

Simulating the Walk of the Dynein Motor Protein

Jin Kiatvongcharoen, John Waczak, Elliott Capek, and David Roundy
Department of Physics, Oregon State University, Corvallis, OR 97331

Dynein

Dynein is one of the three motor proteins that are responsible for the cell's ability to move, divide, and spatially organize itself. They walk with cellular cargo along protein highways (microtubules) in order to convert the cargo's ATP into mechanical work.

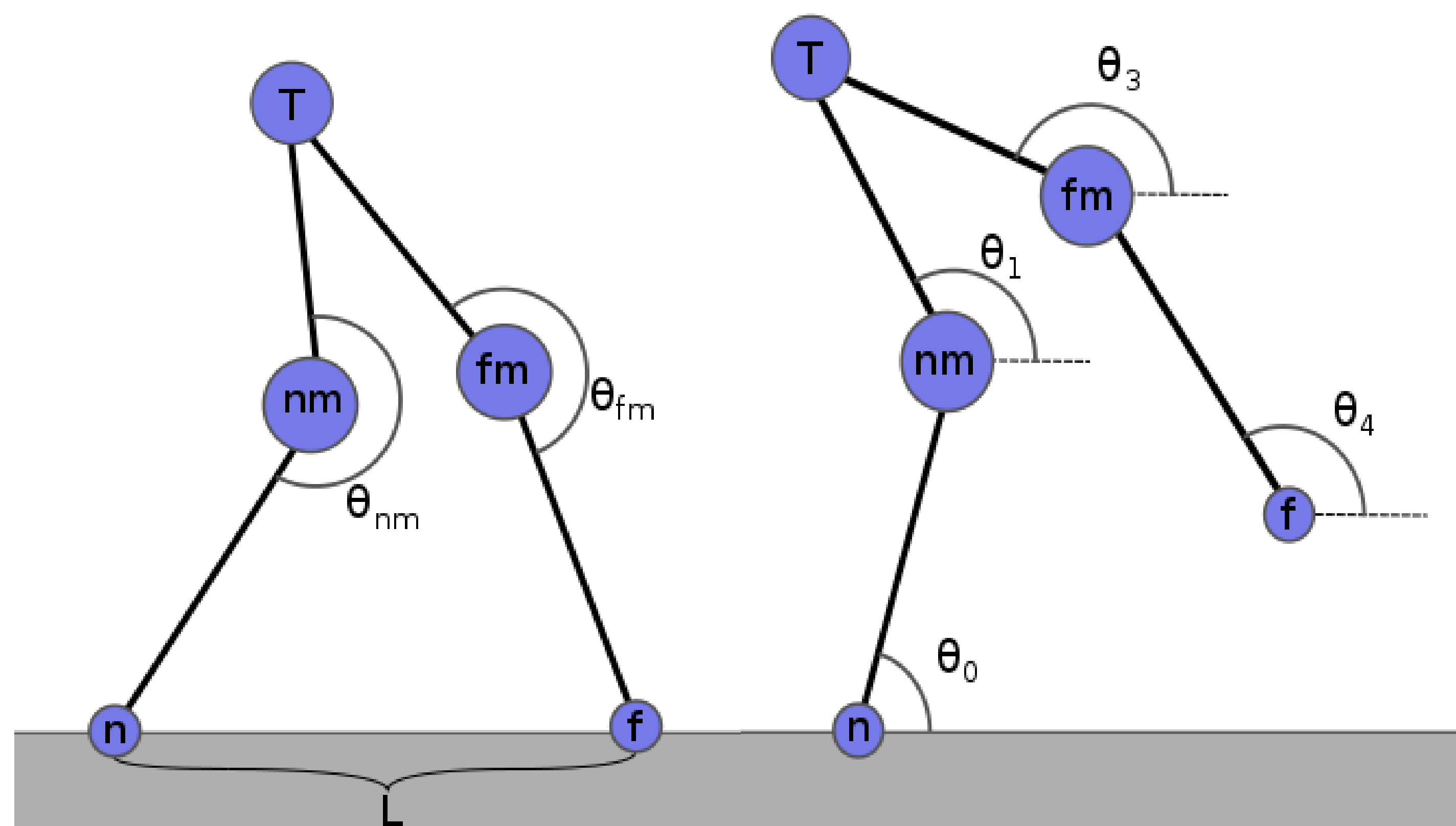
However, it walks drunk. Its two feet can act independently from each other causing erratic steps that vary in distances, directions and time.

Goal

Our goal is to explore the mechanical nature of dynein and simulate its walk with a simple 2-D model. Being able to predict its motion will be advantageous for medical research and beyond.

Model

We model dynein as a point-particle system with five spherical domains that are held together by massless rigid-rods. The domains are the tail domain, the two motor domains, and the two binding domains. We separate dynein into two states: both bound and one bound.



Simulation

Randomly pick motor angles and make configuration

Calculate total energy of the system

Determine probability of unbinding

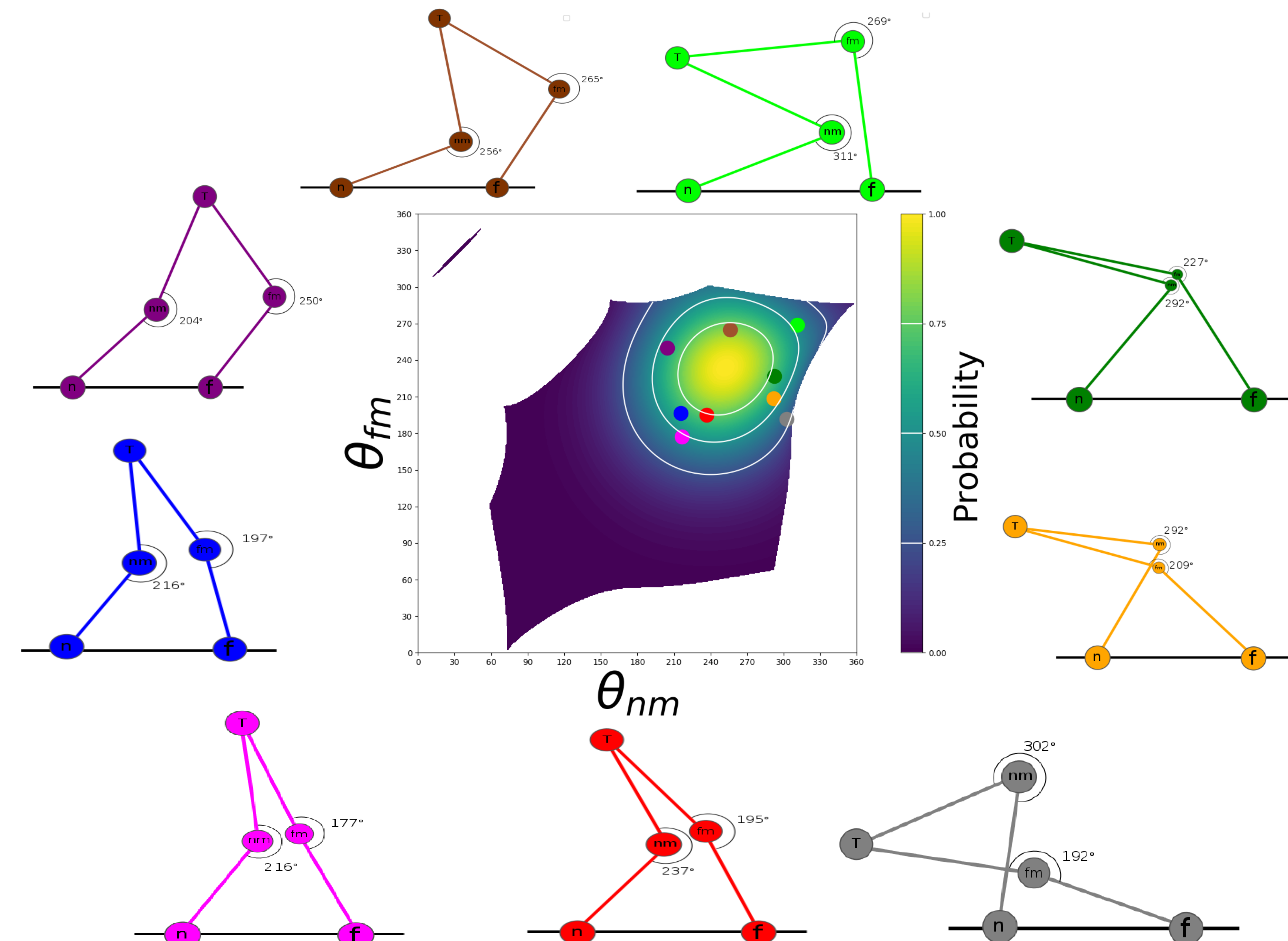
Start one bound simulation if unbinding is successful

Collect data and make statistics

Both Bound: Monte-Carlo (MC)

Monte-Carlo (MC) methods uses repeated random sampling of configurations to output a distribution of results depending on their probability of happening. We use a MC to repeatedly run the simulation process on the left and use the data to generate a distribution of the step sizes, step time, bound time, and unbinding probabilities.

Unbinding Probability Distribution



One Bound: Brownian

We simulate the one bound state using Brownian dynamics, in which the domains randomly interact with molecules to cause diffusion within the system. This leads to chaotic motion in the leg until the binding probability is high enough for it to land.

Future Work

- Improve and simplify one bound simulation.
- Analyze error in MC and compare it to older simulations.
- Compare results to experimental data and further refine model.