epsilon \nabla \psi) = -\kappa_m^2 e z \rho - \kappa^2 e (z_+ c_+ - z_- c_-)$$

### 3. Coupled Transport:

- Species conservation with electrostatics.
- Ion transport and surfactant effects on surface tension.

# Continuum Model

## Conceptual Schematic



**Figure:** *Aggregates are represented by a level-set function that evolves according to the flux of monomers to its interface. Nucleations generate new instances of the level-set function. Surface affinities and long-range interactions are readily included through boundary conditions and by solving partial differential equations in irregular domains under the continuum assumption.*

# Preliminary Results

## Poisson Equation Solver

Bochkov et al. (2019) solved *Poisson-type equations on irregular domains* with Robin boundary conditions using two finite volume schemes.

Consider the PDE on domain  $\Omega \subseteq \mathbb{R}^n$  ( $n = 2$  or  $3$ ) with  $\partial\Omega = \bigcup_{p=1}^N \Gamma_p$ , where  $\Gamma_p$  are smooth subdomains:

### Governing Equation

$$-\nabla \cdot (\mu(\mathbf{r})\nabla u(\mathbf{r})) + k(\mathbf{r})u(\mathbf{r}) = f(\mathbf{r})$$

- $\mathbf{r}$ : spatial coordinates
- $\mu(\mathbf{r})$ : diffusion coefficient
- $k(\mathbf{r})$ : reaction coefficient
- $f(\mathbf{r})$ : source term

D. Bochkov and F. Gibou. [Solving poisson-type equations with robin boundary conditions on piecewise smooth interfaces.](#)

*Journal of Computational Physics*, 376:1156–1198, 2019

# Preliminary Results

## Poisson Equation Solver

The boundary conditions on  $\partial\Omega = \bigcup_{p=1}^N \Gamma_p$  are given by:

### Boundary Condition

$$\mu(\mathbf{r}) \frac{\partial u}{\partial n_p} + \alpha_p(\mathbf{r}) u(\mathbf{r}) = g_p(\mathbf{r})$$

where:

- $p$  is the subdomain index,
- $N$  is the number of smooth  $C^2$  subdomains,
- $\alpha_p(\mathbf{r})$  is the boundary coefficient, and
- $g_p(\mathbf{r})$  is the boundary function.

# Preliminary Results

## Single Protein with Two Patches

We consider a compound domain that is the union of two disks with radii

$$r_2 = \frac{1}{2}r_1.$$

### *The test function:*

$$\begin{cases} u^+ = 2 \log((x + 0.8y)^2 + x - 0.7y + 4) - 3 \\ u^- = 0 \end{cases}$$

We set as  $\mu = 1$  and  $k = 0$ :

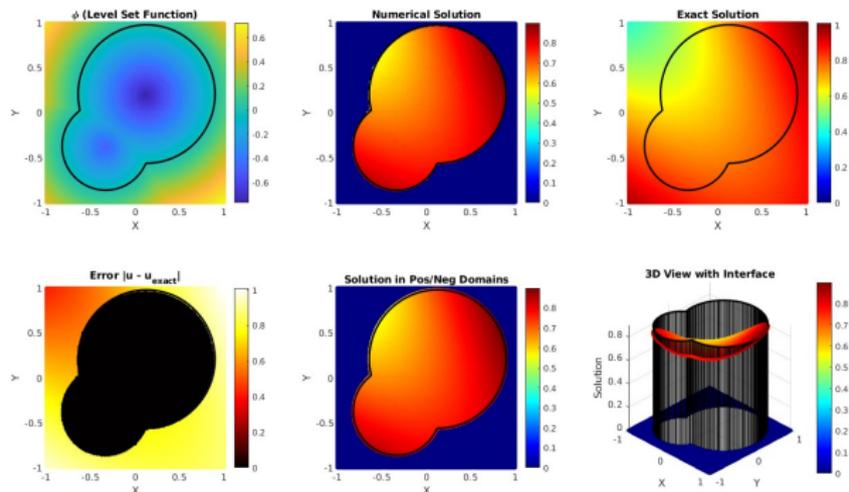
$$-\mu \nabla^2 u(r) = 0.$$

### *The Robin boundary conditions:*

$$\mu \frac{\partial u}{\partial n} + \alpha u = g, \quad \text{with} \quad \alpha = \begin{cases} \alpha_1 = 1, & \text{on } Disk_1, \\ \alpha_2 = \sin(x + y) \cos(x - y), & \text{on } Disk_2. \end{cases}$$

# Preliminary Results

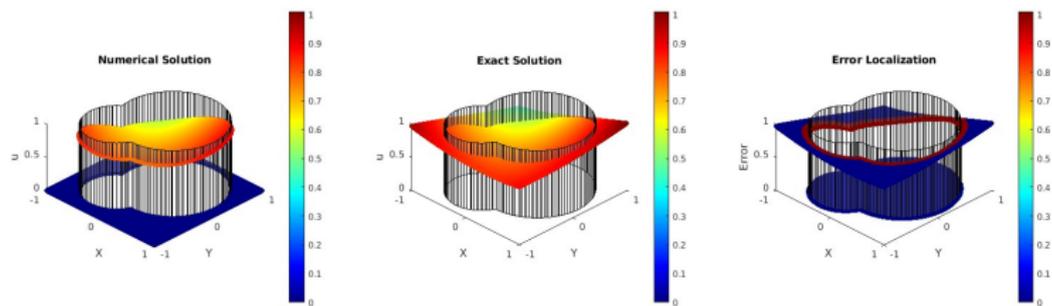
All Subplots



**Figure:** Visualization of the level-set function, numerical and exact solutions, and error contours for both positive and negative domains. The plot shows how the solution behaves across the entire domain, with clear separation of regions.

# Preliminary Results

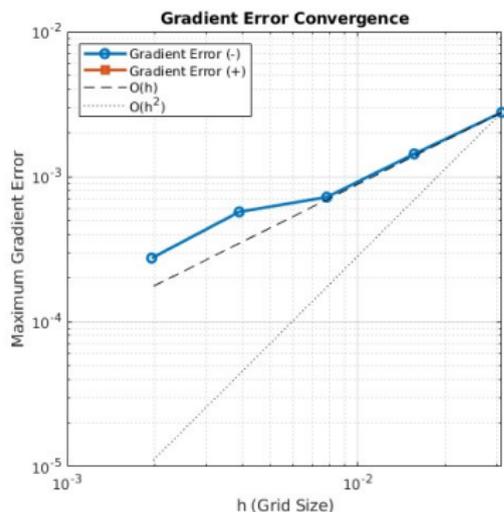
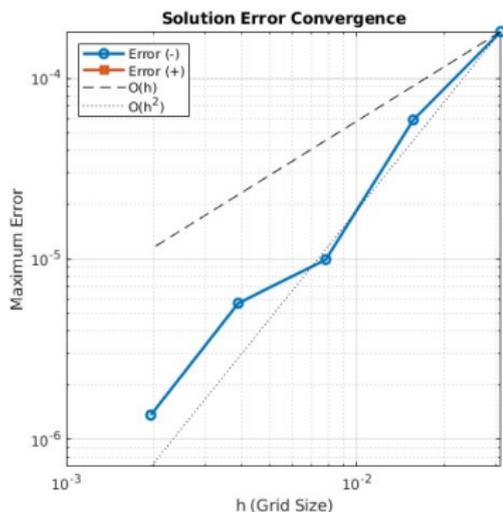
## Single Protein with Two Patches



**Figure:** A comparison of the numerical solution with the exact solution. The semi-transparent plane interface highlights areas where the numerical solution approximates the exact solution, particularly near critical regions like discontinuities or gradients.

# Preliminary Results

## Single Protein with Two Patches



**Figure:** Convergence analysis showing the accuracy of the numerical method as a function of grid resolution. The plot demonstrates how the error decreases, reflecting the reliability of the numerical solution as the grid is refined.

# Ongoing and Future Work

## Proposed Research Direction

### ***Design and implementation of computational model:***

- 1 Monomer and aggregate evolution modeling, including nucleation.
- 2 Represent surface affinities through boundary conditions.
- 3 Incorporate long-range interactions.

### ***Validation:***

- 1 Validate aggregation and cluster size distribution.
- 2 Validate short-time diffusivity predictions.
- 3 Validate viscosity measurements.

Thank for Your Attention

Questions? Comments?