Towards a superparticle population density method for simulating systems of coupled oscillators

Andy Chen*

August 31, 2023

Abstract

We study the problem of simulating coupled oscillators in stable dynamical systems. For example, in Hodgkin-Huxley type models, the voltage and ion activations of neurons can be modeled as oscillating units in four or five-dimensional phase space evolved according to a differential equation. One challenge arises in simulating large populations of units with coupling (neurons tend to induce each other to fire) and noise. The asymmetric particle population density (APPD) method considers the limiting case of an infinite number of oscillators, using multivariate Gaussian particles to approximate a probability distribution in phase space. We detail novel clustering behaviors in a five-dimensional Hodgkin-Huxley model under APPD. We also describe a simplification in the representation of particles on the limit cycle.

1 Method Overview

The asymmetric particle population density method was originally introduced by Wang & Forger (2021) as a way to effectively simulate the evolution of a continuous population density of oscillators in phase space with noise and coupling.

One traditional way to model the behavior of oscillators such as neurons is to use direct Monte Carlo simulation. In the original Hodgkin-Huxley model, for example, a large number of distinct neurons would each be represented as points in four-dimensional phase space (with its state determined by voltage and three ion channel activations). At each time interval, each individual neuron is stepped forward according to the dynamics of the Hodgkin-Huxley equations, as well as contributions from random noise and global averaged coupling (accounting for the fact that neurons induce each other to fire).

Being a direct simulation, the performance of Monte Carlo simulations slows down as the number of oscillators increases. The APPD method aims to bypass this limitation by considering the limiting case of an infinite number of oscillators, continuously distributed in phase space as a population density. As a particle method, the population density is represented as the sum of separate Gaussian particles, which are asymmetric in the sense that they may be stretched, usually in the direction of the limit cycle. The original APPD paper describes how the effects of coupling and noise can be reformulated for this continuous scenario. Noise, for example, is accounted for via a per-timestep diffusion on every particle, which corresponds to adding a constant to the diagonal entries of its covariance square root.

The original APPD method describes a linearity condition under which a particle must be split into three child particles, with a good approximation to overall density distribution of the original particle. This usually occurs when a particle (representing a chunk of the total population of oscillators) moves into a section of the limit cycle where the dynamics become highly nonlinear. For example, this may happen at a section which is very curved or where the velocity given by the equation dynamics is very slow (ex. the resting phase of the Hodgkin-Huxley model). However, particles cannot be split ad infinitum, which results in a method to combine particles which are sufficiently close together. The current paper documents several inefficiencies which arise from the interaction of particle splits

*Mentor: Danny Forger

and combines, and outlines a new method which aims to obviate them. Additionally, we observe the macroscopic behaviors of a five-dimensional cortical Hodgkin-Huxley type model on APPD, and describe how they differ from those of the classical four-dimensional model.

2 Visualization and condition checks

A code was written to visualize the location and frequencies of several phenomena in the original APPD model (with 4D or 5D Hodgkin-Huxley models); the most important results from simulations visualized using it are described below. For example, we found that it was useful to be able to pinpoint the regions of the limit cycle at which the majority of particle splitting occurs.

We worked with the initial assumption that the satisfaction of the alignment condition described here would allow a simplification in the method (see superparticles below). Two key conditions that the APPD model relies on are weak overall coupling (that is, there is just enough coupling strength to nudge most of the population to cluster on the limit cycle over a long timescale) and that the system being simulated is an autonomous ODE (time-independent; this implies that all particles will eventually pass through a chosen point on the limit cycle). Our primary goal is to verify these conditions and correlate them to the "Gaussian-ness" of the weight of the population distribution as it repeatedly passes through a chosen fixed point on the limit cycle.

2.1 Local linearity condition

Let $\mathbf{x}$ be the center point of a particle $\mathbf{p} \coloneqq (w, \mathbf{x}, \Sigma)$, with its velocity (not counting diffusion/noise) evolved as $\mathbf{v}(\mathbf{x}) = \mathbf{v}_d(\mathbf{x}) + \mathbf{v}_c(u, \mathbf{x})$, corresponding to the dynamic (ODE) and coupling velocities; the latter also depends on the coupling strength contribution of the entire population u. Also let $\mathbf{a}_\mathbf{x}$ be the location, relative to $\mathbf{x}$, of an endpoint of the major axis of the one-standard deviation (how do we know σ?) level curve of the Gaussian corresponding to Σ. We are interested in finding the conditions/regions in which the velocity of a particle aligns closely with its ellipsoid's major axis. This is satisfied when

$$\frac{|\langle \mathbf{v}(\mathbf{x}), \mathbf{a}_\mathbf{x} \rangle|}{\|\mathbf{v}(\mathbf{x})\|^2 \|\mathbf{a}_\mathbf{x}\|^2} \approx 1.$$

An approximation to this quantity (coupling velocity excluded) for all particles was calculated in the Python visualization program, indicated by a viridis coloring scheme (purple corresponds to least linear and yellow to quantities close to 1). It was found that the 1-approximation held only for certain regions of the limit cycle, particularly those with high average velocity. For the Hodgkin-Huxley model, this corresponds to the firing or spike region. (Later we saw that this is also true for the van der Pol oscillator.)

Interestingly, when this same linearity test is applied to the 5-dimensional cortical model, the coloring of the limit cycle appears quite different, as can be seen in later snapshots of the two-clustering simulation.

2.2 Density projection in two dimensions

When projecting Gaussian particles onto a 2D subspace of $\mathbb{R}^4$ for the Hodgkin-Huxley model, the covariance matrix of the projected 2D Gaussian is given by $v^T \Sigma v$, where v is the 4-by-2 corresponding to 2 orthogonal unit vectors spanning the subspace, and Σ the covariance matrix of the original Gaussian.

In the simpler case of projecting onto a subspace spanned by 2 of the axes (i.e., plotting 2 out of 4 variables), one may simply extract the 4 entries embedded in the 4-by-4 (or 5-by-5) covariance matrix corresponding to the covariances between the variables of interest, and use those entries for the corresponding 2-by-2 projected Gaussian particle's covariance matrix. The information discarded in the form of the unused entries is not needed, as those entries correspond to the stretching of a given particle in precisely the dimensions which a two-dimensional projection compresses together. This is the default method by which we have produced the density heatmap or distribution parts of simulations (including with the later five-dimensional cortical model), with appropriate axis labels.

[Image of the projection corresponding to the image of dot product above]

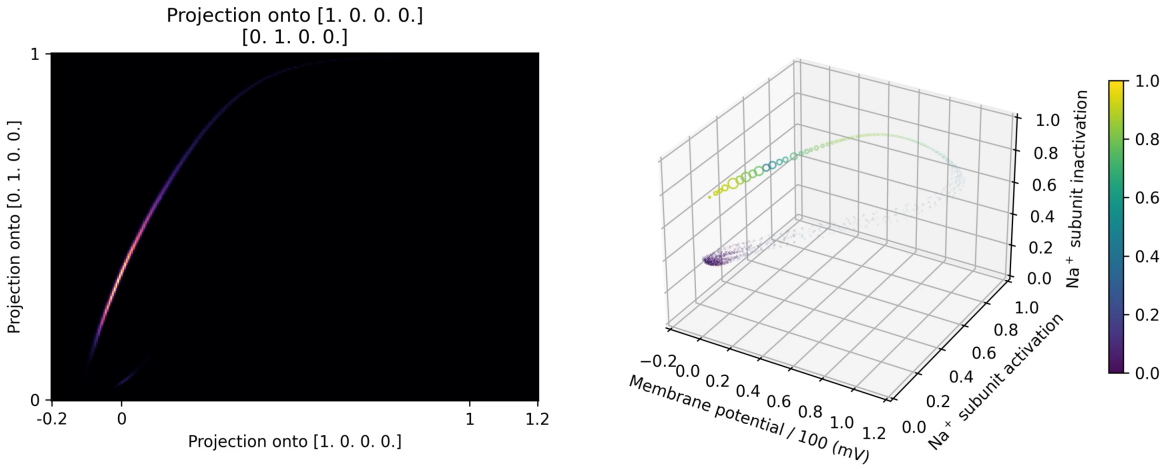

Figure 1: Density and particle plots for a 4D simulation with $I_{app} = 10.0$ V. Note the Gaussian shape of particles on the yellow section of the limit cycle, with the linearity condition largely satisfied.

2.3 Gaussian distribution of Gaussian particles

A further check for Gaussian flowed particles arises from choosing a fixed point or plane on the limit cycle and monitoring the proportion of the population passing through that plane in a given direction over time. In the 4-dimensional Hodgkin-Huxley model under default parameters, we tracked the sum of all particles passing through the 80 millivolt threshold with decreasing voltage over several cycles, which corresponds to the repolarization phase of firing. Because particles in the Hodgkin-Huxley model quickly attract to a limit cycle, the aforementioned criterion is in practice equivalent to monitoring the amount of weight at one point of the limit cycle over time. Note that the test here is not the same as a series of particles on the limit cycle appearing to follow a Gaussian distribution. The fact that we are monitoring the amount of particle weight passing through a point over time, however, means that the two will look similar provided that the velocity of that part of the population during firing remains about the same regardless of the amount of instantaneous mass there. We expect this to be true under the weak coupling assumption.

2.4 Excess split visualization

The locations of particle splits were also visualized in order to check where oversplitting usually occurs. One key idea is that the original C++ implementation of APPD can recursively split particles to a extraordinarily large degree at regions where the dynamics of the given ODE are highly nonlinear. For the classical Hodgkin-Huxley model, this corresponds to the resting phase region, where the above visualizations indicate that the alignment between major axes and velocity is nearly zero. In the two figures below, the number of splits resulting from a single particle is indicated by the size and color of a circle, logarithnically scaled: yellow corresponds to a split number on the order of 10^2. The observation that a large of particles are recursively generated from splitting and subsequently recombined in the same timestep provides motivations for improving the particle method in a way that avoids the need for splitting.

3 Cortical model behaviors

In this section we study the behaviors of a five-dimensional cortical neuron model simulated using APPD. The parameters for the model we used are given by the minimal Hogdkin-Huxley type equations for a cortical neuron in Popischil et al. (2008). The constant values are as follows: $I_{app} = -1.0, V_t = -61.5, G_{\mathrm{Na}} = 50.0, E_{\mathrm{Na}} = 50.0, G_{\mathrm{K}} = 5.0, E_{\mathrm{K}} = -90.0, G_{\mathrm{L}} = 0.003, E_{\mathrm{L}} = -70.0, G_{\mathrm{m}} = 0.13, \tau_{\max} = 1123.0$.

We are interested in simulating the first five dimensions: membrane potential, Na^+ activation, a

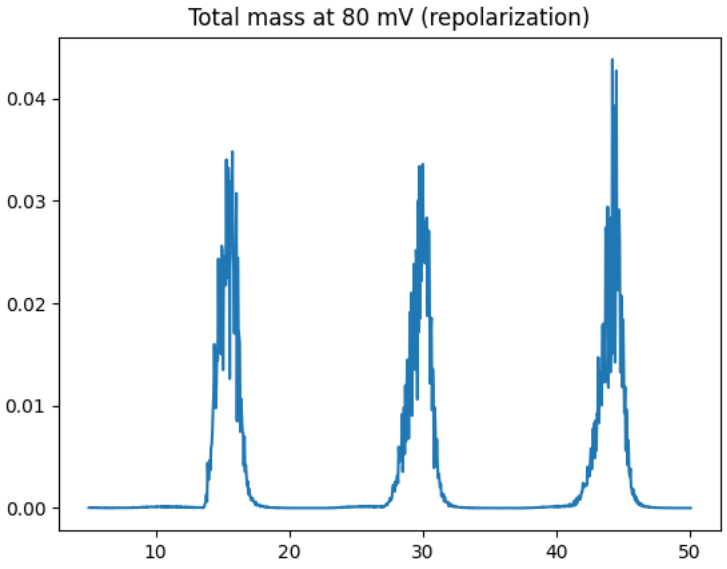

Figure 2: Gaussian distribution of particle weight flowing downwards through the 80 mV threshold.

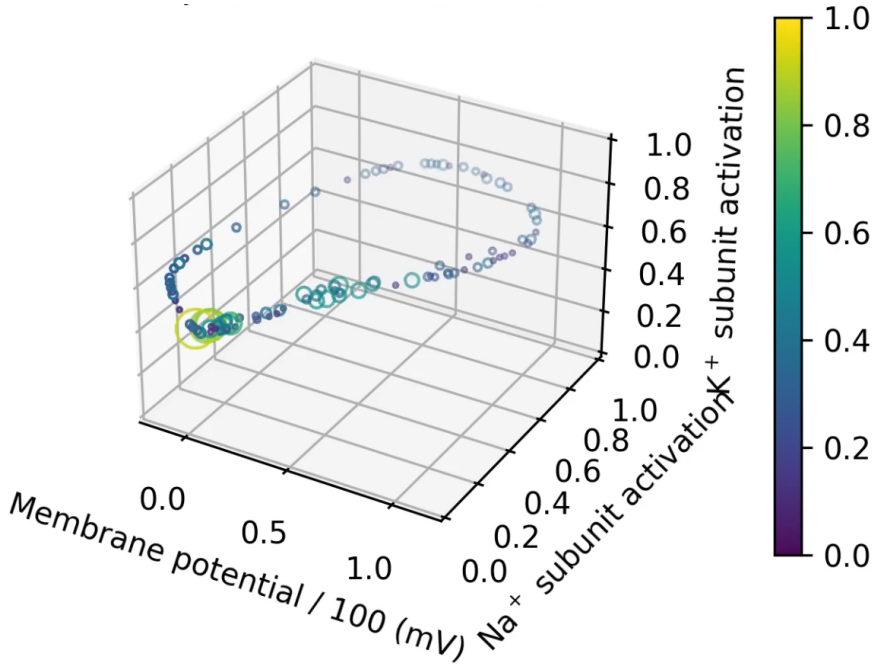

Figure 3: Visualization of split frequencies at firing under a default particle combine box radius τ.

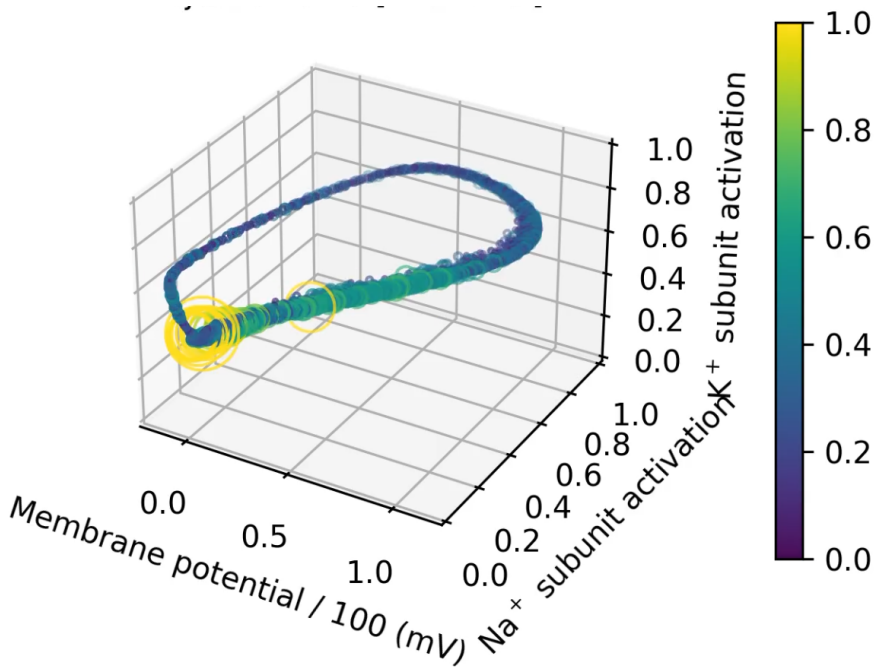

Figure 4: Visualization of split frequencies at firing when the particle combine box radius τ is decreased by a factor of 10.

"delayed-rectifier" K^+ activation, Na^+ inactivation, and a slow non-inactivating K^+ activation. This excludes Ca^{2+} activation, which can give rise to "bursting" behavior, in which a neuron fires rapidly multiple times before a longer resting period.

3.1 Computational trends

On average, the number of particles needed for the five-dimensional model, stablizing on the order of 10^3 is significantly higher than the number in the original, four-dimensional model, which has around a few hundred for default parameters. However, these estimates vary widely depending on chosen values for combine box width and noise levels.

3.2 Qualitative behaviors

An instance of the effects of weak coupling were seen when the cortical model was run under an applied current of -1.0 V, diffusion coefficient 10^{-7}, and coupling coefficient 0.05, resulting in an initially chaotic regime (characterized by the average voltage not showing any discernable periodicity) which transitioned to a stable firing pattern in the long term.

What appears to be quite different behavior emerges when the same model was run, but with an applied current magnitude increased to -5.0 V. In particular, a striking two-clustering regime arose over the course of the first 4000 or so timesteps.

Note that the frequency of voltage peaks (which normally corresponds to the vast majority of the population firing) eventually ends up doubling from its original around timestep 1600, and that this is preceded by a small secondary peak appearing after the primary peak at each firing, which eventually grows in amplitude until the two are roughly equal in strength. A video of the same simulation on the above visualization program reveals that the small secondary peak actually corresponds to backwards flow of a portion of the population: while a large percentage ends up fully firing and going around the limit cycle, the rest moves backwards to the resting phase without ever fully activating its Na^+ subunit dimension to close to 1.0.

One explanation which can be put forth for this initial "backflowing" behavior is the effect of mean-field coupling, which is the method of coupling calculation used in both the original simulations and the cortical model. The multiplier of coupling strength is directly scaled to the proportion of the

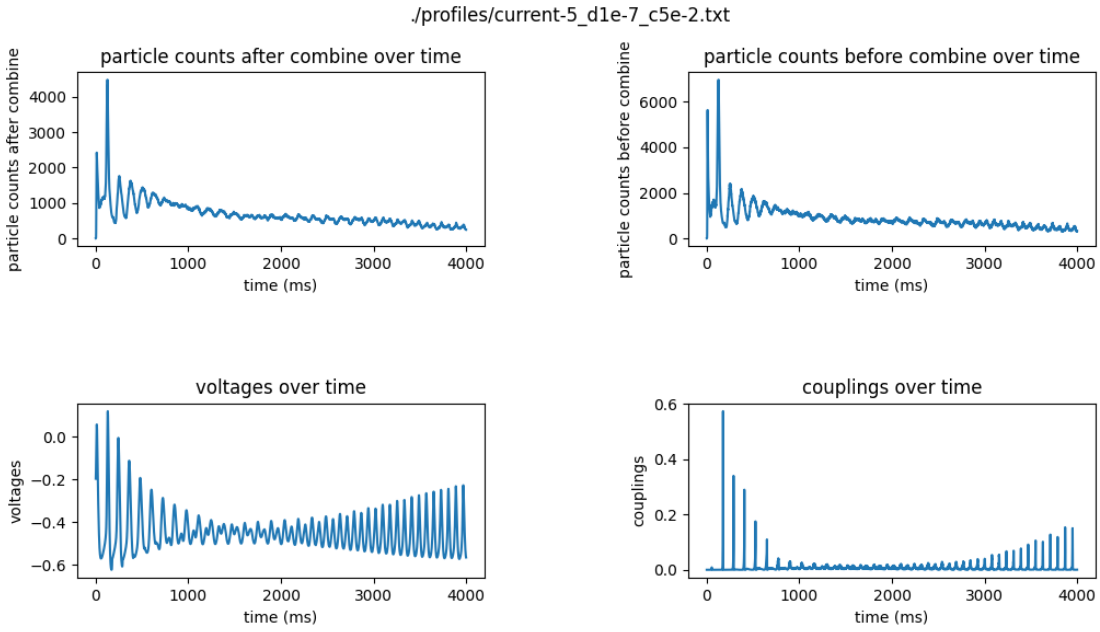

Figure 5: Plots for particle counts, voltage and coupling for two-clustering in a 5D cortical model

population which crosses over the threshold voltage at a given timestep, but the direction of coupling velocity is determined by how distant a particle is from that threshold.

Eventually, though, the prominence of the reverse flow (which should correspond to the secondary peaks) increases to the point that it nearly cancels out the effect of the normal firing which moves in the opposite direction. The effect on the average voltage plot above can been in that its overall magnitude decreases significantly. A new clustered regime then emerges, with the visualization clearly showing two distinct bunches of particles firing around the limit cycle sequentially, with the overall effect of a doubled firing frequency on the average voltage plot.

The asymptotic two-cluster state under the above parameters (and, significantly, with both excitatory mean-field coupling and noise) appears to be novel. Li et al. (2003) describes a two-cluster state, but for a small network of four integrate-and-fire neurons under excitatory coupling. Additionally, Vreeswijk et al. (1994) describes both synchronous and anti-synchronous states, but for a system of two individual Hodgkin-Huxley neurons with excitatory exponential coupling. It may also be fruitful to compare the asymptotic behavior with the results of inhibitory as opposed to mean-field coupling on a simpler model such as the van der Pol oscillator, which can also sometimes result in separation of the population into distinct clusters.

3.3 Python rewrite

The original C++ code from Wang & Forger (2021) relies on the use of MATLAB in order to output particle data over the timescale of a given simulation. The output was then fed into the visualization program to produce videos of the density projection, individual particle weights and locations, and split locations. However, the use of the original program was restricted to Windows machines due to its utilization of x86 architecture-specific parallelization packages.

In order to test out the feasibility of a code which could be more widely ported and simulated on different platforms and devices, a new implementation of the APPD method was written in Python. It is based on the original C++ implementation, and makes several optimizations and changes to accomodate the features of ODE integration packages particular to Python. For example, integration in scipy defaults to a Runge-Kutta method of order 5(4), whereas the C++ code uses a custom Bogacki-Shampine method of order 3(2); more testing is needed to determine which method results in greater local inaccuracy of the Hodgkin-Huxley equations, particularly at the part of the limit cycle corresponding to the resting phase, where overall velocities are slow and the majority of particle splits

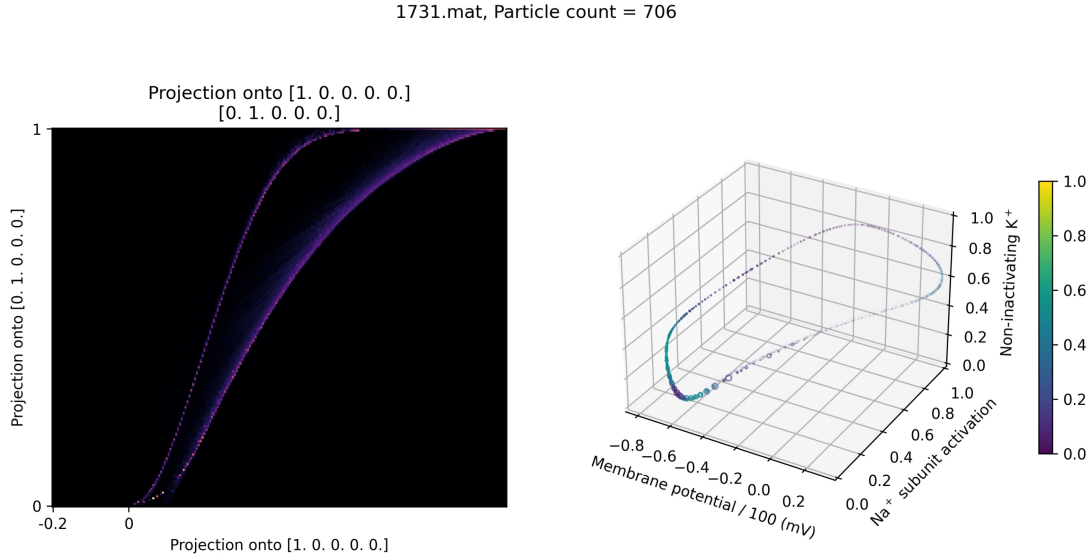

Figure 6: Initial reversed flow of particles under a 5-dimensional cortical model at timestep 1731.

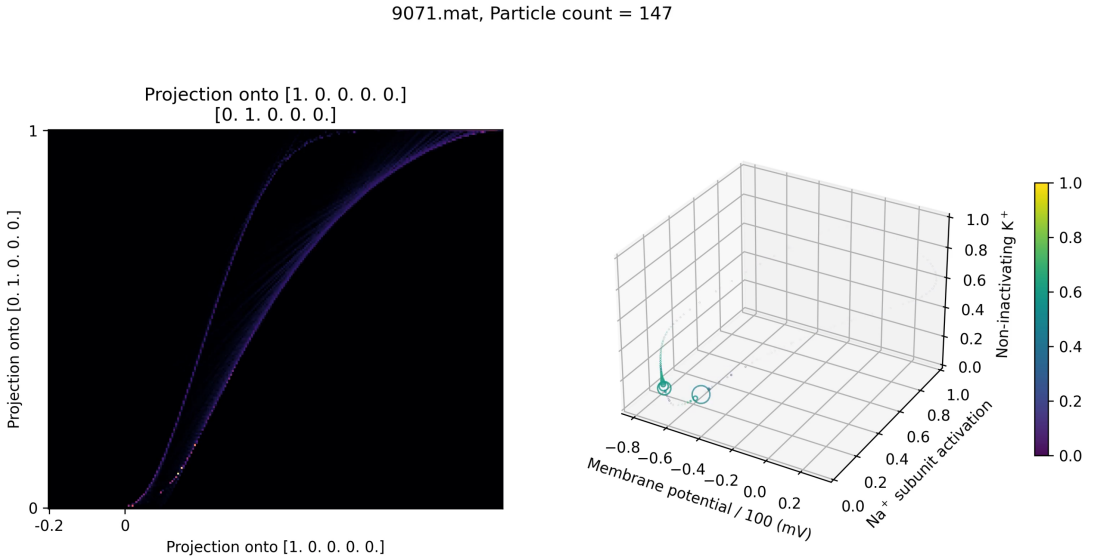

Figure 7: Asymptotic two-clustering behavior under a 5-dimensional cortical model at timestep 9701.

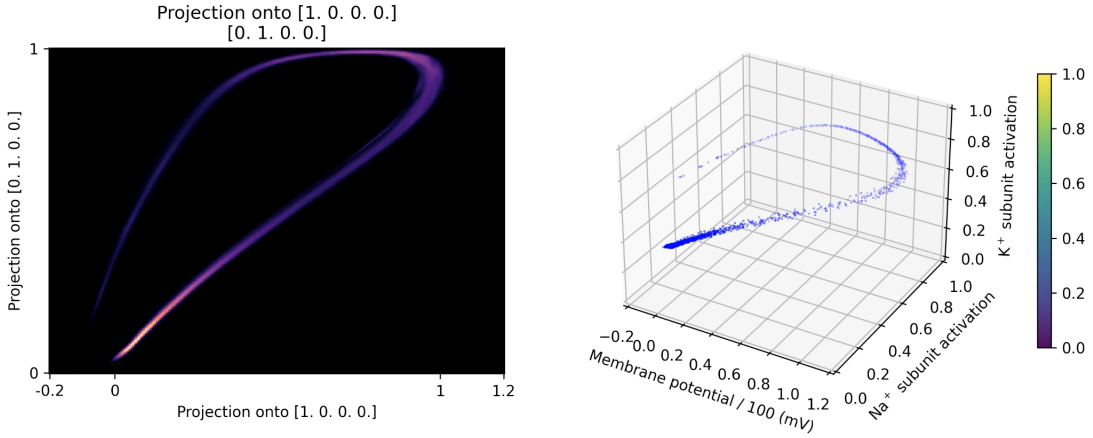

Figure 8: Instance of the 4D Hodgkin-Huxley model under a Python APPD implementation; the linearity calculation for particles was excluded.

occur. Greater precision at this non-linear region may reduce the need for unnecessary particle splitting. The Python code also does not use parallelization. The visualization code from the previous section was incorporated into this program to generate real-time snapshots of the state of a given simulation.