Quantifying the Complexity of Cardiac System Simulation by analyzing APD sequence

Xiaodong An

September 2024

1 Introduction

The study of chaotic dynamics in cardiac systems has gained significant importance due to its role in predicting irregular heart conditions like arrhythmia and fibrillation. This research focuses on quantifying chaos in cardiac simulations using Action Potential Duration (APD) data. By employing simplified cardiac models (Section 2.2), and techniques like Wolf's Algorithm (Section 2.5) and the Spatial-Temporal Algorithm (Section 2.6), this study aims to calculate the chaotic and non-chaotic regions for different parameters that can guide the cardiac medication to avoid deadly chaos. Moreover, I am trying to provide a more computationally efficient APD-based approach to chaos quantification in the cardiac system. Besides the simulations, in the future, with experimental data, this research might be used to improve the diagnosis of cardiac disease by efficiently quantifying the heart's complex behavior.

2 Literature Review

2.1 Action Potential Duration (APD) and Diastolic Interval (DI)

APD refers to the time from the start of depolarization to the end of repolarization, where people usually define a specific value as the starting and ending points. For example, if membrane potential is scaled to be $[0, 1]$, then APD_{70} or APD at 30% depolarization is the time between $u = 0.3$, as shown in Fig. 1.

DI means the time between the end of one APD and the start of the next when the cell tries to recover its excitability for the following depolarization, shown in Fig. 1 as well.

2.2 Three Variables Simplified Cardiac Model (3V SIM model)

3V SIM model or Fenton-Karma Model was developed in 1990s and it quantitatively reproduced APD vs DI curve (restitution curve) which determines the APD and relevant propagation velocity after repolarization [1]. It aims to provide a simplified solution to simulate the electrical activity of the cardiac system. Moreover, it balances the computational cost and physiological accuracy, making it quite useful in large-scale simulations with studies of arrhythmic behaviors in the cardiac system.

Specifically, 3V SIM model has three trans-membrane currents: fast inward (I_{fi} or Na^+), slow inward (I_{si} or Ca^{2+}), and slow outward current (I_{so} or K^+). Now, even with the names Na^+, Ca^{2+}, and K^+, it does not mean they represent those measured currents but only their activation and inactivation characteristics [1]. What's more, three state variables are defined as membrane voltage (u), and gate variables (v, w) where the latter two control the inactivation and reactivation of currents (similar models but with four state variables are defined in [2],[3]). During the whole activation-inactivation process as shown in Fig. 1, I_{fi} is responsible for the depolarization (upstroke part (1)) of membrane voltage, which is controlled by gate v, I_{so} is responsible for the repolarization of membrane voltage (part (3)) and I_{si} is responsible for maintaining the plateau (part (2)) after depolarization, which is controlled by gate w.

By tuning its parameters, there are four different parameter sets where this model could replicate restitution curves from original Beeler-Reuter (BR) model [4], modified forms of the Beeler-Reuter (MBR) model [4], Luo-Rudy-I (MLR-I) model [5], and also optical mapping of the epicardium (outermost heart layer) in the left ventricle of a pig [1]. Besides, it was also used in paper [6] to investigate the mechanisms for different spiral wave breakups.

Finally, the equations are defined as below [6]:

$$\partial_t u(\boldsymbol{x},t) = D\nabla^2 u - (I_{\text{fi}}(u,v) + I_{\text{so}}(u) + I_{\text{si}}(V,w))/C_m \tag{1}$$

$$\partial_t v(\boldsymbol{x},t) = (1-p)(1-v)/\tau_v^-(u) - pv/\tau_v^+(u) \tag{2}$$

$$\partial_t w(\boldsymbol{x},t) = (1-p)(1-w)/\tau_w^-(u) - pw/\tau_w^+(u) \tag{3}$$

$$I_{\text{fi}}(u,v) = -vp(u-u_c)(1-u)/\tau_d \tag{4}$$

$$I_{\text{so}}(u) = u(1-p)/\tau_0 + p/\tau_r \tag{5}$$

$$I_{\text{si}}(u,w) = -w(1+tanh(k(u-u_c^{\text{si}})))/(2\tau_{\text{si}}) \tag{6}$$

where:

$$p = \mathcal{H}(u-u_c) \tag{7}$$

$$q = \mathcal{H}(u-u_v) \tag{8}$$

and $\mathcal{H}()$ is Heaviside step function

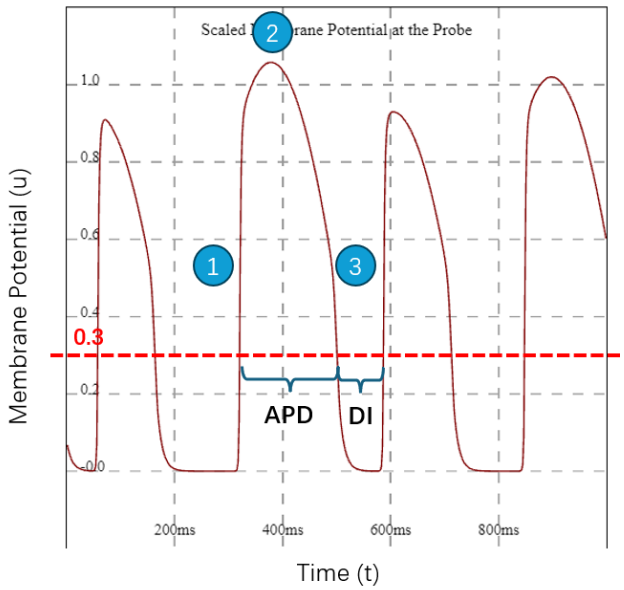

Figure 1: Membrane potential by 3V SIM Model

2.3 Chaos Quantification

Various methods, such as the Lyapunov exponent [7], correlation dimension [8], and return maps, are utilized to quantify chaotic behavior. In the actual simulation or experiment, several constraints may arise during complexity calculation involving inadequate temporal data length, limited observation of state variables (e.g., when only a single state variable could be observed), and the influence of the stochastic process (noise). Particularly, usual Lyapunov exponent methods require long temporal data length to get stable and reliable results [9]; partial state variables require reconstruction of the original phase space of the dynamical system [10]; the stochastic process needs to be distinguished from chaos when both have similar Lyapunov exponents [11],[12]. In the sections below, the calculation methods and solutions to the limitations mentioned will be reviewed.

2.4 Lyapunov Exponent

Lyapunov exponent is a quantitative measure of the divergence rate for nearby trajectories, implying the stability of a nonlinear system, spatially or temporally. Typically, following that basic definition, in a 1D system, the Lyapunov exponent can be naively calculated as [7]:

$$\lambda\big(x(0)\big) = \lim_{t\to\infty}\lim_{\delta x(0)\to 0}\frac{1}{t}\ln\frac{\delta x(t)}{\delta x(0)}. \tag{9}$$

where $\delta x(0)$ is the initial separation of two trajectories.

Now, heading to the higher dimensional system, it becomes [13]:

$$\lambda\big(\boldsymbol{X}(0)\big) = \lim_{t\to\infty}\frac{1}{t}\ln\frac{\|J^t\big(\boldsymbol{X}(0)\big)\delta\boldsymbol{X}(0)\|}{\|\delta\boldsymbol{X}(0)\|} = \lim_{t\to\infty}\frac{1}{2t}\ln\left(\hat{n}^\top J^{t\top}J^t\hat{n}\right). \tag{10}$$

where:

- $\hat{n} = \frac{\delta\boldsymbol{X}(0)}{\|\delta\boldsymbol{X}(0)\|}$ is the direction vector.
- $J^t\big(\boldsymbol{X}(0)\big) = \Pi_{t'=1}^{t'=t}J^{t'}\big(\boldsymbol{X}(0)\big)$ is the Jacobian matrix.
- $J^t_{ij}\big(\boldsymbol{X}(0)\big) = \frac{\partial\boldsymbol{X}_i(t)}{\partial\boldsymbol{X}_j(0)}$ is the element of the Jacobian matrix.

However, in actual calculation, we cannot have infinite time series, so the only one we can get is finite-time Lyapunov exponent, and it is defined as [13]:

$$\lambda\big(\boldsymbol{X}(0),t\big) = \frac{1}{2t}\ln\left(\hat{n}^\top J^t J^{t\top}\hat{n}\right). \tag{11}$$

The approach is considered naive for several reasons. Firstly, the Lyapunov exponent is largely dependent on the choice of initial vector $\boldsymbol{X}(0)$, which means if $\boldsymbol{X}(0)$ is unluckily located on some periodic orbit, we will get some λ close to zero. Moreover, even if $\boldsymbol{X}(0)$ happens to be a 'generic' state space point, it is still not clear that the Lyapunov exponent could reflect anything in particular [13]. Finally, the observed temporal length t may also influence the accuracy and reliability of the computed Lyapunov exponent.

Therefore, two other more advanced methods, Wolf's Algorithm [7] and Spatial-Temporal Algorithm [9], are considered, which will be discussed later.

2.5 Wolf's Algorithm

Wolf's algorithm for calculating Lyapunov exponents is widely utilized to quantify the rate at which nearby trajectories in a dynamical system diverge. Instead of relying solely on one trajectory with initial point $\boldsymbol{X}(0)$ as shown in Eq. 11, the algorithm selects multiple trajectories in a temporal data set. This approach mitigates the risk of selecting a non-representative trajectory, such as one located on a periodic orbit or within atypical attractors, thereby ensuring a more robust and accurate estimation of the Lyapunov exponent. However, the method does not incorporate spatial information if the points are correlated, such as the Laplacian in Eq. 1. Consequently, this approach focuses primarily on the temporal Lyapunov exponent

without accounting for spatial dynamics.

STEP 1: Phase Space Reconstruction [10]

During the time when Wolf's Algorithm was developed, people tried to solve the nonlinearity measurement issue when only limited observation could be done during the experiment. That is to say, people need to reconstruct the original phase space based on the scalar time series they have, i.e., scalar measurement of only one state variable. This step is essential for analyzing the system's underlying dynamics, which will be reviewed thoroughly in Section 2.7.

Given a time series γ:

$$\gamma = \{x_i\}, i = 1, \ldots, N, \tag{12}$$

we could recontruct the phase space $\mathbf{\Gamma}$ as :

$$\mathbf{\Gamma}^{m,\tau} = \left\{ \boldsymbol{X}_i^{m,\tau} = \left(x_i(\boldsymbol{k}), x_{i+\tau}(\boldsymbol{k}), \ldots, x_{i+(m-1)\tau}, (\boldsymbol{k}) \right) \right\}, i = 1, \ldots, N - m\tau + \tau. \tag{13}$$

where

- i is the time step.
- m is the embedded dimension of the time series.
- τ is the lagging of the time series.
- N is the total time steps of the time series.

Theoretically, if given high enough embedded dimension m and proper lagging τ (will be discussed more in Section 2.7), we could say $\mathbf{\Gamma}^{m,\tau}$ could be considered topologically equivalent or diffeomorphic to the true phase space $\mathbf{\Gamma}$ [10], i.e., having the same Lyapunov exponent, where the scalar measurement x, for instance, can be the membrane potential u or the APD of u in 3V SIM model. To summarize, in the 3V SIM model it can be shown as:

$$\mathbf{\Gamma}^{m,\tau} \cong \mathbf{\Gamma} = \left\{ \boldsymbol{X}_i = \left(u_i, v_i, w_i \right) \right\}, i = 1, \ldots, N. \tag{14}$$

In the later steps, for simplicity, the reconstructed phase space $\mathbf{\Gamma}^{m,\tau} = \left\{ \boldsymbol{X}_i^{m,\tau} \right\}$, will be used as the input data set.

STEP 2: Identify Nearby Trajectories

For a point in the phase space, find a nearby point (a neighbor) that lies on a different trajectory. This neighbor should be close in space (both their magnitude and direction) but not necessarily in time to avoid correlations between temporally adjacent trajectories.

Specifically, a point $\boldsymbol{X}_i$ where $i \approx \frac{N}{2}$ in first iteration and $i = i'$ otherwise, is chosen, then by iterating through $\mathbf{\Gamma}$, we find its nearest neighbor $\boldsymbol{X}_j$ by calculating the Euclidean distance:

$$j = \arg\min_j \| \boldsymbol{X}_i^{m,\tau} - \boldsymbol{X}_j^{m,\tau} \| = \arg\min_j L_i < \epsilon, \tag{15}$$

where

- $j \in [1, N - m\tau + \tau]$.
- $j \neq i$.
- $\frac{\boldsymbol{X}_i^{m,\tau} \boldsymbol{X}_j^{m,\tau}}{|\boldsymbol{X}_i^{m,\tau}||\boldsymbol{X}_j^{m,\tau}|} < \theta$.
- $\theta = \frac{\pi}{9}$ is the maximum initial angular distance.
- ϵ is the maximum initial separation.

If the algorithm cannot find a close enough pair whose Euclidean distance is smaller than ϵ, it should report to the user and change the ϵ accordingly.

STEP 3: Evolve and Measure Divergence

Once we obtain a pair of neighbors, following Eq. 10, the finite-time Lyapunov exponent is then computed by monitoring the exponential divergence of the trajectory difference over a certain time interval, indicating a chaotic behavior.

Evolve L_i by one time step each until:

$$\|\boldsymbol{X}_{i'}^{m,\tau} - \boldsymbol{X}_{j'}^{m,\tau}\| = L_{i'} > \epsilon, \tag{16}$$

or

$$i' = N - m\tau + \tau \text{ or } j' = N - m\tau + \tau, \tag{17}$$

where ϵ should be chosen sufficiently large to ensure the two neighbors exhibit chaotic behavior.

If the evolved distance exceeds ϵ, repeat **STEP 2 & 3**. If not, break the loop and continue to **STEP 4**.

STEP 4: Measure the Lyapunov Exponent [14]

Having obtained multiple finite-time Lyapunov exponents throughout the iterations, the next step is to compute an averaged value, yielding a more robust estimate of the system's Lyapunov exponent.

During the loop, record all the $\frac{L_{i'}}{L_i}$, the i in first loop as i_0 and the i' in last loop as i_f. Finally, the Lyapunov exponent is:

$$\lambda = \frac{1}{i_f - i_0} \sum_{\text{All Loops}} \log_2 \frac{L_{i'}}{L_i}. \tag{18}$$

2.6 Spatial-Temporal Algorithm [9]

Both spatial and temporal factors matter in nonlinear dynamical systems with spatial correlations, such as those governed by Laplacian operators. Classical Lyapunov exponent algorithm, for example, Eq. 10 or Wolf's Algorithm, mainly focuses on temporal chaos. While spatial-temporal Lyapunov exponent (SLE) [9] can quantify how local perturbations evolve and spread through space over time. Therefore, the key challenge this algorithm tries to solve is to understand how chaos manifests when spatial interactions play a role in the system's dynamics, where the growth of perturbations at one point can influence nearby regions. This provides a more comprehensive understanding of the system's global dynamics by capturing both temporal and spatial instability.

A two-dimensional square map with a resolution of 512^2 pixels (based on the 3V SIM model) will be applied, with a variable x (could be u or APD) at each pixel serving as the input time series for the purpose of the algorithm's demonstration.

Now, a 2D map $\boldsymbol{\Lambda}^2(L)$ is defined as:

$$\boldsymbol{\Lambda}^2(L) = \{\boldsymbol{k} = (\alpha, \beta) \mid 1 \leq \alpha, \beta \leq L\}, \tag{19}$$

where:

- $L = 512$ is the length of map.
- $\boldsymbol{k}$ is the pixel in map.
- α, β is the x, y coordinate of the pixel $\boldsymbol{k}$.

STEP 1: Phase Space Reconstruction

Then for point $\boldsymbol{k}$, we have the time series γ below:

$$\gamma(\boldsymbol{k}) = \{x_1(\boldsymbol{k}), x_2(\boldsymbol{k}), \ldots, x_N(\boldsymbol{k})\}. \tag{20}$$

Similar to Eq. 13, we could reconstruct the phase space as

$$\mathbf{\Gamma}^{m,\tau}(\boldsymbol{k}) = \left\{ \boldsymbol{X}_i^{m,\tau}(\boldsymbol{k}) = \left(x_i(\boldsymbol{k}), x_{i+\tau}(\boldsymbol{k}), \ldots, x_{i+(m-1)\tau}, (\boldsymbol{k}) \right) \right\}, i = 1, \ldots, N - m\tau + \tau. \tag{21}$$

Therefore, if we consider all embedded vectors in the map, we could see:

$$\mathbf{\Gamma}^{m,\tau}(\mathbf{\Lambda}^2) = \bigcup_{\boldsymbol{k} \in \mathbf{\Lambda}} \mathbf{\Gamma}^{m,\tau}(\boldsymbol{k}). \tag{22}$$

STEP 2: Identify Nearby Trajectories

For each vector $\boldsymbol{X}_i^{m,\tau}(\boldsymbol{k}) \in \mathbf{\Gamma}^{m,\tau}(\boldsymbol{k}), \forall \boldsymbol{k} \in \mathbf{\Lambda}^2$, we search its neighbor $\boldsymbol{h} \in \mathbf{\Lambda}^2, \boldsymbol{h} \neq \boldsymbol{k}$ such that the following condition holds:

$$\|\boldsymbol{X}_i^{m,\tau}(\boldsymbol{k}) - \boldsymbol{X}_i^{m,\tau}(\boldsymbol{h})\| = \sqrt{\sum_{i'=i}^{i+m\tau-\tau} \left(\boldsymbol{X}_{i'}^{m,\tau}(\boldsymbol{k}) - \boldsymbol{X}_{i'}^{m,\tau}(\boldsymbol{h}) \right)^2} < \epsilon, \tag{23}$$

where ϵ is the maximum initial separation.

STEP 3: Evolve and Measure Divergence

At time step i, we name the certain neighboring pair by $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$. Then we evolve the pair to one more step further, which is to calculate

$$\|\boldsymbol{X}_{i+1}^{m,\tau}(\boldsymbol{k}) - \boldsymbol{X}_{i+1}^{m,\tau}(\boldsymbol{h})\|. \tag{24}$$

STEP 4: Measure the Lyapunov Exponent

Finally, SLE is defined as:

$$\lambda(m, \tau) = \frac{1}{N_{pair}} \sum_{i=1}^{N-m\tau} \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} \log \left(\frac{\|\boldsymbol{X}_{i+1}^{m,\tau}(\boldsymbol{k}) - \boldsymbol{X}_{i+1}^{m,\tau}(\boldsymbol{h})\|}{\|\boldsymbol{X}_i^{m,\tau}(\boldsymbol{k}) - \boldsymbol{X}_i^{m,\tau}(\boldsymbol{h})\|} \right). \tag{25}$$

2.7 Phase Space Reconstruction

Phase space reconstruction is a fundamental technique in the analysis of nonlinear dynamical systems, especially for systems where direct observation of all state variables is not easy. The concept is based on Takens' Embedding Theorem [10], which provides the theoretical foundation for reconstructing a system's phase space from time series data. Two parameters need to be decided: embedded dimension m and lagging τ. The lagging τ is typically selected first, followed by determining the embedding dimension m.

Lagging: τ

Embeddings with the same m but different τ are equivalent in the mathematical sense for noise-free data [15]. Therefore, for the purposes of this research, where simulations are conducted without the influence of noise, a simple choice of $\tau = 1$ is sufficient. However, in the presence of noise, a smaller value of lagging will result in reduced variance within $m\tau$, or we can say most embedded vectors clustering around the diagonal in the $\mathbb{R}^m$ – a phenomenon referred to as redundancy in [16], then if noise variance is larger than the variance of $m\tau$, the calculation would be nonsense.

For general cases, i.e., for experimental data contaminated by noise, common methods for determining lagging would be the mutual information method [17] and autocorrelation function [18]. When selecting τ, it is undesirable to be too small, as explained before, and also not too large, which might cause embedded vectors to occupy a broad region in the $\mathbb{R}^m$, leading to the deterministic structures [15] being confined to

very small scales.

Embbeded Dimension: m

The embedding dimension refers to the number of delayed elements for the time series $\gamma = \{x_i\}$ used to construct the embedded vector $\boldsymbol{X}_i$. To reconstruct phase space successfully, the sufficient condition would be [10],[19]:

$$m > 2d, m > 2d_c. \tag{26}$$

where:

• d is the phase space dimension.

• d_c is the box-counting dimension (similar to the correlation dimension) of strange attractors.

So, for example, in the 3V SIM model, according to Takan's theorem, the embedded dimension can guarantee a successful phase reconstruction if it is larger than 2*3 (3 state variables), which is 7. However, there is another reference stating [20] that the embedded dimension can be:

$$m \approx d, \tag{27}$$

where it successfully reconstructs the Lorenz system ($d = 3$) with $m = 3$ by plotting the reconstructed phase space and original phase space and showing that they are geometrically same.

Therefore, we may conclude that m in $[d, 2d]$ or $[d, 2d_c]$ could be appropriate, with its exact minimum value depending on the nature of the dataset being analyzed. Several methods exist for determining an optimal embedding dimension, including the False Nearest Neighbor (FNN) method [21], the Characteristic Decay Rate [9], the Fuzzy Clustering [22], the Fill-Factor method [23], the Average Integral Local Deformation (AILD) method [23], and so on. A comprehensive summary of these algorithms can be found in Jiang and Adeli [22]. Most of the approaches above rely on the fact that with an embedded dimension larger than a minimum value, the reconstructed phase space would be geometrically the same as the true phase space. Furthermore, a reasonable data length required to determine the embedded dimension is at least 10^m [24]. Accordingly, the success of those algorithms requires the dynamical system to behave in a low dimension with sufficient long data as input.

In the following part, I will provide a detailed review of the False Nearest Neighbor (FNN) method, which is implemented to determine the embedding dimension in this research.

STEP 1: Phase Space Reconstruction

$\tau = 1$ is obtained before, and we need to initialize different embedded dimensions $\{m\}$.

For every embedded vector with a certain m:

$$\boldsymbol{X}_i^{m,\tau=1} = \{x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(m-1)\tau}\}, i = 1, 2, \ldots, N - m\tau + \tau. \tag{28}$$

STEP 2: Identify Nearby Trajectories

Find its nearest neighbor $\boldsymbol{X}_j^m$ such that $j = \arg\min_j \|\boldsymbol{X}_i^m - \boldsymbol{X}_j^m\| = \arg\min_j R_i^m$.

STEP 3: Evolve and Count False Neighbors

We get $R_i^{m+1} = \|\boldsymbol{X}_i^{m+1} - \boldsymbol{X}_j^{m+1}\|$. Now i, j will be identified as a pair of false neighbors if $|R_i^m - R_i^{m+1}| \gg 0$.

Specifically, if the ratio:

$$R(i, m) = \sqrt{\frac{(R_i^m)^2 - (R_i^{m+1})^2}{(R_i^m)^2}} \tag{29}$$

is bigger than a pre-defined value $R_0 = 10$, we call $\{i, j\}$ a pair of false neighbors or point i has a false neighbor j.

STEP 4: Find Correct Embedded Dimension

Plot the percentage $\frac{\text{false pairs of neighbors}}{\text{total pairs of neighbors}}$ for every m.

Now, if at a certain embedded dimension m_0, the percentage of false nearest neighbors does not decrease, we could identify m_0 as the correct dimension, where, for instance, $m_0 = 4$ from Fig. 3.

2.8 Noise and Chaos Distinguishment

Both algorithms, especially Wolf's algorithm, have been criticized because they assume and do not test for the exponential divergence of the orbits [15],[25]. As a result, these algorithms cannot effectively distinguish between chaos and noise. Therefore, including a robust method for distinguishing between noise and chaos is necessary for this research. Now, the fundamental difference is that chaos arises from deterministic, nonlinear processes. This means that chaotic systems are sensitive to initial conditions but follow predictable paths for a short period of time. Nevertheless, noise (typically uncorrelated one) is random, leading to unpredictable behavior that is not governed by deterministic equations.

Many methods exist for noise and chaos distinguishment, including the Simplex Projection method [26], Return Map method, Correlation Dimension method[8], Permutation Entropy [27], Machine Learning method, etc.

In this study, the Simplex Projection method is utilized and reviewed. Notice that this method focuses on uncorrelated noise, while more details on autocorrelated noise can be found in [28]. Simplex projection can be described as a nearest-neighbor forecasting algorithm [29]. It divides input time series into training and prediction sets, where similar past embedding vectors in the training set could forecast those in the prediction sets. Since chaos and noise have essential differences in their predictability, their performance on prediction scores in the testing set differs. Chaos shows a decreasing prediction score curve over time, reflecting the short-term predictability. In contrast, noise results in a flat prediction score curve due to its lack of a deterministic structure.

STEP 1: Phase Space Reconstruction[28],[30]

To begin with, for a certain embedded dimension m and lagging τ of the input data set $\gamma = \{x_i\}$, we have its embedded version $\mathbf{\Gamma}^{m,\tau} = \{\boldsymbol{X}_i^{m,\tau}\}$ as defined in Eq. 13. Then the prediction set is defined as:

$$\{\hat{x}_{i+s}\} = \{f(\boldsymbol{X}_{i-m\tau}^{m,\tau}, s)\}, i = N/2, \ldots, N, \tag{30}$$

where s is the prediction step.

STEP 2: Identify Nearby Trajectories

Then we find 3 nearest neighbors with index $\{j, k, l\}$ to embedded data at step i:

$$j = \arg\min_{j \neq i} \|\boldsymbol{X}_{i-m\tau}^{m,\tau} - \boldsymbol{X}_{j-m\tau}^{m,\tau}\|, j \in [1 + m\tau, N - s], \tag{31}$$

$$k = \arg\min_{k \neq i,j} \|\boldsymbol{X}_{i-m\tau}^{m,\tau} - \boldsymbol{X}_{k-m\tau}^{m,\tau}\|, k \in [1 + m\tau, N - s], \tag{32}$$

$$l = \arg\min_{l \neq i,j,k} \|\boldsymbol{X}_{i-m\tau}^{m,\tau} - \boldsymbol{X}_{l-m\tau}^{m,\tau}\|, l \in [1 + m\tau, N - s]. \tag{33}$$

STEP 3: Predict

After that, we predict $\hat{x}_{i+s}$ as:

$$\hat{x}_{i+s} = f(\boldsymbol{X}_{i-m\tau}^{m,\tau}, s) = \frac{\sum_{i'=\{j,k,l\}} \frac{x_{i'+s}}{\|\boldsymbol{X}_{i-m\tau}^{m,\tau} - \boldsymbol{X}_{i'-m\tau}^{m,\tau}\|}}{\sum_{i'=\{j,k,l\}} \frac{1}{\|\boldsymbol{X}_{i-m\tau}^{m,\tau} - \boldsymbol{X}_{i'-m\tau}^{m,\tau}\|}}, \tag{34}$$

where we weight the neighbors by their reciprocal of the Euclidean distance to $\boldsymbol{X}_{i-m\tau}^{m,\tau}$.

STEP 4: Plot Correlation Coefficient (Prediction Score)

To proceed, we plot $\{X_{i+s}\}$ vs. $\{\hat{X}_{i+s}\}$, as shown in Fig. 5 and get the correlation coefficient (prediction score) as:

$$\rho(s) = \rho_{\{X_{i+s}\},\{\hat{X}_{i+s}\}} = \frac{\mathrm{Cov}\left[\{X_{i+s}\},\{\hat{X}_{i+s}\}\right]}{\sigma_{\{X_{i+s}\}}\sigma_{\{\hat{X}_{i+s}\}}} \tag{35}$$

Finally, by plotting the correlation coefficient $\rho(s)$ vs. s, we could distinguish between chaos and noise, as shown in Fig. 5.

3 Proposed Work

3.1 Motivation

Quantifying the chaos of the cardiac system is crucial to heart health, particularly in the early detection of arrhythmia and potentially fatal fibrillation. This research now tries to quantify chaotic behavior within the cardiac system using simulations without requiring full knowledge of all state variables. By focusing on the APD from voltage data, it has the potential to be directly applied in clinical settings, where the Electrocardiogram (ECG) can provide sufficient APD information. Also, with a good understanding of the chaotic, transition, and periodic region for different parameters (here I show an example for τ_d in Section 4.3), it is essential for drug development because it helps pinpoint which particular regions of parameters are sensitive and likely to induce chaotic behavior. Drugs targeting the heart should carefully modulate the parameters to avoid the chaotic regions that may trigger deadly chaos.

Notably, the Spatial-Temporal Algorithm requires fewer data points ($\sim$ 200 APD) than traditional temporal methods ($\sim$ 20000 APD), allowing for rapid measurement of chaos. Furthermore, the influence of noise is also considered, with noise filtering algorithms applied in the future to enhance the robustness when dealing with the experimental data.

3.2 Simulation

The simulation is browser-based (original model coded by Dr. Abouzar Kaboudian) and can be found here: 3V model [31]. It is implemented by WebGL using Abubu.js, which is a library that can utilize GPU using easier syntax than the classical WebGL, allowing researchers unfamiliar with graphics programming to focus on the numerical aspects. Abubu.js uses texture as the primary data structure, where each texture contains 512*512 pixels and every pixel has four channels: red (r), green (g), blue (b), alpha (α, transparency). Here in this 3V SIM model, only three channels (r, g, b) are used to store variables (u, v, w) and α is set to be 1.0.

When solving the PDEs in Eq. 1,2,3, finite difference and forward Euler methods are applied respectively to solve Laplacian ∇^2 and to integrate over time. The unit time $dt = 0.1$ ms and unit space $dx = dy = 18/512$ cm. The boundary condition is using GL_CLAMP_TO_EDGE in WebGL, which means if the coordinates of pixels are $[0, 1]^2$, then any coordinate greater than 1.0 is set to 1.0, and any coordinate less than 0.0 is set to 0.0 [32].

The same initial condition I coded is applied to all different sets of parameters, as shown in Fig. 2. After Fig. 2(A,B), persistent fibrillation will be generated, which means sustainable and stable spiral dynamics will be generated for nonlinearity calculation.

Transient $T_t = 20000$ has been discarded, which is around 50 APD per pixel, and simulation time $\Delta T = 50000$ has been used for data acquisition (for Spatial-Temporal Algo, and $\Delta T = 320000$ for Wolf's Algo, as shown in Tab. 1). As shown in Fig. 2(C), totally 25^2 data points are recorded and they are evenly distributed in the map. The recording happens every 10 dt or every $\Delta t = 2$

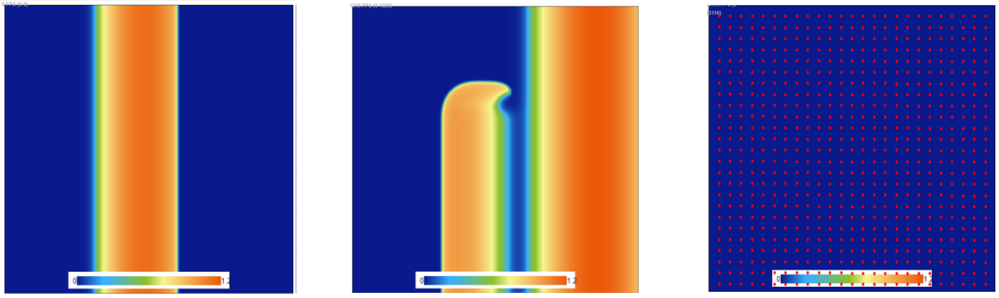

Figure 2: Initial condition: (A) Begin with voltage activation having a square shape from the left edge. (B) When the most left part of the square passes the middle of the map, another activation square in the middle bottom starts. (C) Red dots (25*25 dots in total) are the locations of data recorded.

Parameter	**Symbol**	**Value**
Simulation Time Step	$d\tau$	0.1 ms
Measurement Time Step	dt	4 ms
Space Step	dx, dy	$\frac{18}{512}$ cm
Texture Size	\	$512 * 512$ pixels
Measurement Pixels	$\{\boldsymbol{k}\}$	$25 * 25$ pixels
Transient Time	T_t	20000
Measurement Time	ΔT	[20000, 70000], [20000,340000]
# of Voltage per Pixel	$\{u(\boldsymbol{k})\}$	12500, 1250000
# of APD per Pixel	$\{\text{APD}(\boldsymbol{k})\}$	~ 200, ~ 20000

Table 1: Information of the input data. The red color is the input data for Spatial-Temporal Algo, and the blue color is for Wolf's Algo

3.3 Chaos Quantification of 3V SIM model

By adjusting $\tau_d \in [0.4, 0.42]$ and more parameters in the future, I plan to identify chaotic and transition regions (chaos to quasiperiodic behavior) within the 3V SIM model. In parallel, I will compare the results derived from APD with those from the full state-space analysis to demonstrate that APD can effectively be used for chaos quantification. Additionally, the Spatial-Temporal Algorithm with less APD needed will also be evaluated. Moreover, certain meandering phenomena observed in the 3V SIM model and relevant experimental data will be quantified. Future work will incorporate additional parameters from the model, allowing for a more comprehensive understanding of its complex dynamics. Furthermore, more advanced simulation models, such as the OVVR human model [33] and stochastic Beeler-Reuter model [4] and experiments, will be included.

3.4 Proposed Timeline

I have been working on the chaos quantification of the cardiac simulations in 2024. I plan to finish this research on finding chaotic and non-chaotic regions for different parameters in the 3V SIM Model with a manuscript submitted before the end of the year 2024.

I will begin to work on the chaos quantification of more advanced models and experimental data from the year 2025.

4 Current Progress

4.1 Embedded Dimension

To embed the APD data, I used FNN to find the optimized Embedded Dimension.

As described in Section 2.7, I plotted the percentage of false neighbors percentage vs. embedded dimension, where the input data (shown in Tab. 1 with red color) is the APD sequence (120 APD) of a pixel (index 25, randomly chosen with the results similar for all points) in the 512^2 map and $\tau_w^+ = 350, \tau_d = [0.4, 0.401, \ldots, 0.42]$.

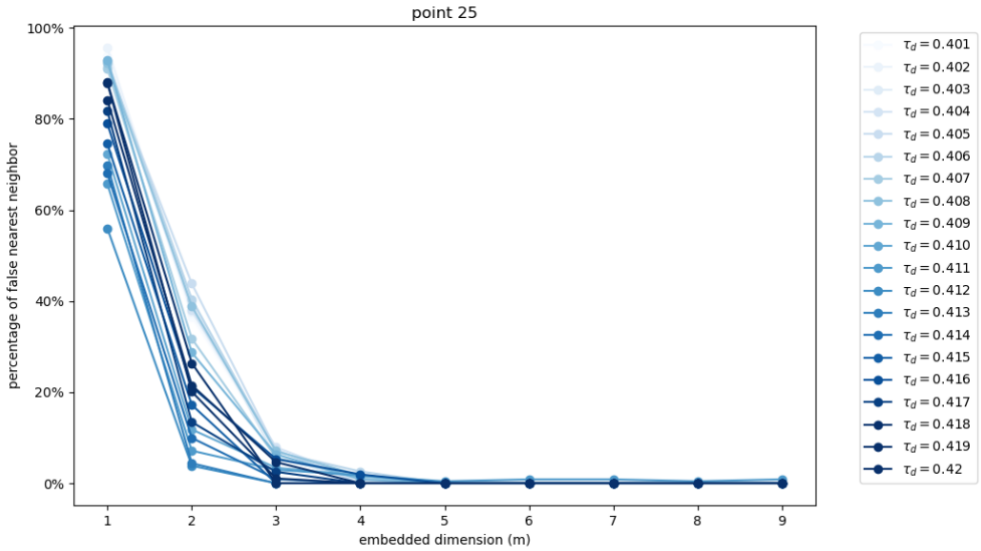

Figure 3: The percentage of false nearest neighbours with input data $x \equiv$ APD, with lag $\tau = 1$.

It could be seen that the false neighbors start to converge after $m = 4$. Therefore, we could decide the embedded parameter for APD to be $m = 4, \tau = 1$

4.2 Stochastic testing

Since neither algorithm assesses the exponential divergence of the orbits, they cannot effectively distinguish between chaos and noise. Also, in the experimental setting, measurement noise is a large part of the signal, which cannot be disregarded. Therefore, stochastic testing on current simulation data is necessary to ensure robustness when considering future experimental data. Because membrane potential u is the most important variable in state space and the origin of APD, I do the stochastic testing solely on u. By selecting one pixel in the input data and doing the stochastic testing, as shown in Fig. 4, the difference between chaos and noise can be seen in Fig. 5, where the decrease in predictability would be a key factor of chaos. Some details of correlation map is shown in Fig. 6.

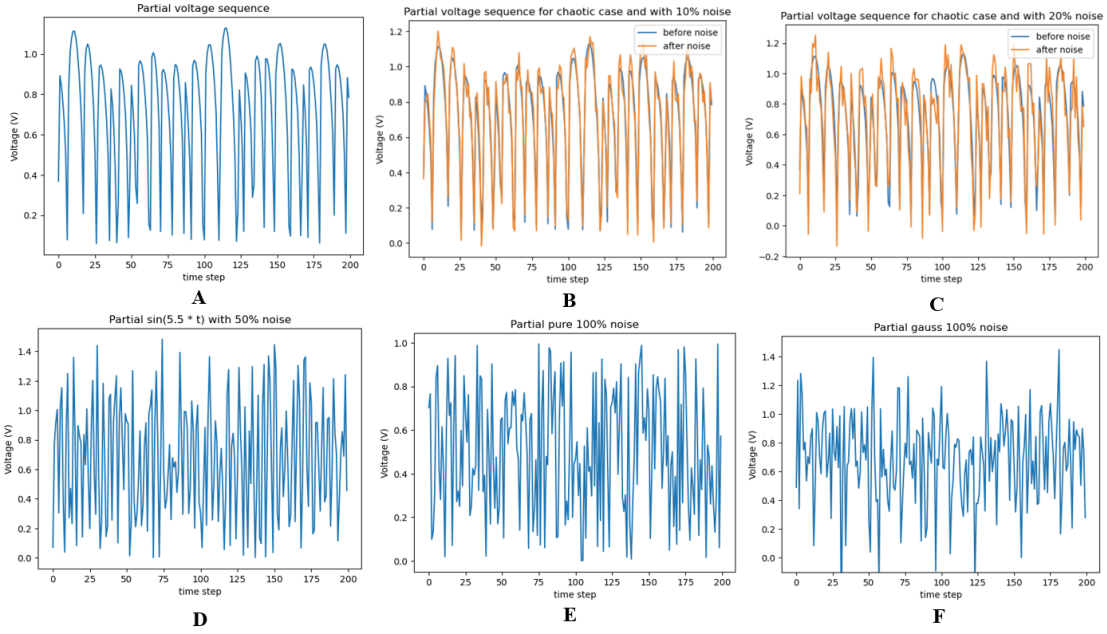

Figure 4: For the following plots, only a partial curve is shown for better visualization. (A) Data input for $\tau_d = 0.405$, (B) Voltage from (A) but with 10% noise, (C) similar to the previous one but with 20% noise, (D) noise generated by sin-wave plus 50% noise, (E) 100% uniform noise with range [0,1], (F) 100% gaussian noise with mean and standard deviation same as data input.

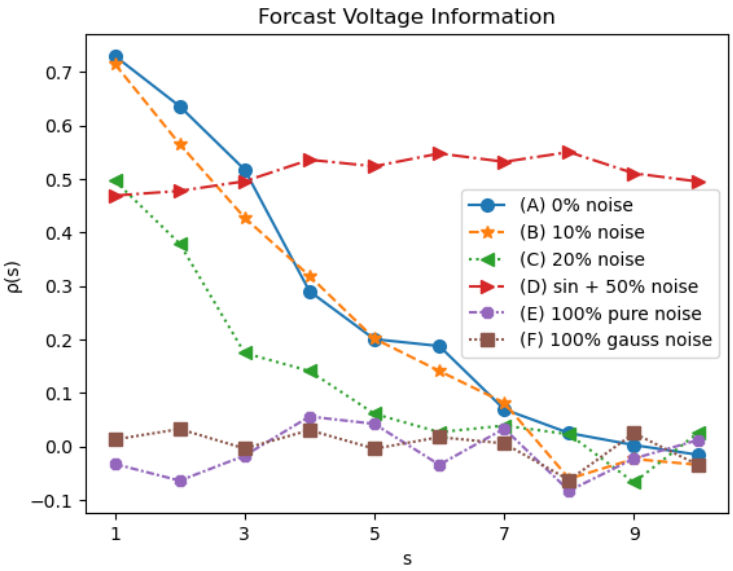

Figure 5: Forcast Plot. $\rho(s)$ is the correlation coefficient (prediction score) and s is the prediction step. More details can be seen in Eq. 35.

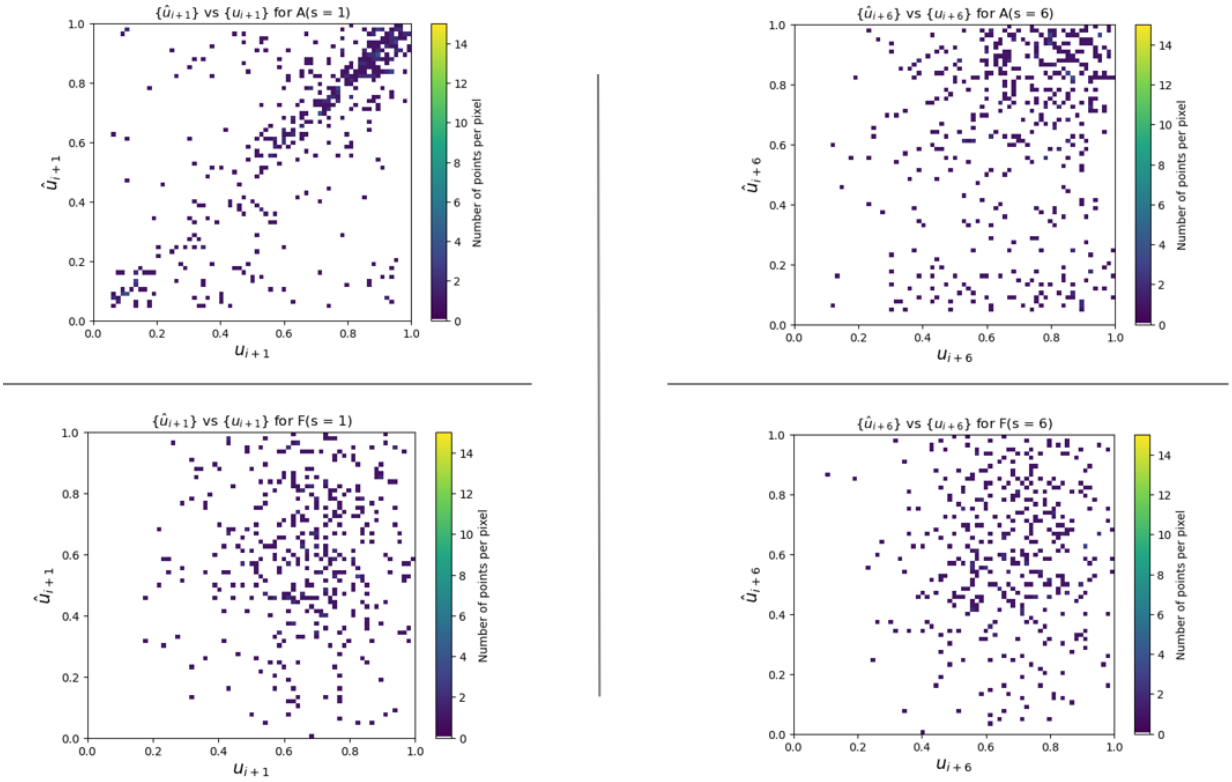

Figure 6: Predicted vs Observed Voltage. The correlation between predicted and observed Voltage is the $\rho(s)$ in Fig. 5. Top Left: A($s = 1$), $s = 1$ (prediction step = 1) point for curve A in Fig. 5. Top Right: A($s = 6$). Bottom Left: F($s = 1$). Bottom Right: F($s = 6$).

4.3 Chaos Quantification for τ_d

Here, I compare the results from the APD to the ones from full-state space using Wolf's Algorithm. The results indicate that the APD data preserves the nonlinear properties of the system. Using the Spatial-Temporal Algorithm also shows that similar outcomes could be achieved with less APD. The Spatial-Temporal Algorithm focuses on the one-step divergence, as shown in Eq. 24, which makes it hard to calculate voltage data since voltage usually performs divergence in multiple steps (after 1 APD). This is the reason for not including voltage data within the Spatial-Temporal Algorithm. In Fig. 7, the plot shows that the result from APD performs well, and we can see a high variation in the transition region (state B) due to it having a combination of both chaotic and quasiperiodic patterns.

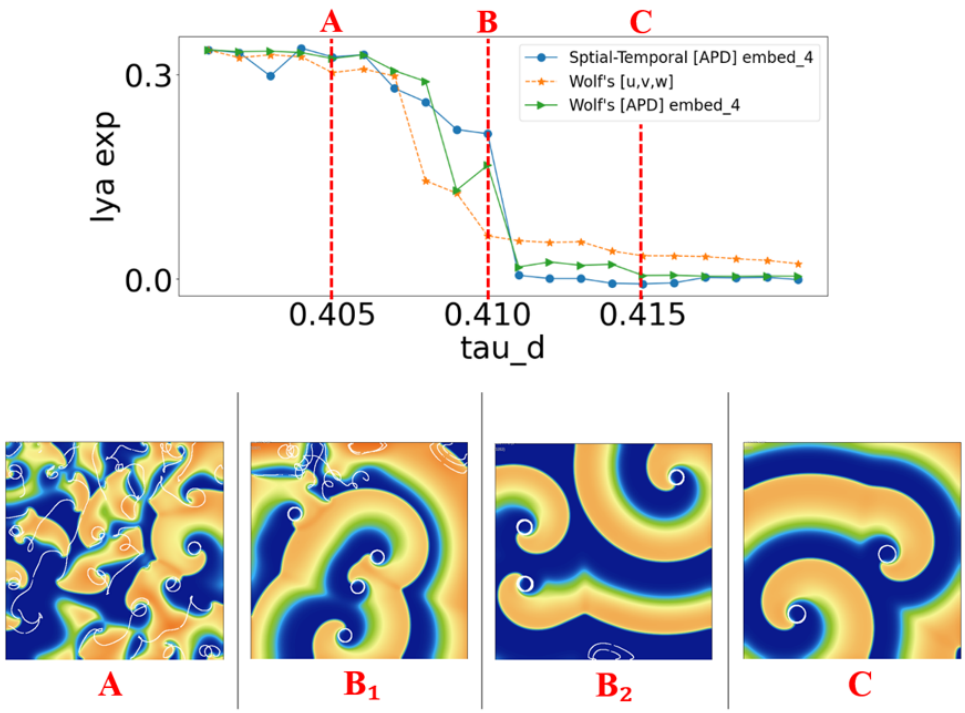

Figure 7: Top: Lyapunov Exponent (scaled) for τ_d where A, B, C represents different tau_d state. Bottom: Actual simulations of different τ_d states. The white line represents the tip trajectory of the spiral wave. B_1 and B_2 represent two possibilities that the B state could become. State A stays chaotic, state B stays either less chaotic or quasiperiodic, and state C stays only periodic.

4.4 Future Work: Chaos Quantification for Experiment and More

Future work will focus on applying the chaos quantification method to experimental data, as demonstrated in Fig. 8, which I presented at SIAM AN 2024. The "temporal lya exp" was calculated by TISEAN, a widely used nonlinear analysis software [15] and the spatial one was by Spatial-Temporal Algorithm. In addition, Fig. 9 showed the Lyapunov Exponent for different meander cases in the 3V SIM Model. Simulations by more advanced models will be tested in the future as well.

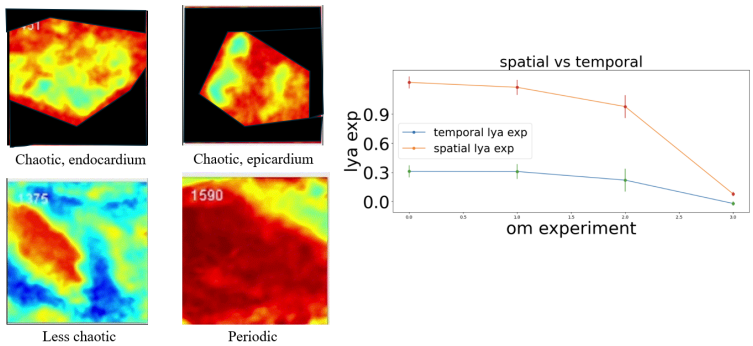

Figure 8: Lyapunov Exponent for Optical Mapping (OM) [34] Experiment

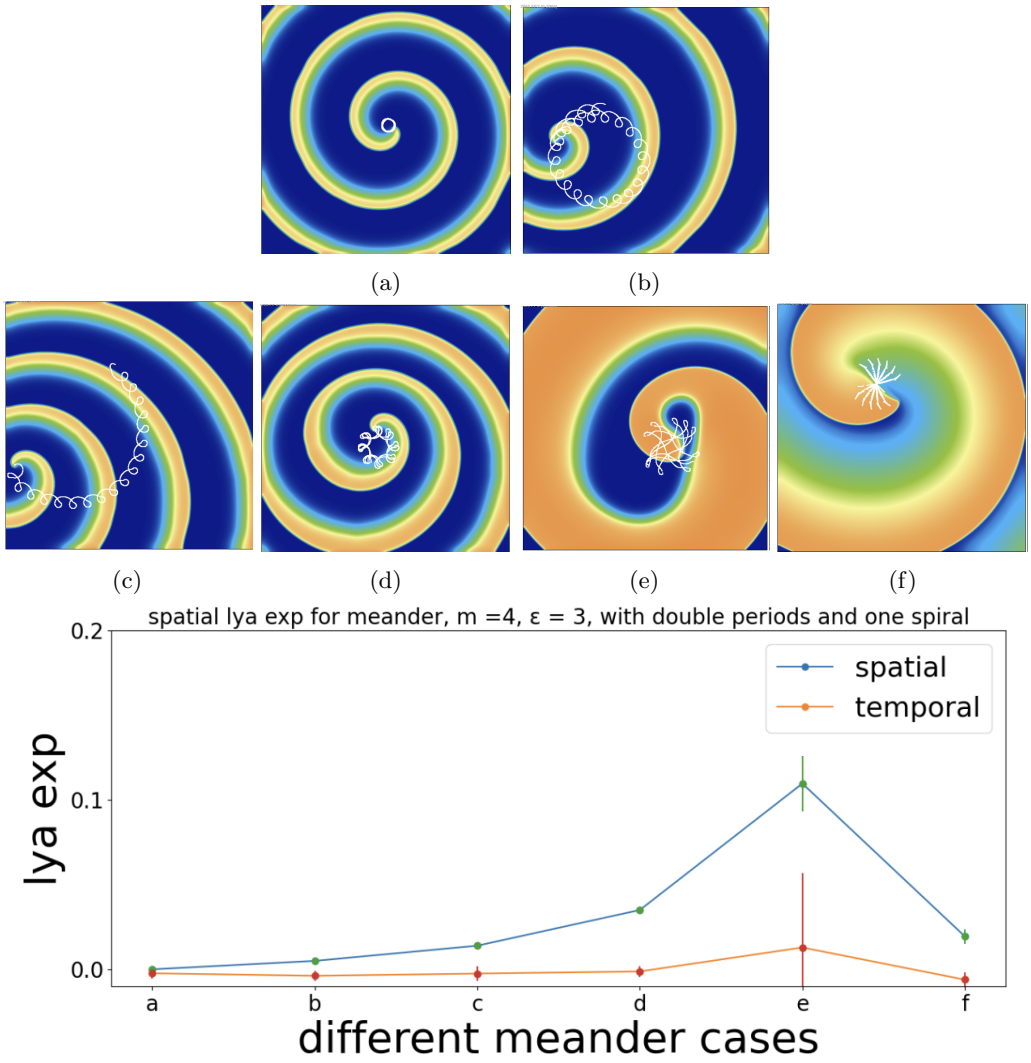

Figure 9: (a) to (e): Different meander cases in 3V SIM Model. [(a): $\tau_d = 0.41$; (b): $\tau_d = 0.392$; (c): $\tau_d = 0.381$; (d): $\tau_d = 0.36$; (e): $\tau_d = 0.25$; (f): parameter set 2[1]].

5 Conclusion

The 3V SIM Model is a simplified cardiac model, retaining essential activation and inactivation characteristics while having fewer variables. With simulations in this model, I showed that APD data could be an alternative choice for determining the Lyapunov exponent by comparing it to results from full state space. The possible application of this research in the future is to quantify the complexity of the cardiac system without knowing exact voltage data in experimental and clinical settings. What's more, integrating spatial information with the Spatial-Temporal Algorithm could significantly reduce the amount of APD data needed, enabling quicker and even real-time determination of the Lyapunov exponent.

[1]Parameter set 2 was listed in the appendix of [1]. (a) to (e) used parameter set 1. For all other simulation data in this research, they used parameter set 4 (but tuning τ_d and τ_w^+). The applications of all parameter sets have been discussed in Section 2.2

References

[1] F. Fenton and A. Karma, "Vortex dynamics in three-dimensional continuous myocardium with fiber rotation: Filament instability and fibrillation," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 8, no. 1, pp. 20–47, 1998.

[2] E. M. Cherry and F. H. Fenton, "Suppression of alternans and conduction blocks despite steep apd restitution: electrotonic, memory, and conduction velocity restitution effects," *American journal of physiology-Heart and circulatory physiology*, vol. 286, no. 6, pp. H2332–H2341, 2004.

[3] E. Cherry, F. Fenton *et al.*, "Minimal model for human ventricular action potentials in tissue," *J Theor Biol 253 (3): 544- 60*, 2008.

[4] M. Courtemanche and A. T. Winfree, "Re-entrant rotating waves in a beeler–reuter based model of two-dimensional cardiac electrical activity," *International Journal of Bifurcation and Chaos*, vol. 1, no. 02, pp. 431–444, 1991.

[5] C.-h. Luo and Y. Rudy, "A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction." *Circulation research*, vol. 68, no. 6, pp. 1501–1526, 1991.

[6] F. H. Fenton, E. M. Cherry, H. M. Hastings, and S. J. Evans, "Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 12, no. 3, pp. 852–892, 2002.

[7] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: nonlinear phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[8] P. Grassberger and I. Procaccia, "Characterization of strange attractors," *Physical review letters*, vol. 50, no. 5, p. 346, 1983.

[9] R. V. Solé and J. Bascompte, "Measuring chaos from spatial information," *Journal of Theoretical Biology*, vol. 175, no. 2, pp. 139–147, 1995.

[10] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*. Springer, 2006, pp. 366–381.

[11] H. Abarbanel, *Analysis of observed chaotic data*. Springer Science & Business Media, 2012.

[12] D. T. Kaplan and L. Glass, "Direct test for determinism in a time series," *Physical review letters*, vol. 68, no. 4, p. 427, 1992.

[13] P. Cvitanovic, R. Artuso, R. Mainieri, G. Tanner, G. Vattay, N. Whelan, and A. Wirzba, "Chaos: classical and quantum," *ChaosBook. org (Niels Bohr Institute, Copenhagen 2005)*, vol. 69, p. 25, 2005.

[14] A. Wolf *et al.*, "Quantifying chaos with lyapunov exponents," *Chaos*, vol. 16, pp. 285–317, 1986.

[15] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge university press, 2003.

[16] M. Casdagli, "Chaos and deterministic versus stochastic non-linear modelling," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 54, no. 2, pp. 303–328, 1992.

[17] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical review A*, vol. 33, no. 2, p. 1134, 1986.

[18] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[19] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, pp. 579–616, 1991.

[20] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch, "Detecting causality in complex ecosystems," *science*, vol. 338, no. 6106, pp. 496–500, 2012.

[21] M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical review A*, vol. 45, no. 6, p. 3403, 1992.

[22] X. Jiang and H. Adeli, "Fuzzy clustering approach for accurate embedding dimension identification in chaotic time series," *Integrated Computer-Aided Engineering*, vol. 10, no. 3, pp. 287–302, 2003.

[23] T. Buzug and G. Pfister, "Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors," *Physical review A*, vol. 45, no. 10, p. 7073, 1992.

[24] J. Theiler, "Estimating fractal dimension," *JOSA a*, vol. 7, no. 6, pp. 1055–1073, 1990.

[25] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "A practical method for calculating largest lyapunov exponents from small data sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117–134, 1993.

[26] G. Sugihara and R. M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," *Nature*, vol. 344, no. 6268, pp. 734–741, 1990.

[27] C. Bandt, "Permutation entropy and order patterns in long time series," in *Time Series Analysis and Forecasting: Selected Contributions from the ITISE Conference*. Springer, 2016, pp. 61–73.

[28] G. Sugihara and R. M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," *Nature*, vol. 344, no. 6268, pp. 734–741, 1990.

[29] C.-h. Hsieh, S. M. Glaser, A. J. Lucas, and G. Sugihara, "Distinguishing random environmental fluctuations from ecological catastrophes for the north pacific ocean," *Nature*, vol. 435, no. 7040, pp. 336–340, 2005.

[30] O. L. Petchey, "Simplex projection walkthrough," 2016.

[31] A. Kaboudian, E. M. Cherry, and F. H. Fenton, "Real-time interactive simulations of large-scale systems on personal computers and cell phones: Toward patient-specific heart modeling and other applications," *Science advances*, vol. 5, no. 3, p. eaav6019, 2019.

[32] "Webgl specifications," https://registry.khronos.org/webgl/specs/latest/1.0/.

[33] T. O'Hara, L. Virág, A. Varró, and Y. Rudy, "Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation," *PLoS computational biology*, vol. 7, no. 5, p. e1002061, 2011.

[34] V. Kappadan, A. Sohi, U. Parlitz, S. Luther, I. Uzelac, F. Fenton, N. S. Peters, J. Christoph, and F. S. Ng, "Optical mapping of contracting hearts," *The Journal of Physiology*, vol. 601, no. 8, pp. 1353–1370, 2023.