CODEE Journal

Volume 13 Article 2

11-30-2020

Extending Power Series Methods for the Hodgkin-Huxley Equations, Including Sensitive Dependence

James S. Sochacki
James Madison University

Recommended Citation

Sochacki, James S. (2020) "Extending Power Series Methods for the Hodgkin-Huxley Equations, Including Sensitive Dependence," *CODEE Journal*: Vol. 13, Article 2.
Available at: https://scholarship.claremont.edu/codee/vol13/iss1/2

Extending Power Series Methods for the Hodgkin-Huxley Equations, Including Sensitive Dependence

Cover Page Footnote (optional)

I thank G.E. Parker for all the years we talked about PSM, the JMU PSM team (S. Lucas, R. Thelwell, A. Tongen, P. Warne) for all their encouragements and thoughts and all the other mathematicians I met because of PSM. I also thank all the undergraduate students who over the last 30 years have worked with me on PSM, with a congratulations to those undergraduates who were able to get their work published. A special thanks to my undergraduate research student Jeffrey Kopsick for help in writing the codes that made this paper possible.

Extending Power Series Methods for the Hodgkin-Huxley Equations, Including Sensitive Dependence

James Sochacki
James Madison University

Keywords: Neuron Modeling, Hodgkin-Huxley Equations, Power Series, Sensitive Dependence
Manuscript received on October 30, 2020; published on November 30, 2020.

Abstract: A neural cell or neuron is the basic building block of the brain and transmits information to other neurons. This paper demonstrates the complicated dynamics of the neuron through a numerical study of the Hodgkin-Huxley differential equations that model the ionic mechanisms of the neuron: slight changes in parameter values and inputted electrical impulses can lead to very different (unexpected) results. The methods and ideas developed for the ordinary differential equations are extended to partial differential equations for Hodgkin-Huxley networks of neurons in one, two and three dimensions.

1 Introduction

In their 1952 paper that led to the 1963 Nobel Prize in Physiology, Alan Hodgkin and Andrew Huxley [1] laid out a path to understand the excitations (ionic mechanism) of a nerve fibre. The results they presented in the paper were for the nerve cell in a giant squid axon. Today a nerve cell is usually referred to as a neuron. A major tool that was used by these two mathematical physiologists to produce their results was the voltage clamp developed by the biophysicists Kenneth Cole [2] and George Marmont [3] in the spring of 1947.

There was not much known about a neuron in 1952 and the technology that was available to study a neuron in 1952 is not comparable to the technology that is used today to study neurons. This includes, both the experimental equipment (microscopes, voltage clamps, voltmeters, etc.) and the computers that were used to generate data and to solve the equations that the data allowed to be developed.

However, it is also *important* to point out (and remember) that these differential equations are still used today to improve our understanding of neurons and modifications of them are used to model neurons other than the giant squid's. A google search will show how many neuroscientists are studying these differential equations to gain a better understanding of neurons.

The statement “Details of the mechanism will probably not be settled for some time” in their paper was certainly true then and is still true today. Many neuroscientists both experimental and computational have unwillingly said - the more we learn about neurons ... the more we realize how complicated they are. In fact the neuron shown in Figure 1 was not available to Hodgkin and Huxley and more of the components of the neuron have been discovered recently. Figure 1 is adequate to develop the differential equations of Hodgkin and Huxley.

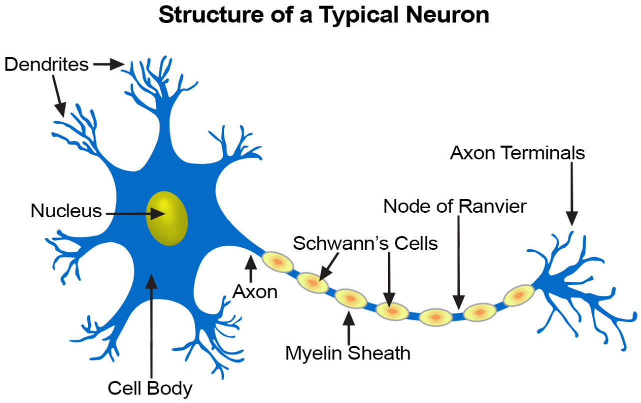

Figure 1: A typical neuron (nerve cell) structure.

Hodgkin and Huxley point out in their paper

> *The results presented here show that the equations derived in Part II of this paper predict with fair accuracy many of the electrical properties of the squid giant axon: the form, duration and amplitude of spike, both ‘membrane’ and propagated; the conduction velocity; the impedance changes during the spike; the refractory period; ionic exchanges; subthreshold responses; and oscillations.*

They also state

> *Our calculations of excitation processes were all made for the case where the membrane potential is uniform over the whole area considered, and not for the case of local stimulation of a whole nerve. There were two reasons for this: first, that such data from the squid giant fibre as we had for comparison were obtained by uniform stimulation of the membrane with the long electrode; and, secondly, that calculations for the whole nerve case would have been extremely laborious since the main equation is then a partial differential equation.*

In Part II, we will address this piece of their paper more. In Part I we will introduce the ordinary differential equations, some properties of these ordinary differential equations and figures demonstrating the information the ordinary differential equations give us about neurons and their complexity as pointed out by Hodgkin and Huxley.

In order to develop differential equations for neurons, Hodgkin and Huxley give the schematic in Figure 2 as a depiction of the neuron in Figure 1. (This schematic was taken directly from their paper.) Using the known theory for electro-dynamics and electro-chemistry, Hodgkin and Huxley derive differential equations for neurons.

In Part I, we will focus on the original Hodgkin-Huxley ordinary differential equations and in Part II, we will extend these ideas to address some of the more intricate details of these equations and the corresponding partial differential equations. The ordinary differential equations developed by Hodgkin and Huxley to model the electro-dynamics of the neuron have removable singularities in the ion rate parameters for the sodium and potassium concentration differential equations. Obtaining numerical solutions to these differential equations is difficult because of these removable singularities. Using composition rules for power series and approximating these rate parameters with functions not having singularities are methods that allow one to compute power series solutions around these removable singularities.

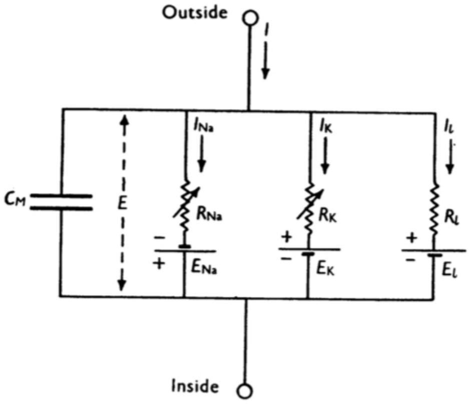

Fig. 1. Electrical circuit representing membrane. $R_{Na}=1/g_{Na}$; $R_K=1/g_K$; $R_l=1/\bar{g}_l$. R_{Na} and R_K vary with time and membrane potential; the other components are constant.

Figure 2: The Hodgkin-Huxley Circuit Schematic for a Neuron (Figure 1).

2 Part I. The Hodgkin-Huxley ODEs

The original Hodgkin-Huxley (HH) system of ordinary differential equations (ODEs) that can be derived using Ohm's voltage laws and Kirchoff's circuit laws from Figure 2 are

$$\begin{aligned} I_M(t) = C_M \frac{d}{dt}V(t) + \bar{g}_K\, n(t)^4\,(V(t) - E_K) + \\ \bar{g}_{Na}\, m(t)^3 h(t)\,(V(t) - E_{Na}) + \bar{g}_L\,(V(t) - E_L) \\ \frac{d}{dt}n(t) = \alpha_n(V(t))\,(1 - n(t)) - \beta_n(V(t))\,n(t) \\ \frac{d}{dt}m(t) = \alpha_m(V(t))\,(1 - m(t)) - \beta_m(V(t))\,m(t) \\ \frac{d}{dt}h(t) = \alpha_h(V(t))\,(1 - h(t)) - \beta_h(V(t))\,h(t) \end{aligned} \tag{2.1}$$

where V is the voltage in millivolts (mV) at time t (in milliseconds), n is the potassium (K) concentration, m is the sodium (Na) concentration, h is the concentration of various other ions (L for leakage, the word used by Hodgkin and Huxley), C_M is the membrane capacitance, E_K, E_{Na}, E_L are basis voltage potentials and produce electro-chemical currents, I_M is an input current to the neuron. The parameters $\bar{g}_K, \bar{g}_{Na}, \bar{g}_L$ are maximum conductance values (inverse of resistance). The function parameters $\alpha_n, \alpha_m, \alpha_h$ and $\alpha_n, \alpha_m, \alpha_h$ depend only on the voltage and monitor transition rate for the opening and closing of ion channel gates for potassium, sodium and the other concentrations, respectively. They will be shown to be significant in determining the electro-dynamics of the neuron.

This terminology used by Hodgkin and Huxley is still used today. Neuroscientist's goal is to determine $V(t), n(t), m(t)$ and $h(t)$ given the initial conditions $V(0) = V_0, n(0) = n_0, m(0) = m_0, h(0) = h_0$, the input impulse $I_M(t)$ and the parameters $C_M, \bar{g}_K, E_K, \bar{g}_{Na}, E_{Na}, \bar{g}_L, E_L$ and the function parameters $\alpha_n, \alpha_m, \alpha_h$ and $\beta_n, \beta_m, \beta_h$. This is often called modeling the dynamics of a neuron. (Dynamical systems are often modeled by differential equations.)

The parameters and function parameters are discussed and given in the 1952 Hodgkin Huxley paper and were determined experimentally with the use of Boltzman's laws. Neuroscientists modify these parameters and function parameters to fit the neuron they are studying. In this study the function parameters given by Hodgkin and Huxley will be used. They will be introduced in Part II and analyzed there. In Part I these function parameters are used to generate the figures presented, but the analysis presented is true for any smooth function parameters.

If the reader is interested in a detailed derivation of the System 2.1, Friedman, *et al* [5] or Izhikevitch [6] are good texts to read. Even though the System 2.1 was developed in 1952 with technology that provided few details of the composition of the neuron, they are surprisingly useful and still studied by many mathematical neuroscientists to gain a better understanding of the dynamics of neurons. Hodgkin and Huxley considered both the analytics of these ODEs and produced numerical results from these ODEs. Some of the results presented in the work Hodgkin and Huxley knew and others they did not. Mathematical neuroscientists are developing analytical techniques for neurons with the aid of numerical results and neuroscientists learn more about neurons by analyzing and observing the plots and animations developed from numerical results.

Mathematical neuroscientists are still discovering more about System 2.1 and in turn neurons and vice versa. These equations have been modified many times to study neurons other than that of the giant squid. In this study, we will look at System 2.1 in honor of these two mathematical physiologists, but note that everything we present can be generalized to handle similar type neuron models. In fact, in Part II we address modifying the HH ODEs and using partial differential equations to model neurons.

We rewrite System 2.1 (suppressing t) as

$$\begin{aligned}
V' &= -\frac{1}{C_M}\left(\bar{g}_K n^4 + \bar{g}_{Na} m^3 h + \bar{g}_L\right) V + \\
&\qquad \frac{1}{C_M}\left(\bar{g}_K E_K n^4 + \bar{g}_{Na} E_{Na} m^3 h + \bar{g}_L E_L\right) + \frac{I_M}{C_M} \\
n' &= -\left(\alpha_n(V) + \beta_n(V)\right) n + \alpha_n(V) \qquad (2.2) \\
m' &= -\left(\alpha_m(V) + \beta_m(V)\right) m + \alpha_m(V) \\
h' &= -\left(\alpha_h(V) + \beta_h(V)\right) h + \alpha_h(V)
\end{aligned}$$

in order to consider properties and solution of these ODEs which we will also refer to as the HH ODEs. In determining properties and solutions of the HH ODEs, it will be necessary to generate approximate solutions using a computer. In this study we will use Matlab to generate the numerical approximations. We will call the solutions we generate in Matlab numerical solutions. Numerical solutions were obtained in Matlab using the Runge-Kutta 4 algorithm (RK4), the ODE45 algorithm of Matlab and the power series method (PSM) on System 2.2. These are three common numerical methods used to obtain numerical solutions to ordinary differential equations. PSM is an extension of the power series method that is introduced for linear differential equations in most elementary differential equations texts (Peterson, Sochacki [4]). We will discuss PSM more in Part II, but point out that it generates highly accurate solutions by determining Maclaurin polynomials for the solution to the degree desired by the user. PSM with Maclaurin polynomials of degree 8 were used to generate the numerical solutions in Part I. These results are in agreement with RK4 and ODE45.

Notice that each individual ODE in the System 2.2 has a similar form of $x' = ()x + ()$, where $()$ represents an expression not containing x. These ODEs are fundamental differential equations in both the autonomous and non-autonomous cases.

In their 1952 paper, Hodgkin and Huxley discuss the resting voltage state often. In mathematics this would be when the derivative is 0, i.e. V has a constant value. In this study, this will be called an equilibrium solution or equilibrium. If $V(t)$ is constant in the HH ODEs then $V'(t) = 0$. Let V_{rest} be this constant value. If $I_M(t) = 0$ in the V ODE of System 2.2 then the ODE for V becomes

$$V' = -\frac{\bar{g}_K n(t)^4 + \bar{g}_{Na} m(t)^3 h(t) + \bar{g}_L}{C_M} V + \frac{\bar{g}_K E_K n(t)^4 + \bar{g}_{Na} E_{Na} m(t)^3 h(t) + \bar{g}_L E_L}{C_M}$$

If $V(t) = V_{rest}$ so that $V'(t) = 0$ then from this ODE

$$V_{rest} = \frac{\bar{g}_K \; n(t)^4 E_K + \bar{g}_{Na} \; m(t)^3 h(t) E_{Na} + \bar{g}_L E_L}{\bar{g}_K \; n(t)^4 + \bar{g}_{Na} \; m(t)^3 h(t) + \bar{g}_L}.$$

Since $V(t) = V_{rest}$ for all t, the ODEs for n, m, h (in this case) are

$$\begin{aligned} n' &= -\left(\alpha_n(V_{rest}) + \beta_n(V_{rest})\right) n + \alpha_n(V_{rest}) \\ m' &= -\left(\alpha_m(V_{rest}) + \beta_m(V_{rest})\right) m + \alpha_m(V_{rest}) \\ h' &= -\left(\alpha_h(V_{rest}) + \beta_h(V_{rest})\right) h + \alpha_h(V_{rest}). \end{aligned}$$

Since these ODEs are linear, they can be solved. In fact, if $V(t) = V_{rest}$ then n, m, h are given by

$$\begin{aligned} n &= n_0 \; e^{-(\alpha_n(V_{rest})+\beta_n(V_{rest}))t} \; + \frac{\alpha_n(V_{rest})}{\alpha_n(V_{rest}) + \beta_n(V_{rest})} \\ m &= m_0 \; e^{-(\alpha_m(V_{rest})+\beta_m(V_{rest}))t} \; + \frac{\alpha_m(V_{rest})}{\alpha_m(V_{rest}) + \beta_m(V_{rest})} \\ h &= h_0 \; e^{-(\alpha_h(V_{rest})+\beta_h(V_{rest}))t} \; + \frac{\alpha_h(V_{rest})}{\alpha_h(V_{rest}) + \beta_h(V_{rest})}. \end{aligned}$$

Therefore, n, m, h decrease to an equilibrium state. That is, if $V \longrightarrow V_{rest}$ so that $V'(t) \approx 0$ for all $t > T$ for some $T > 0$ then

$$n, m, h \longrightarrow \frac{\alpha_n(V_{rest})}{\alpha_n(V_{rest}+\beta_n(V_{rest})}, \frac{\alpha_m(V_{rest})}{\alpha_m(V_{rest})+\beta_m(V_{rest})}, \frac{\alpha_h(V_{rest})}{\alpha_h(V_{rest})+\beta_h(V_{rest})},$$

respectively as t gets large ($t >> T$). Notice that all three of these limits have values greater than 0 and less than 1. Since n, m, h are concentrations, their values should always be greater than or equal to 0 and less than or equal to 1.

Observe that if n_E, m_E, h_E are each close to 0 then $V_{rest} \approx E_L$. If we let

$$n_E = \frac{\alpha_n(V_{rest})}{\alpha_n(V_{rest} + \beta_n(V_{rest})}$$
$$m_E = \frac{\alpha_m(V_{rest})}{\alpha_m(V_{rest}) + \beta_m(V_{rest})}$$
$$h_E = \frac{\alpha_h(V_{rest})}{\alpha_h(V_{rest}) + \beta_h(V_{rest})}.$$

be the equilibrium values for n, m, h, respectively then

$$V_{rest} = \frac{\bar{g}_K\ n_E^4\ E_K + \bar{g}_{Na}\ m_E^3\ h_E E_{Na} + \bar{g}_L E_L}{\bar{g}_K\ n_E^4 + \bar{g}_{Na}\ m_E^3\ h_E + \bar{g}_L}.$$

If $(V(t), n(t), m(t), h(t)) \longrightarrow (V_{rest}, n_E, m_E, h_E)$ then we call the equilibrium stable. Knowing these equilibrium results and corresponding equilibrium solutions allows one to test the accuracy (and consistency) of his or her algorithms on the HH ODEs. The software package Maple was used to obtain the numerical values given in this paper and these values will be demonstrated in the figures generated by numerical methods using Matlab on the HH ODES and in tables in Part II. It is also important to remember that all four equations on the right hand side of System 2.2 (or any system of ODEs) must be 0 at the equilibria. (We had Maple solve for these four equations - which it was able to do. However, the solutions were quite long. Therefore, numerical approximations to the solution were obtained.) The equilibrium values shown were tested in each of the four equations on the right hand side of the HH ODEs to ensure their accuracy.

The voltage potential of a neuron going from a minimum to a maximum and then to the resting state is often referred to as an *action potential*. We now present four action potential plots and figures showing the n, m, h concentrations for these four action potentials for numerical solutions using PSM in Matlab using the data from the 1952 Hodgkin Huxley paper with different $I_M(t)$ which will be referred to as input impulses. These are the types of plots (figures) that neuroscientists use to understand neurons and their properties. (The reader may think of other plots that would also be beneficial.) One notices in the voltage vs time plots (action potential) that the voltage reaches a maximum and then lowers to an equilibrium as mentioned above. (The reader should be able to notice the equilibrium state(s) from the figures.) This voltage can be observed (and recorded) in an experiment. This dynamic of reaching a maximum is often referred to as the *firing* of the neuron. The models presented will show four different types of firing. In each model we will also show the n, m, h concentration levels together and the concentrations as a three-dimensional curve. These figures (plots) show how the concentrations are related to each other and how the concentrations are related to the voltage. These figures were generated in Matlab. The initial conditions in all four models are$V(0) = V_0 = -30, n(0) = n_0 = 0.25, m(0) = m_0 = 0.25, h(0) = h_0 = 0.5$ and can be seen in each of the figures presented. The figures presented should be able to convince you that the dynamics of neurons is not only complicated, but exciting, since each of us has over eighty billion of them.

2.1 Model 1

Figure 3 is the action potential from the data given in the 1952 Hodgkin-Huxley paper and is similar to the one shown there. Notice that the starting voltage is -30 mV and the firing reaches a maximum near 110 mV at around 8 ms and then drops to an equilibrium. The simulation is for 80 ms so that one can observe the equilibrium solution derived above. This equilibrium value is 0.0036 mV for the voltage and will be discussed more in Part II. Figure 4 has two plots. The plot on the left shows the concentration levels of sodium (n), potassium (m) and the other ions (h) making up the 'leaky' concentration during the action potential. Notice they all tend to an equilibrium as the theory above showed. These equilibrium values will be given in Part II. The plot on the right is a three dimensional (3D) curve (n, m, h) of the concentrations showing how they are interrelated during the action potential. BEG denotes the initial values (n_0, m_0, h_0) and END denotes the equilibrium point (n_E, m_E, h_E) where the simulation time ends. Figure 5 has three plots showing how the voltage changes with each concentration. One should observe the cyclical nature and that the voltage makes large changes over short changes in the concentration levels. The plots shown in these figures for the action potential are typical in the neuroscience literature. Of course, other types of plots are also studied by mathematicians and neuroscientists.

As an example of what PSM provides, the 8^{th} degree Maclaurin polynomial PSM produced in Matlab for V is

$$\begin{aligned} V_8(t) = &- 30 + 150.65\,t - 4386.6\,t^2 + 107,130\,t^3 - 2,373,500\,t^4 + \\ &(5.2470 \times 10^7)t^5 - (1.1961 \times 10^9)t^6 + (2.8064 \times 10^{10})t^7 - (6.7184 \times 10^{11})t^8. \end{aligned}$$

From the remainder theorem for Maclaurin polynomials, the error in approximating V with V_8 is some number times t^9. Therefore, the magnitude of the coefficients in this polynomial increasing with the power of t and especially the magnitued of t^8 indicates t must be chosen small so that the error is not large. This was also verified in all three numerical methods used in this study. To get the Maclaurin polynomial of degree 8 for the next small time interval, a value for t is substituted into V_8 to get the initial value for V for the next time interval and so on. After examining V_8, the reader can appreciate having a tool like Matlab to generate the Maclaurin polynomials. This polynomial also helps explain why a small time step (2^{-8}) had to be chosen for the RK4 method and a low error tolerance (10^{-13}) had to be chosen for ODE45 in Matlab since the error of these two methods can be computed from Taylor polynomial estimates. Maclaurin polynomials will be discussed further in Part II.

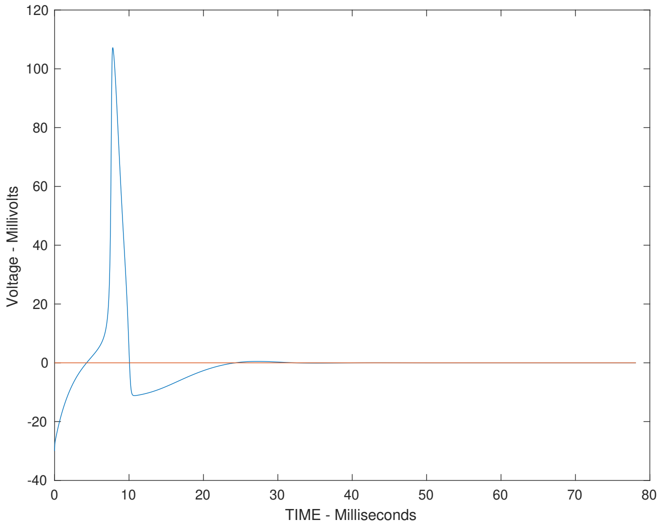

Figure 3: Voltage vs Time Plot Showing an Action Potential.

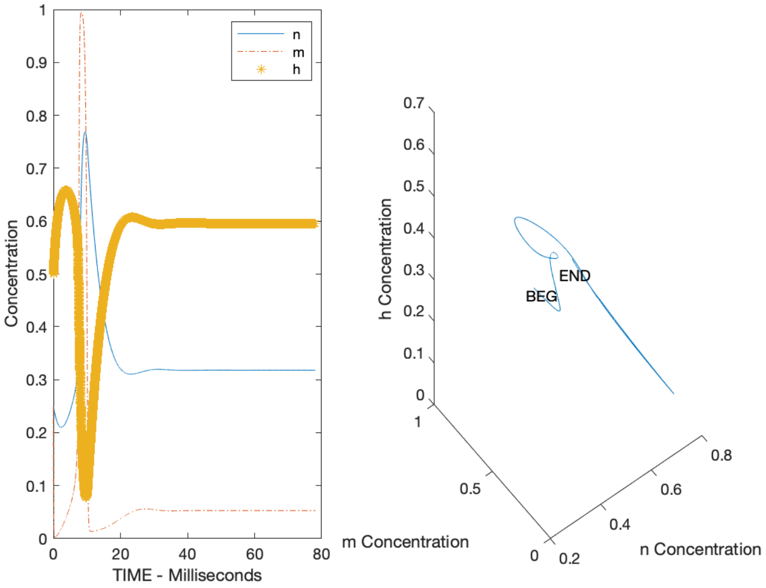

Figure 4: Plots of Concentration Levels for n, m, h.

2.2 Model 2

Figure 6 is the excitation using the same data as in Model 1 together with the input impulse $I_M(t) = 10\exp(-0.125(t-50)^2)$. The plot on the left shows two action potentials and the plot on the right shows the second action potential occurs near where $I_M(t)$ reaches its maximum which is 10 at 50 ms. ($I_M(t)$ is shown with the action potential on the right.) This simulation is also for 80 ms so that one can observe the equilibrium solutions derived

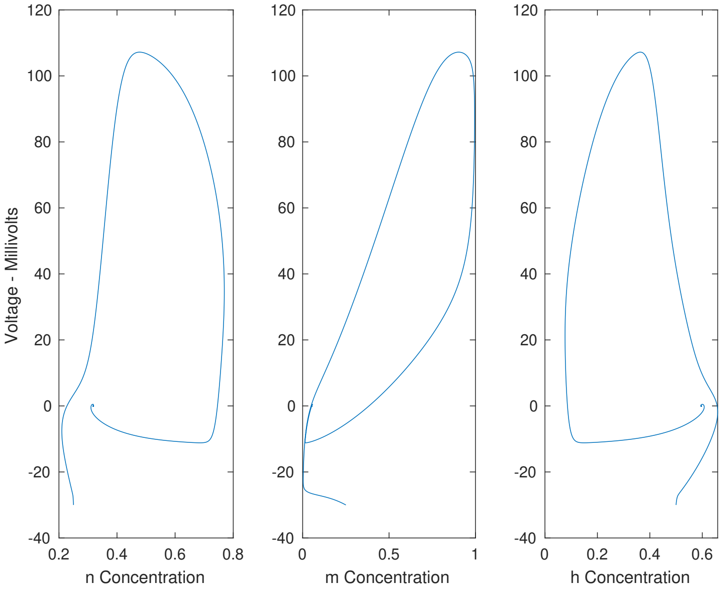

Figure 5: Plots of Concentration vs Voltage.

above still occur. This equilibrium value is also 0.0036 mV for the voltage. The plots in Figure 7 show that the concentration levels of sodium (n), potassium (m) and the other ions (h) making up the 'leaky' concentration are quite similar for the initial firing and the second firing due to the input impulse. Notice all the concentrations again tend to an equilibrium. The 3D curve (n, m, h) of the concentrations shows the two action potentials. In Figure 8 one should observe the double cyclical nature and that the voltages again make large changes over short changes in the concentration levels.

2.3 Model 3

Figure 9 is the excitation with the data in Model 1 and an input impulse $I_M(t) = 10\sin(0.125t)$. One observes three firings and that the second firing occurs after the input becomes negative and the third firing occurs quickly after the second firing. Note also that the maximum voltage of each firing is different. The plots in Figure 10 show more cyclical behavior including the 3D curve. In Figure 11 the cyclical voltage vs the concentration levels is noticeable.

2.4 Model 4

Figure 12 is the repeating firings due to an input impulse with higher frequency. The input impulse is $I_M(t) = 10\sin(0.5t)$. Notice that there is an action potential shortly after the input hits a maximum value and that this phenomenon repeats. In Figure 13 we see the repeating process in the oscillations of the concentration levels and in the cylic orbit of the 3D curve. The concentration vs voltage curves in Figure 14 approach a periodic orbit. This phenomenon of approaching a periodic orbit is referred to as a limit cycle.

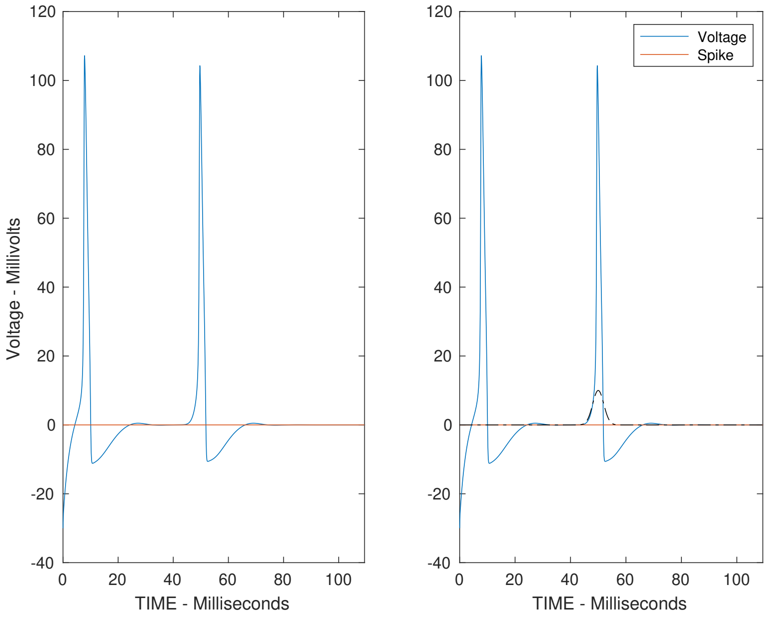

Figure 6: Voltage with a Spike Impulse Included.

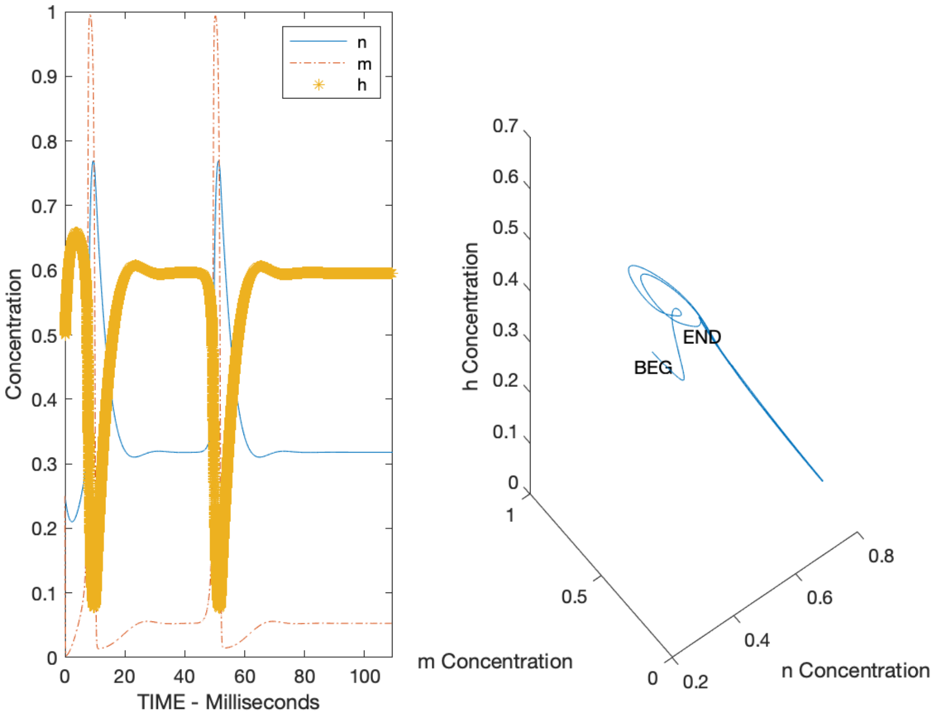

Figure 7: *n*, *m*, *h* Concentrations with a Spike Impulse Included.

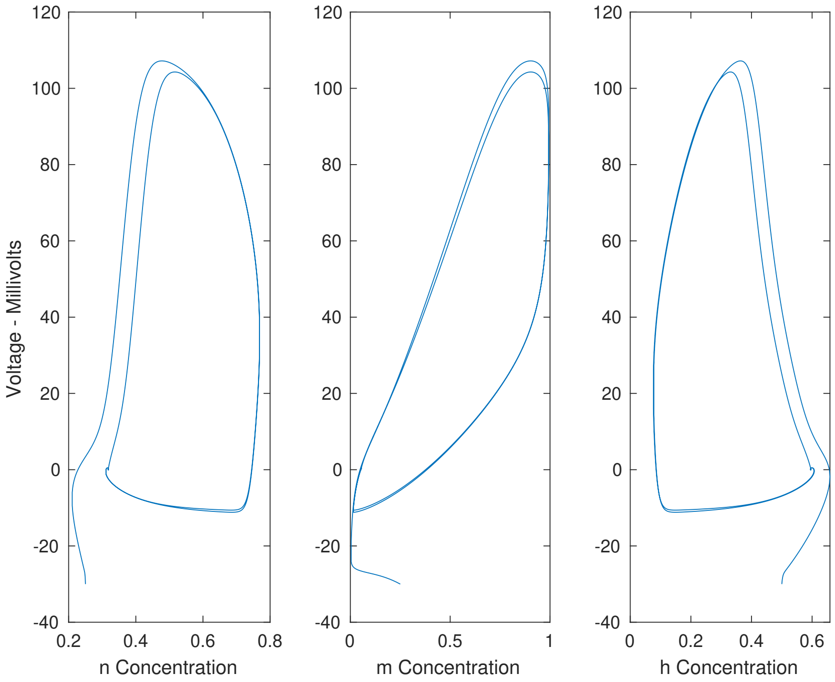

Figure 8: Plots of Concentration vs Voltage with a Spike Impulse Included.

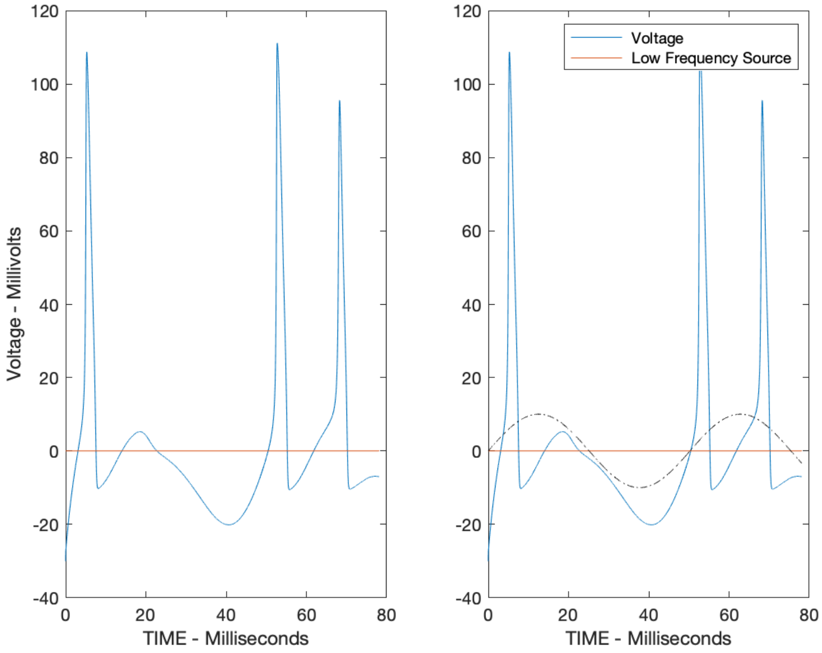

Figure 9: Voltage Plot with a Low Frequency Impulse Included.

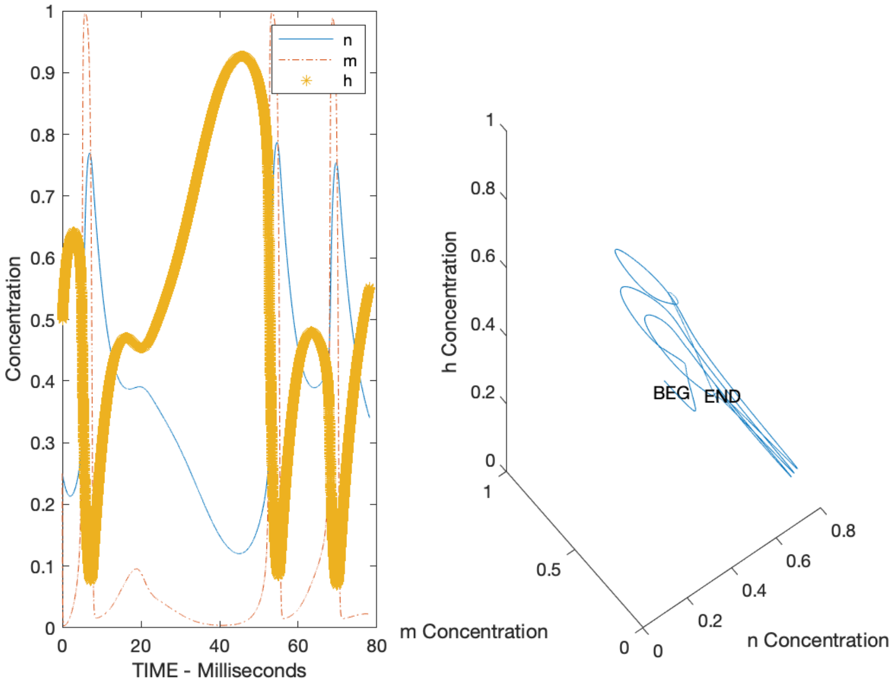

Figure 10: n, m, h Concentration Curves with a Low Frequency Impulse Included.

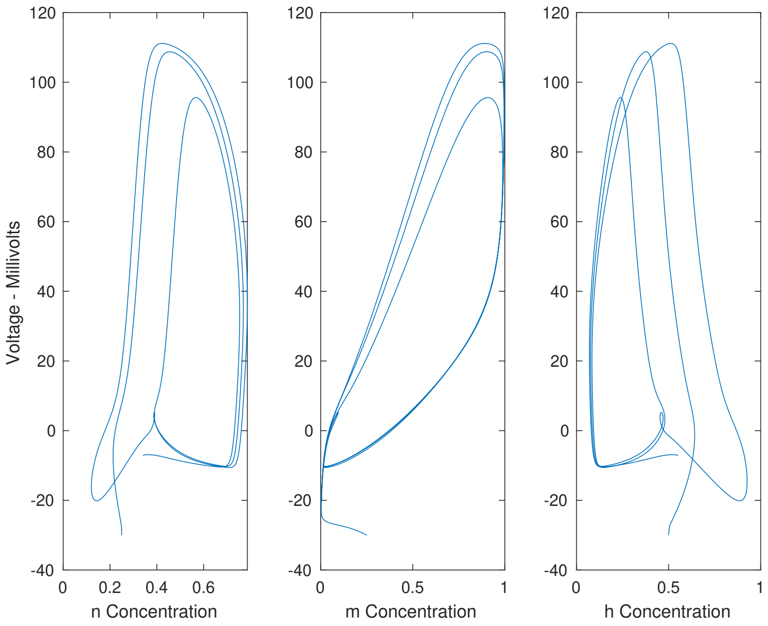

Figure 11: Plots of Concentration vs Voltage with a Low Frequency Impulse Included.

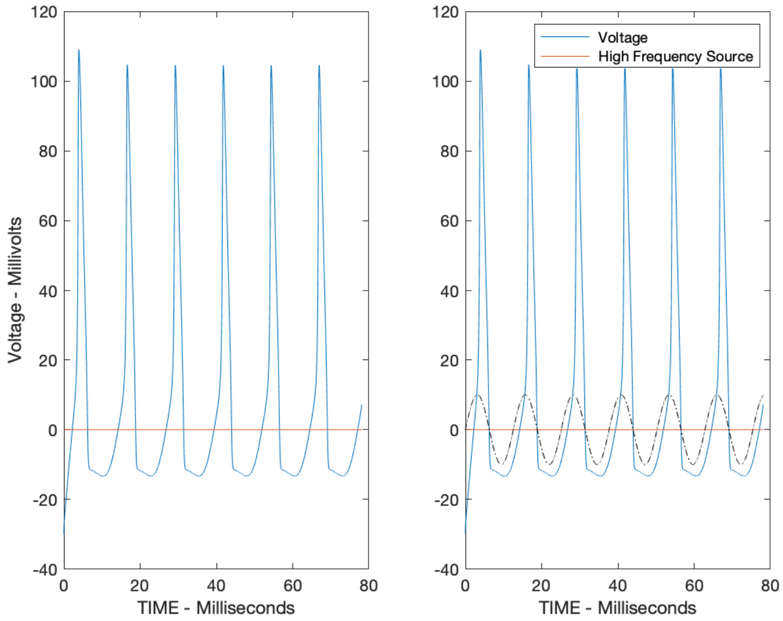

Figure 12: Voltage Plot with a High Frequency Impulse Included.

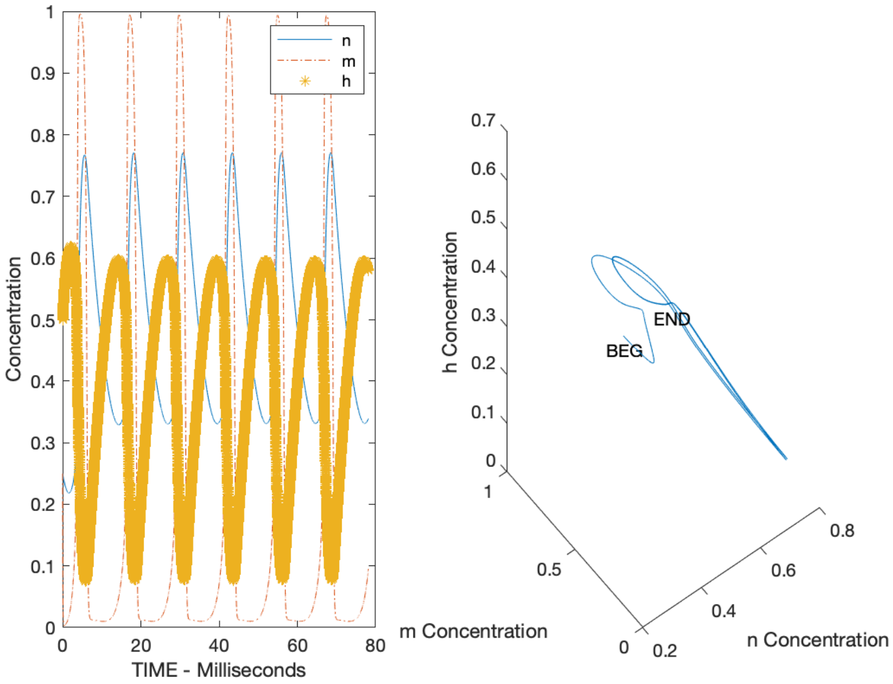

Figure 13: n, m, h Concentration Curves with a High Frequency Impulse Included.

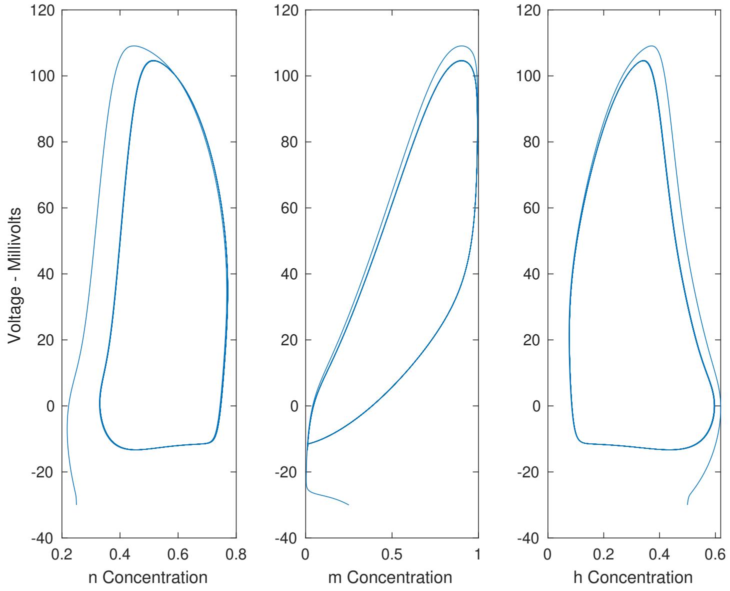

Figure 14: Plots of Concentration vs Voltage with a High Frequency Impulse Included.

Those who want to run their own HH models for various initial conditions, parameters and inputs to generate figures like the ones presented here, can go to

`http://educ.jmu.edu/~sochacjs/NeuronMatlabCodes/PSM`

There the reader will find the two Matlab files HHPSMparameters.m and HHPSMODEsolver.m and the pdf file HHPSMinstructions.pdf that explains these two Matlab files and the routines they need to run HHPSMODEsolver. The reader can then download these files and set up various models to run and plot in Matlab. (The RK4 codes are at `http://educ.jmu.edu/~sochacjs/NeuronMatlabCodes/RK4` and the ODE45 codes are at `http://educ.jmu.edu/~sochacjs/NeuronMatlabCodes/ODE45`. The instructions for these codes are HHRK4instructions.pdf and HHODE45instructions.pdf, respectively.)

The site `http://educ.jmu.edu/~sochacjs/PSM.html` has access to many papers (published) showing the properties and applicability of PSM. Some of these papers were written by undergraduates. The website `http://www.taylorcenter.org/Index.htm` also has a lot of information on using power series to solve ordinary differential equations.

3 Part II. Extending the Hodgkin-Huxley Model

Hopefully, the reader noticed in Part I that many of the graphs in the plots presented had complicated features and would be difficult to discover without plotting the numerical solution. Also, one should realize showing that the numerical solution is an accurate representation of the true solution is important. Researchers have shown that PSM is an accurate method and has an *a priori* error bound. That is, one can determine what the maximum possible error can be from using PSM to generate a numerical approximation to the true solution before calculating the approximate solution (not including errors brought in by the computer). (See Warne *et al* [7].)

In 1989, Parker and Sochacki [8] showed that using Picard iteration on ordinary differential equations for which the expressions on the right hand side of the ordinary differential equations are polynomial in the variables to be solved generates the Maclaurin polynomials for the solution to the ordinary differential equations. They also showed how to convert ordinary differential equations like the HH ODEs into a system of ordinary differential equations with polynomial right hand side so that Picard iteration can be used to generate the Maclaurin polynomials for the solution to the system of ordinary differential equations. This will be demonstrated in this part of the paper.

In 2009, Stewart and Bair [9] named the technique Parker and Sochacki developed in 1989, the Parker-Sochacki method and introduced it as an approach for obtaining accurate numerical solutions to the HH ODEs. They showed how to convert the HH ODEs that they were using to polynomial form. They also showed that the Parker-Sochacki method was competitive with the Runge-Kutta order 4 method (RK4) and the Bulirsch-Stoer method (BSM) and, in most cases, superior for Hodgkin-Huxley type ODEs. Stewart and Bair also pointed out the removable singularities in the function parameters in the HH ODEs. In 2011, Stewart and Gurney [10] demonstrated the feasibility of using the Parker-Sochacki method to study synaptic dynamics in neurons.

Automatic differentiation (AD) (Gofen [11] and Neidinger [12]), the differential transform method (DTM) (Mirzaee [13]) and PSM are methods that are similar to the Parker-Sochacki method. In this article, we will use the terminology power series method (PSM) for the method that was discovered by Parker and Sochacki through Picard iteration, but is similar to AD and DTM. We also mention that since the late 1950's many researchers have used PSM type methods to obtain solutions to ODEs. However, as far as this author knows, Parker and Sochacki are the only ones to do it through Picard iteration.

In fact, one can consider the above techniques as complimentary and intersecting. However, since the Parker-Sochacki method is also based on the Picard Iteration, this method allows another tool to be used in analyzing the numerics through convergence, stability and error estimation. In this presentation, we will show how to use PSM to address the removable singularity in α_n and α_m in the original HH ODEs in four ways.

It will also be shown that PSM extends to the one dimensional (1D), two dimensional (2D) and 3D HH partial differential equations (PDEs) in a straightforward manner. Numerical results for these three cases will be presented through figures demonstrating the feasibility of PSM to PDEs.

Since many of the previous researchers mentioned above, and, in fact many other researchers, have shown the efficiency and robustness of PSM, this paper will not focus on those subjects, but rather on dealing with the removable singularities and the sensitive dependence for the HH ODEs (and PDEs). This author is unaware of this study being done in the neuron literature.

If the reader is interested in some applications and theory of PSM, Pruett, Rudmin and Lacy [14] showed that PSM was in many cases superior to RK4 and BSM for Newton's N-Body problem. Also, the astrophysics teams Nurminskii and Buryi [15] and Pruett, Ingham and Herman [16] and the neuroscientist Szynkiewicz [17] and the neuroscientist team Yudanov, Shaaban, Melton and Reznik [18] have shown that it is in general easier to parallelize PSM codes and obtain close to linear speed up than other methods, including on graphical processing units.

In his Ph.D. thesis, Money [19] shows that when applied discretely to PDEs, PSM is a generalization of the Lax-Wendroff method (LWM) [20] that is commonly used and/or modified in computational fluid dynamical systems that involve nonlinearities. Money determined stability conditions for some linear and non-linear PDEs. In this work, his ideas will be extended and applied to HH PDEs.

The outline of the study for Part II is as follows. The original HH ODEs function parameters $\alpha_n, \beta_n, \alpha_m, \beta_m, \alpha_h, \beta_h$ are introduced and the singularity is discussed. Four ways to work around the singularity are suggested. These are then studied to demonstrate how sensitive to parameters the HH ODEs are. Next, the 1D cable PDE (Ermentrout and Terman) is given and solved using the discrete PSM (DPSM) introduced by Money. This presentation is then extended to 2D and 3D in a natural way. Figures showing the results and the sensitive dependence are given for both the ODEs and the PDEs developed. These results are presented with PSM, but the results shown are in agreement with RK4 and ODE45 for the ODEs using the same grid size in t. Figures for the 1D, 2D and 3D cases verifying the legitimacy of the results and the efficacy of the HH PDEs are presented using PSM. The methods developed and the figures presented demonstrate the sensitive nature of neurons and the need to have several numerical methods for analyzing these dynamical systems.

As Sochacki [21] pointed out, transforming a nonlinear ODE to polynomial form allows many other types of analysis. Since an ODE or PDE in polynomial form can be put into a symbolic computing environment like Mathematica or Maple, one can generate a completely symbolic (Maclaurin) polynomial that approximates the solution to a system of ODEs or PDEs and analyze this solution for sensitivity in any of the parameters, including the initial conditions. Also, putting the HH ODEs in polynomial form allows one to derive a priori error estimates as shown by Warne, *et al* [7] and extended by Thelwell, *et al* [22]. Furthermore, the polynomial form of the ODEs allows one to develop conserved quantities as shown by Carothers, *et al* [23]. They also show how to extend PSM to Pade' rational approximations allowing another numerical method to be used for neuron simulation.

The parameter values Hodgkin and Huxley (HH) used in their 1952 paper and the values given in Izhikevich (Iz) [6] will be used in the numerical study done here. These are $g_K = 36, g_{Na} = 120, g_L = 0.3, C_M = 1; E_K = 12; E_{Na} = -115, E_L = -10.613$ and $g_K = 36, g_{Na} = 120, g_L = 0.3, C_M = 1; E_K = -12; E_{Na} = 120, E_L = 10.6$, respectively. For HH, the equilibrium solutions are $V_{rest} \approx 0.0036, n_E \approx 0.3177, m_E \approx 0.053, h_E \approx 0.596$ and for Izhikevich, they are $V_{rest} \approx 0.0462, n_E \approx 0.3184, m_E \approx 0.0532, h_E \approx 0.5945$. The equilibrium values for the HH data were demonstrated in the figures given in Part I. Even though the HH parameters and the Iz parameters are nearly the same, the reader will note that the outcomes can be different. One can even see that in the equilibrium values where all are almost the same, except for V_{rest}.

3.1 The Removable Singularities

Hodgkin and Huxley experimentally determined $\alpha_n, \beta_n, \alpha_m, \beta_m, \alpha_h, \beta_h$ for a particular squid and presented them in their paper as

$$\begin{aligned}
\alpha_n &= \frac{0.01(10-V)}{\exp\left(\frac{10-V}{10}\right)-1}\ ; & \beta_n &= 0.125\exp\left(\frac{-V}{80}\right)\\
\alpha_m &= \frac{0.1(25-V)}{\exp\left(\frac{25-V}{10}\right)-1}\ ; & \beta_m &= 4\exp\left(\frac{-V}{18}\right)\\
\alpha_h &= 0.07\exp\left(\frac{-V}{20}\right)\ ; & \beta_h &= \frac{1}{\exp\left(\frac{30-V}{10}\right)+1}.
\end{aligned}$$

Note that

$$\alpha_n = 0.1\,\frac{x}{\exp(x)-1}$$

where $x = 0.1(10-V)$ and

$$\alpha_m = \frac{y}{\exp(y)-1}$$

where $y = 0.1(25-V)$ and that these two functions are not defined at $x = 0$ ($V = 10$) and $y = 0$ ($V = 25$), respectively. (We leave it to the reader to show using L'Hospital's rule that the two functions can be defined at 0, by using the limit value.)

Certainly, $V = 10$ and $V = 25$ are valid voltages for the squid neuron. What does one do in this case? Calculating the limits for α_n and α_m at $V = 10$ and $V = 25$, respectively, gives 0.1 for α_n and 1 for α_m. Therefore, these two functions have removable singularities at $V = 10$ and $V = 25$, respectively. No matter what numerical method one uses, one has to deal with this removable singularity in some manner.

One method for doing this with PSM is presented. Since the two functions α_n and α_m are similar, we consider only α_n. (The method presented is easily extended for α_m). The derivative of α_n with respect to t using the chain rule is

$$\alpha_n' = 0.1\frac{x'(e^x-1)-xx'e^x}{(e^x-1)^2} = 0.1x'\frac{e^x(1-x)-1}{(e^x-1)^2}.$$

Since e^x is not a polynomial, we make some change of variables and use the chain rule. First let $\alpha_{n,1} = (e^x-1)^{-1}$ and $\alpha_{n,2} = e^x$. Differentiating these with respect to t gives us

$$\alpha_{n,1}' = -(e^x-1)^{-2}\,e^x\,x' = -{\alpha_{n,1}}^2\,\alpha_{n,2}\,x'$$

and

$$\alpha_{n,2}' = e^x x' = \alpha_{n,2}x'.$$

The right hand sides of these two ODEs are polynomial in the unkowns. Now note that $\alpha_n = 0.1x\,\alpha_{n,1}$. Differentiating α_n in this form gives

$$\alpha_n' = 0.1(x\alpha_{n,1})' = 0.1(x'\alpha_{n,1} + x\alpha_{n,1}').$$

The right hand side of this ODE is now polynomial in the unknowns $x, x', \alpha_{n,1}$ and $\alpha_{n,2}$. Since these are all polynomial, by introducing the variables $\alpha_{n,1}$ and $\alpha_{n,2}$, an ODE for α_n with a polynomial expression in the unknowns on the right hand side has been derived. Therefore, as long as $x \neq 0$ and $y \neq 0$, one can create polynomial ODEs for α_n and α_m. Fehlberg [24] of Runge-Kutta-Fehlberg fame called the functions that were introduced to make the ODE polynomial: *auxiliary variables.*

Now consider $\beta_h = (\exp\left(\frac{30-V}{10}\right) + 1)^{-1}$. Introducing the auxiliary variables $z = 0.1(30 - V)$ and $\beta_{h,1} = e^z$ leads to

$$\beta_h' = -{\beta_h}^2 \beta_{h,1} z'.$$

Except for the singularities at $x = 0$ in α_n and at $y = 0$ in α_m it has been demonstrated that the HH ODEs can be converted to a system of polynomial ODEs which can be solved with PSM. It is important to remember that now $\alpha_n, \beta_n, \alpha_m, \beta_m, \alpha_h, \beta_h$ are considered as functions of t instead of V through the fact that V is a function of t.

The polynomial form of the HH ODES appear as

$$\begin{aligned}
V' &= -\frac{1}{C_M}\left(\bar{g}_K n^4 + \bar{g}_{Na} m^3 h + \bar{g}_L\right) V + \\
&\quad \frac{1}{C_M}\left(\bar{g}_K E_K n^4 + \bar{g}_{Na} E_{Na} m^3 h + \bar{g}_L E_L\right) + \frac{I_M}{C_M} \\
n' &= -\left(\alpha_n + \beta_n\right) n + \alpha_n \\
m' &= -\left(\alpha_m + \beta_m\right) m + \alpha_m \\
h' &= -\left(\alpha_h + \beta_h\right) h + \alpha_h,
\end{aligned}$$

together with polynomial ordinary differential equations for $\alpha_n, \beta_n, \alpha_m, \beta_m, \alpha_h, \beta_h$. For example, we showed that a polynomial ordinary differential equation for α_n is

$$\alpha_n' = 0.1(x'\alpha_{n,1} - xx'{\alpha_{n,1}}^2\,\alpha_{n,2}),$$

where $x = 0.1(10 - V)$ so that $x' = -0.1V'$ and V' is given in the first ODE in the system in polynomial form. (The functions $\alpha_{n,1}$ and $\alpha_{n,2}$ and their ODEs in polynomial form were given above.) We leave it for the reader to generate polynomial ODEs for $\beta_n, \alpha_m, \beta_m$ and α_h through auxiliary variables. This is the type of polynomial system that is used with PSM to generate solutions to the HH ODEs in this paper. (For RK4 and ODE45 we used the first form of the HH ODEs presented in Part I.) If the reader wants to gain a thorough understanding of forming polynomial ODEs, the first link on the site http://educ.jmu.edu/~sochacjs/PSM.html is a good tutorial.

We will also demonstrate four ways to work around the singularities in α_n and α_m so that PSM can be used. In our numerical examples, an 8^{th} degree Maclaurin polynomial to approximate the solution is used, but any degree desired can be used. We denote the method by PSM8 to stress the 8^{th} degree.

As Stewart and Bair discovered, PSM will determine the time at which the singularities occur to the precision of the order (degree of the (Maclaurin) polynomial) of PSM used. Since the HH ODEs can be cast in polynomial form, a symbolic form of the approximate polynomial solution to any degree desired can be obtained in a symbolic computational software environment like Mathematica or Maple. This polynomial can be analyzed for sensitivity in any of the parameters, initial conditions or inputs. This will not be done in this paper. Here we will focus on sensitivity dependence using PSM8.

3.2 Three Approximations for α_n and α_m

First, we present three functions that will be used in this analysis to approximate the original HH α_n (α_m will be approximated by these same three functions, but with different parameters). These three functions will have similar properties to the original, but will have no singularities in V. Our first function is

$$\alpha_{n_2} = a\ln(\exp(cx) + r) + bx,$$

with $a = 0.1414908967, b = 0.009940471319, c = -0.07023657394, r = 0.5088042066$ (for α_{m_2} one has $a = 1.353627622, b = 0.09779785093, c = -0.07224256783, r = 0.1795806050$). We will denote the HH model with $\alpha_{n_2}, \alpha_{m_2}$ for α_n, α_m as BF for best fit because of our three approximations this one is the best fit to the original HH. The second function is

$$\alpha_{n_3} = 0.1\,(\ln(\exp(x) + 1) - x)$$

($\alpha_{m_3} = \ln(\exp(y) + 1) - y$). We will denote this approximation model as LN. The third function is

$$\alpha_{n_4} = p\exp(sx) + q$$

with $p = 0.06494755254, q = -0.006749881849, s = 0.02985000448$ (for α_{m_4} one has $p = .2352963135, q = -0.01173258887, s = 0.03947343893$). We will denote this approximation model as EXP.

In Figure 15 there are four plots. The top two plots are for $\alpha_n, \alpha_{n_2}, \alpha_{n_3}, \alpha_{n_4}$ and then for $\alpha_m, \alpha_{m_2}, \alpha_{m_3}, \alpha_{m_4}$ that will be examined in this paper. The α_n and α_m from the Hodgkin and Huxley paper are denoted with black asterisks, the approximation α_{n_2} and α_{m_2} are denoted with red plus signs, α_{n_3} and α_{m_3} are denoted with green circles and α_{n_4} and α_{m_4} are denoted with blue diamonds. Notice that α_{n_4} and α_{m_4} are near the other three for low voltages, but diverge from the other three for high voltages. The bottom two plots are the same curves without α_{n_4} and α_{m_4}. Notice how close these curves are over the voltage spread. These plots are over a typical mV range for the HH neuron. To be specific, the maximum difference between α_n and $\alpha_{n_2}, \alpha_{n_3}, \alpha_{n_4}$ is 0.0039, 0.0300, 2.9344 respectively and the maximum difference between α_m and $\alpha_{m_2}, \alpha_{m_3}, \alpha_{m_4}$ is 0.1325, 0.3068, 47.5965, respectively for $-30 \leq V \leq 120$. These values will be interesting to recall in the numerical results presented later. Note that for voltage in [-30,50] all the graphs are close.

Observe that none of $\alpha_{n_2}, \alpha_{n_3}, \alpha_{n_4}$ or $\alpha_{m_2}, \alpha_{m_3}, \alpha_{m_4}$ have singularities. Therefore, PSM can be used on all three of these models with these replacing α_n and α_m. One can then compare and contrast the results given by RK4, ODE45 and PSM which was done for the ODE cases presented below.

In Table 1 the equilibrium solutions for V, n, m and h are given for the HH parameters and in Table 2 for the Izhikevich parameters for the four α_n and α_m presented. Again, these equilibria will be highlighted in the numerical results presented.

We point out that even though $\alpha_{n_3}, \alpha_{m_3}$ are 'close' approximations to α_n, α_m, the values of V_{rest} are noticeably different for the two models. However, it is possible to determine

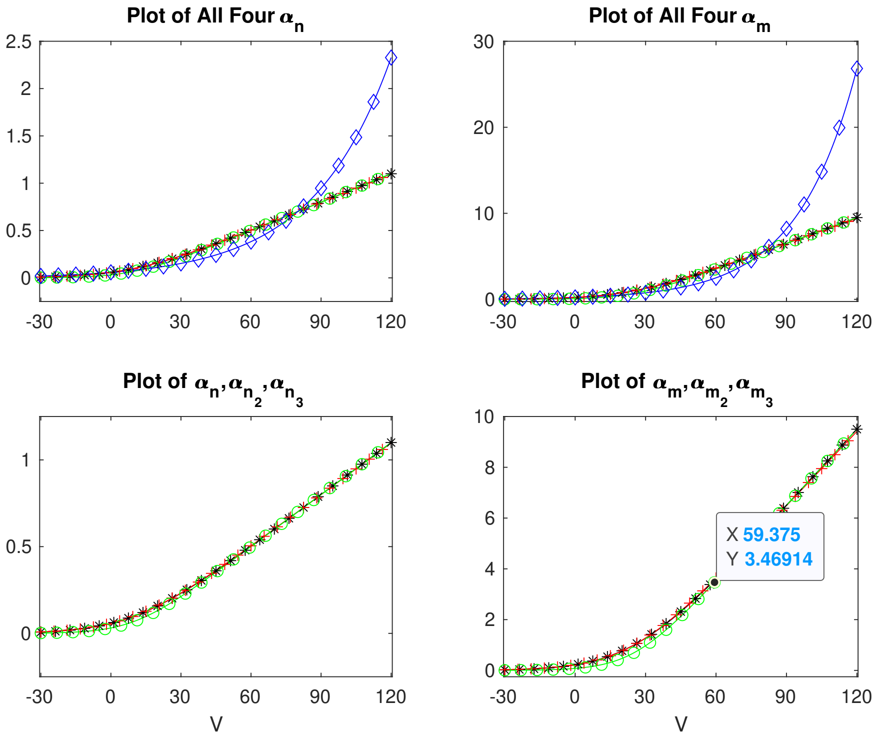

Figure 15: Plot of α_n and α_m for the HH,BF,LN and EXP models

Table 1: Original Hodgkin-Huxley Parameter Equilibria

α	V_{rest}	n_E	m_E	h_E
HH	0.003621	0.317732	0.052955	0.595994
BF	0.003617	0.317732	0.052955	0.595994
LN	3.317822	0.256558	0.031529	0.477551
EXP	0.004388	0.317720	0.052954	0.595967

Table 2: Izhikevich Hodgkin-Huxley Parameter Equilibria

α	V_{rest}	n_E	m_E	h_E
HH	0.046215	0.318385	0.053222	0.594504
BF	0.046151	0.318379	0.053216	0.594506
LN	3.322646	0.256645	0.477378	0.031551
EXP	0.055958	0.318233	0.053206	0.594162

HH voltage and conductance parameters so that the equilibria are all noticeably different. Even for these models, one will notice significant differences in the outcomes of the models as shown in the numerical section. First, we address the removable singularities in the original HH model.

3.3 Using Power Series (PS) to address the removable singularities

Stewart and Bair presented two ways to determine the PS of $\frac{1}{g}$ where g is a PS. One was through $z = \frac{f}{g}$ (let $f = 1$) and the other was through $z = g^r$ (let $r = -1$). Both of these give the same recurrence relation for determining the PS for $z = \frac{1}{g}$. We use this to determine the PS of $\frac{x}{e^x-1}$ $(\frac{y}{e^y-1})$ from the PS of $g = \frac{e^x-1}{x}$ $(g = \frac{e^y-1}{y})$.

We point out that the reciprocal of α_n (α_m) has a PS representation that converges for all x (y) and is given by

$$\frac{e^x - 1}{x} = 1 + \frac{1}{2}x + \frac{1}{3!}x^2 + \ldots = \sum_{n=0}^{\infty} \frac{x^n}{(n+1)!}$$

for $x \neq 0$. However, the PS is defined at $x = 0$. If we use the power series for g instead of g then there is no singularity. Therefore, using the reciprocal rule on the PS for g, we now have a PS for z. In order to obtain the PS rule for $z(V)$ where V is a polynomial, we layout how to determine the PS for the composition of two PS.

Let p_M be the M^{th} degree (Maclaurin) polynomial defined by

$$p_M = \sum_{m=0}^{M} p_{M,m} t^m = p_{M,0} + p_{M,1}\, t + p_{M,2}\, t^2 + \ldots + p_{M,M} t^M$$

and q_N be the N^{th} degree (Maclaurin) polynomial defined by

$$q_N = \sum_{n=0}^{N} q_{N,n} t^n = q_{N,0} + q_{N,1}\, t + q_{N,2}\, t^2 + \ldots + q_{N,N} t^N$$

then the (Maclaurin) polynomial for $p_M(q_N(t))$ can be obtained by letting

$$\begin{aligned} w_0 &= 1 \\ w_1 &= q_N \\ w_2 &= (q_N)^2 \\ &\vdots \\ w_M &= (q_N)^M \end{aligned}$$

and noting that

$$p_M(q_N) = \sum_{m=0}^{M} p_{M,m} w_m = p_{M,0} + p_{M,1}\, w_1 + p_{M,2}\, w_2 + \ldots + p_{M,M} w_M$$

and

$$\begin{aligned} w_0' &= 0 \\ w_1' &= w_0 \\ w_2' &= 2w_1 \\ &\vdots \\ w_M' &= M w_{M-1} \end{aligned}$$

Using PSM on this system of linear ODEs determines the coefficients of w_m for $m = 0..M$. That is,

$$w_m(t) = \sum_{j=0}^{J} w_{m,j}\, t^j,\ m = 0..M$$

and from this that

$$\begin{aligned}
p_M(q_N(t)) &= \sum_{m=0}^{M} p_{M,m} \left(\sum_{j=0}^{J} w_{m,j} t^j \right) \\
&= p_{M,0} + p_{M,1} \sum_{j=0}^{J} w_{1,j}\, t^j + p_{M,2} \sum_{j=0}^{J} w_{2,j}\, t^j + \dots + p_{M,M} \sum_{j=0}^{J} w_{M,j}\, t^j \\
&= \left(p_{M,0} + p_{M,1}w_{1,0} + \dots + p_{M,M}w_{M,0}\right) + t\left(p_{M,1}w_{1,1} + \dots + p_{M,M}w_{M,1}\right) \\
&+ t^2\left(p_{M,1}w_{1,2} + \dots + p_{M,M}w_{M,2}\right) + \dots + t^J\left(p_{M,1}w_{1,J} + \dots + p_{M,M}w_{M,J}\right) \\
&= \sum_{m=0}^{M} p_{M,m}w_{m,0} + \sum_{j=1}^{J} \left(\sum_{m=1}^{M} p_{M,m}w_{m,j} \right) t^j,
\end{aligned}$$

the (Maclaurin) polynomial for $p_M(q_N(t))$ of degree J. Therefore, using this algorithm, one can determine a polynomial approximation to $\alpha_n(V)$ ($\alpha_m(V)$) given a polynomial approximation to V and using the PS for $\alpha_n(x)$ ($\alpha_m(y)$).

Since α_n and α_m have singularities only near $x = 0$ and $y = 0$, respectively, we only use this algorithm when $|x| < 0.5$ ($0.1|V - 10)| < 0.5$) and $|y| < 0.5$ ($0.1|V - 25)| < 0.5$). Everywhere else PSM is used on the polynomial form of the HH ODEs. In these intervals of convergence of the PS for α_n and α_m, using the first 24 terms of the PS gives an error $< 10^{-19}$ for α_n and α_m. (It can be determined that eventually the PS for z is an alternating PS with only even exponents.) Outside these intervals, the polynomial form of the ODEs for α_n and α_m presented above are used. One can now determine polynomial solutions to the HH ODEs everywhere to the accuracy that Stewart and Bair outlined using these PS. In the numerical examples, we will compare using this PS technique for α_n and α_m compared to that of using the three approximations for α_n and α_m given above. In all the ODE and PDE examples below, we use PSM8 with the PS approximating α_n and α_m, using the first 24 terms of the PS.

We now add two other PS to the tool box of Stewart and Bair. Since we have to calculate the PS of $z = \exp(f)$ and $z = \ln(f)$ for a PS f for $\alpha_{n_2}, \alpha_{n_3}, \alpha_{n_4}$ and $\alpha_{m_2}, \alpha_{m_3}, \alpha_{m_4}$, we note

1. If $z = \exp(f)$ then $z' = f'z$
2. If $z = \ln(f)$ then $z' = \frac{f'}{f}$.

Using these two rules, allows one to generate a polynomial form of the ODEs for each of $\alpha_{n_2}, \alpha_{n_3}, \alpha_{n_4}$ and $\alpha_{m_2}, \alpha_{m_3}, \alpha_{m_4}$. We present numerical results of PSM8 below using these. (The quotient rule used by Stewart and Bair is needed with item 2.) Before that, though, we present a PSM algorithm for HH PDEs using the original α_n and α_m with the PS algorithm presented above to work around the singularities.

3.4 HH Partial Differential Equations

The HH cable PDE, nicely developed in Ermentrout and Terman [25], is given by

$$\begin{aligned}
C_M \frac{\partial V}{\partial t} &= A \frac{\partial^2 V}{\partial x^2} - (\bar{g}_K \, n(x,t)^4 \, (V - E_K) + \bar{g}_{Na} \, m(x,t)^3 h(x,t) \, (V - E_{Na}) \\
&\quad + \bar{g}_L \, (V - E_L)) + I_M(t) \\
\frac{d}{dt} n &= -(\alpha_n(V) + \beta_n(V)) \, n + \alpha_n(V) \\
\frac{d}{dt} m &= -(\alpha_m(V) + \beta_m(V)) \, m + \alpha_m(V) \\
\frac{d}{dt} h &= -(\alpha_h(V) + \beta_h(V)) \, h + \alpha_h(V)
\end{aligned}$$

and forms the basis for the present numerical study. The derivation in this text treats the complete neuron cell as shown in Figure 1 as an electrical cable with components. Here this PDE-ODE system will be used to develop an electrical system for multiple neurons. In order to do this and to generate meaningful numerics for a system of neurons, the finite difference form that Ermentrout and Terman use to get this PDE, which is

$$\begin{aligned}
C_M \frac{\partial V}{\partial t} &= F_+ \, (V(x + \Delta x) - V(x,t)) + F_-(V(x) - V(x - \Delta x, t)) \\
&\quad - (\bar{g}_K \, n(x,t)^4 \, (V - E_K) + \bar{g}_{Na} \, m(x,t)^3 h(x,t) \, (V - E_{Na}) \\
&\quad + \bar{g}_L \, (V(x,t) - E_L)) + I_M(x,t) \\
n'(x,t) &= -(\alpha_n(V(x,t)) + \beta_n(V(x,t))) \, n(x,t) + \alpha_n(V(x,t)) \\
m'(x,t) &= -(\alpha_m(V(x,t)) + \beta_m(V(x,t))) \, m(x,t) + \alpha_m(V(x,t)) \\
h'(x,t) &= -(\alpha_h(V(x,t)) + \beta_h(V(x,t))) \, h(x,t) + \alpha_h(V(x,t)),
\end{aligned}$$

will be studied. The parameters F_+ and F_- define the flow of electricity across neurons (or all the components of a system of neurons) or the electrical connection between neurons and can be quite complicated. In this study, only constant values for these are considered. However, even in this case one will observe that the electro-dynamics is sensitive to these parameters and can generate highly unpredictable outcomes.

We let $x = i\Delta x$ for $i = 1, \ldots, I$. In particular, $x_i = i\Delta x$ denotes the i^{th} neuron in our system and $x_{i-1} = (i-1)\Delta x, x_{i+1} = (i+1)\Delta x$ locate the neuron before and after the i^{th} neuron, respectively. In this Using, $V_i(t) = V(x_i, t), n_i(t) = n(x_i, t), m_i(t) = m(x_i, t), h_i(t) = h(x_i, t)$, the DPSM approximating the 1D cable PDE is the system of ODEs

$$\begin{aligned}
C_M \, V_i'(t) &= F_+ \, (V_{i+1} - V_i) + F_-(V_i - V_{i-1}) \\
&\quad - \left(\bar{g}_K \, n_i(t)^4 \, (V_i - E_K) + \bar{g}_{Na} \, m_i(t)^3 h_i(t)(V_i - E_{Na}\right) \\
&\quad + \bar{g}_L \, (V_i - E_L)) + (I_M)_i(t) \\
n_i' &= -(\alpha_n(V_i) + \beta_n(V_i)) \, n_i + \alpha_n(V_i) \\
m_i' &= -(\alpha_m(V_i) + \beta_m(V_i)) \, m_i + \alpha_m(V_i) \\
h_i' &= -(\alpha_h(V_i) + \beta_h(V_i)) \, h_i + \alpha_h(V_i)
\end{aligned}$$

for $i = 1, ..., I$. One can now use PSM with the polynomial approximations for $\alpha_n(V)$ and $\alpha_m(V)$ near the singularities to generate as high a degree (Maclaurin) polynomial approximation as desired for $i = 1, ..., I$.

For 2D, with $(x_i, y_j) = (i\Delta x, j\Delta y)$ for $i = 1, ..., I; j = 1, ..., J$, we use

$$\begin{aligned}
C_M\, V'_{i,j}(t) &= F_x\,(V_{i+1,j} - 2V_{i,j} + V_{i-1,j}) + F_y\,(V_{i,j+1} - 2V_{i,j} + V_{i,j-1}) \\
&\quad - \left(\bar{g}_K\, n_{i,j}(t)^4 \left(V_{i,j} - E_K\right) + \bar{g}_{Na}\, m_{i,j}(t)^3 h_{i,j}(t)(V_{i,j} - E_{Na}\right) \\
&\quad + \bar{g}_L \left(V_{i,j} - E_L\right)) + (I_M)_{i,j}(t) \\
n'_{i,j} &= -\left(\alpha_n(V_{i,j}) + \beta_n(V_{i,j})\right) n_{i,j} + \alpha_n(V_{i,j}) \\
m'_{i,j} &= -\left(\alpha_m(V_{i,j}) + \beta_m(V_{i,j})\right) m_{i,j} + \alpha_m(V_{i,j}) \\
h'_{i,j} &= -\left(\alpha_h(V_{i,j}) + \beta_h(V_{i,j})\right) h_{i,j} + \alpha_h(V_{i,j})
\end{aligned}$$

and for 3D, with $(x_i, y_j, z_k) = (i\Delta x, j\Delta y, k\Delta z)$ for $i = 1, ..., I; j = 1, ..., J; k = 1, ..., K$, we use

$$\begin{aligned}
C_M\, V'_{i,j,k}(t) &= F_x\,(V_{i+1,j,k} - 2V_{i,j,k} + V_{i-1,j,k}) + F_y\,(V_{i,j+1,k} - 2V_{i,j,k} + V_{i,j-1,k}) \\
&\quad + F_z\,(V_{i,j,k+1} - 2V_{i,j,k} + V_{i,j,k-1}) \\
&\quad - \left(\bar{g}_K\, n_{i,j,k}(t)^4 \left(V_{i,j,k} - E_K\right) + \bar{g}_{Na}\, m_{i,j,k}(t)^3 h_{i,j,k}(t)(V_{i,j,k} - E_{Na}\right) \\
&\quad + \bar{g}_L \left(V_{i,j,k} - E_L\right)) + (I_M)_{i,j,k}(t) \\
n'_{i,j,k} &= -\left(\alpha_n(V_{i,j,k}) + \beta_n(V_{i,j,k})\right) n_{i,j,k} + \alpha_n(V_{i,j,k}) \\
m'_{i,j,k} &= -\left(\alpha_m(V_{i,j,k}) + \beta_m(V_{i,j,k})\right) m_{i,j,k} + \alpha_m(V_{i,j,k}) \\
h'_{i,j,k} &= -\left(\alpha_h(V_{i,j,k}) + \beta_h(V_{i,j,k})\right) h_{i,j,k} + \alpha_h(V_{i,j,k})
\end{aligned}$$

for the DPSM together with LWM to generate high degree polynomials for the solutions. It is important to mention again that F_x, F_y, F_z can be complicated functions. However, once again it will be demonstrated that even for constant values for these, the electro-dynamics can be intriguing and help neuroscientists to better understand the dynamics of neuronal systems. One will observe that varying these parameters leads to the realization that understanding these connections biologically, chemically and electrically is important in gaining a true understanding of neuronal networks. Neuroscientists often say,’ Neurons that are wired together, fire together. ‘ The models presented in the section HH PDE Examples should make you think about this statement.

For the development of neuronal systems here, in 1D, each neuron depends intimately on its two closest neighbors, but eventually receives information from all the nodes, in 2D each neuron depends intimately on its four closest neighbors, but eventually receives information from all the nodes and in 3D each neuron depends intimately on its six closest neighbors, but again eventually receives information from all the nodes. Of course, one can change the stencil in 1D, 2D and 3D described above and have more (or less) intimate neighbors. Note that in 1D, one also has to have a value for V, n, m, h at $i = 0$ and $i = I + 1$, in 2D for $i = 0, j = 0, i = I+1$ and $j = J+1$ and in 3D for $i = 0, j = 0, k = 0, i = I+1, j = J+1$ and $k = K + 1$.

3.5 HH ODE Examples

In this section, we present solutions to the HH ODEs for all four cases of α_n and α_m described above for both the original HH model parameters and the Izhikevich (Iz) parameters given above using PSM8 with a time step of $h = 2^{-8}$. We highlight the noticeable similarities and differences in the output and the unusual outputs. We also demonstrate each of these cases under no current input ($I_M(t) = 0$), a single pulse current input ($I_M(t) = A\exp(-\omega(t-t_0)^2)$) and sinusoidal current inputs ($I_M(t) = A\sin\omega t$ and $I_M(t) = A\sin^2\omega t$). We run the simulation long enough for the voltage to reach equilibrium to verify that the numerical solutions hit the theoretical values. The initial conditions in all four models are $V(0) = V_0 = 0, n(0) = n_0 = 0.25, m(0) = m_0 = 0.25, h(0) = h_0 = 0.5$ and can be seen in each of the figures presented.

In Figure 16, the voltage is shown for both the HH and Iz parameters with $I_M(t) = 0$ for the four different α_n and α_m. PSM8 produced these results, but note that RK4 and ODE45 gave similar results. That is, all three numerical ODE solvers produced basically the same numerical approximation. It is interesting to observe that even though the HH and Iz parameters are close, the maximum voltage under HH is about 107 and under Iz it is about 112. The minimum voltages are about -11 for both of these parameter sets. The time interval is long enough to show that PSM8 (RK4 and ODE45) produced the theoretical equilibrium. One should also recognize that all four cases for α_n and α_m produced similar results for both HH and Iz.

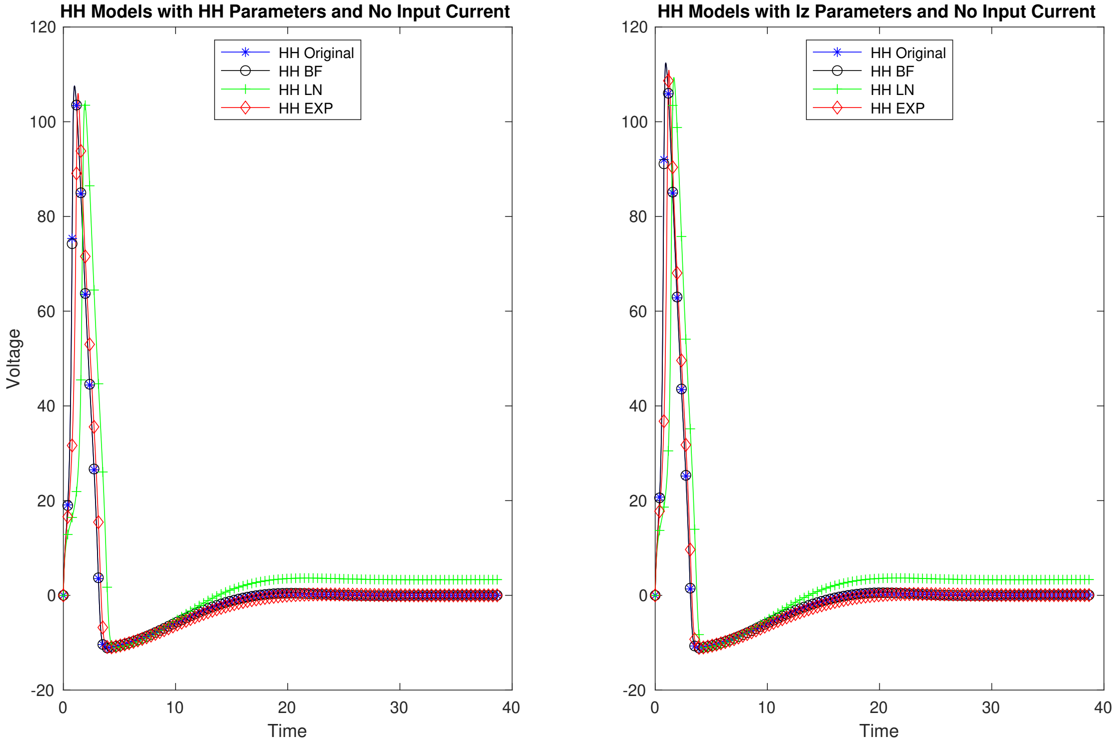

Figure 16: Action Potentials for the four models with HH and Iz Parameters.

The Maclaurin coefficients also tell an important story. The first nine Maclaurin coefficients (the coefficents of the 8^{th} degree Maclaurin polynomial approximation (PSM8)) for the original HH model at $t = 0$ (note $V(0) = 0$) are $\{0.0; 109.3089; -612.68282; 4131.44174; -28, 299.4293; 203, 295.9398; -1, 511, 196.8294; 11, 548, 619.453; -90, 113, 631.18\}$. An oscillating power series (especially one with coefficients increasing in magnitude) is well known to be difficult to approximate. The ratios of these Maclaurin coefficients are $\{0, -5.6, -6.74, -6.85, -7.18, -7.43, -7.64, -7.8, -7.93\}$. This shows why the HH ODEs are sensitive with potential for large errors when solved numerically. In fact, the coefficients can be shown to increase in magnitude and oscillate until they become too large for a computer to represent. A small time step ($h = 2^{-8}$) with 8^{th} degree Maclaurin polynomials was used in this study to handle the sensitive nature of the HH ODEs and because RK4 and ODE45 were not always able to generate a solution with a larger time step. Also, the small time step and agreement of all three numerical solvers verifies the highly sensitive nature of the HH ODEs.

This phenomenon of the Maclaurin coefficients does not occur in the time regions where the voltage is nearly constant, for example, near the equilibria (stable) or shortly after the action potential. For these regions, the Maclaurin coefficients tend to decrease quickly in magnitude. In models with oscillating inputs that lead to bursting spikes, the Maclaurin coefficients can be unwieldy.

Figure 17 presents the same four cases with input spike $I_M(t) = 10\exp(-0.125(t-15)^2)$. All the solvers produce basically the same approximation to V for all four cases. The input current's effect at $t = 15$ is clearly marked by a second action potential. Again it is seen that the Iz parameters lead to a larger voltage at the peak of the action potential.

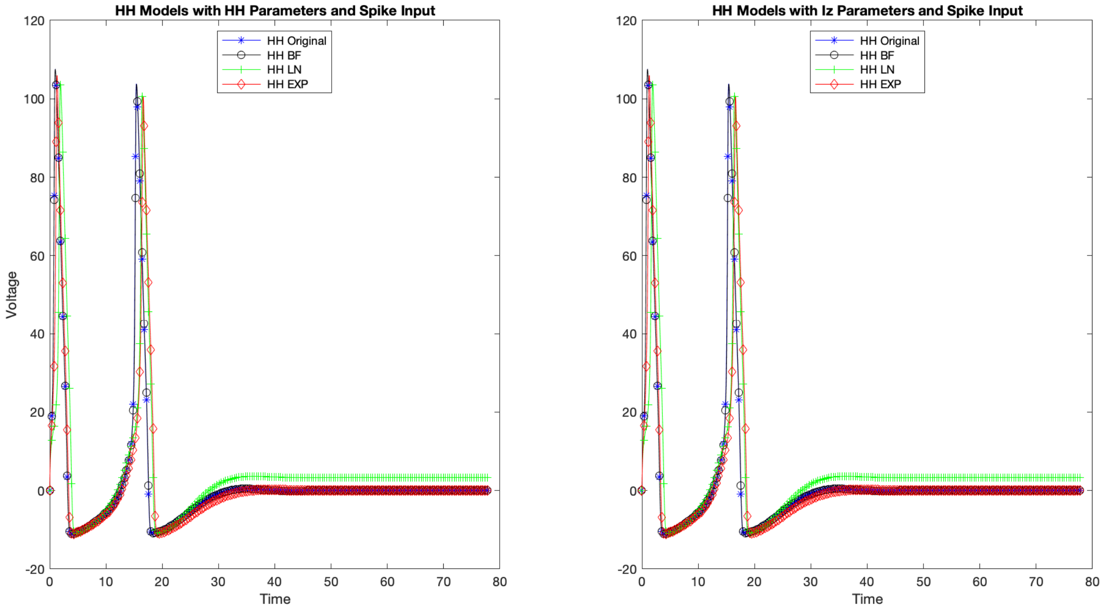

Figure 17: The Voltages for HH,BF,LN and EXP with a Spike Input.

In Figures 18 and 19, the same four cases are presented with $I_M(t) = 10\sin 0.125t$ and $I_M(t) = 10\sin 0.25t$, respectively. The similarities and differences between the function parameters and the frequencies is noticeable and unpredictable.

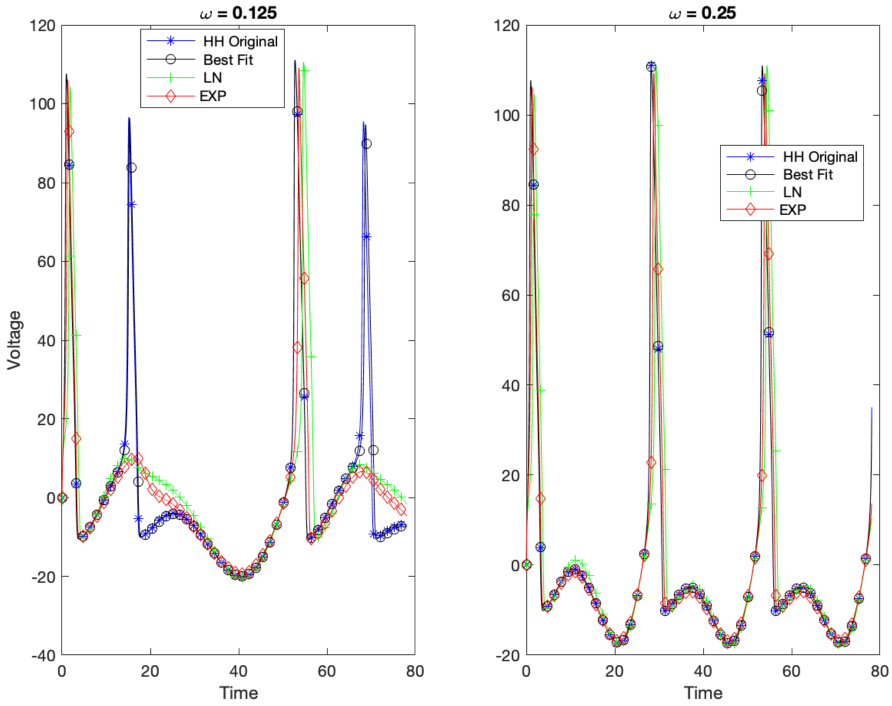

Figure 18: The HH Voltages Under Two Different Sinusoidal Inputs.

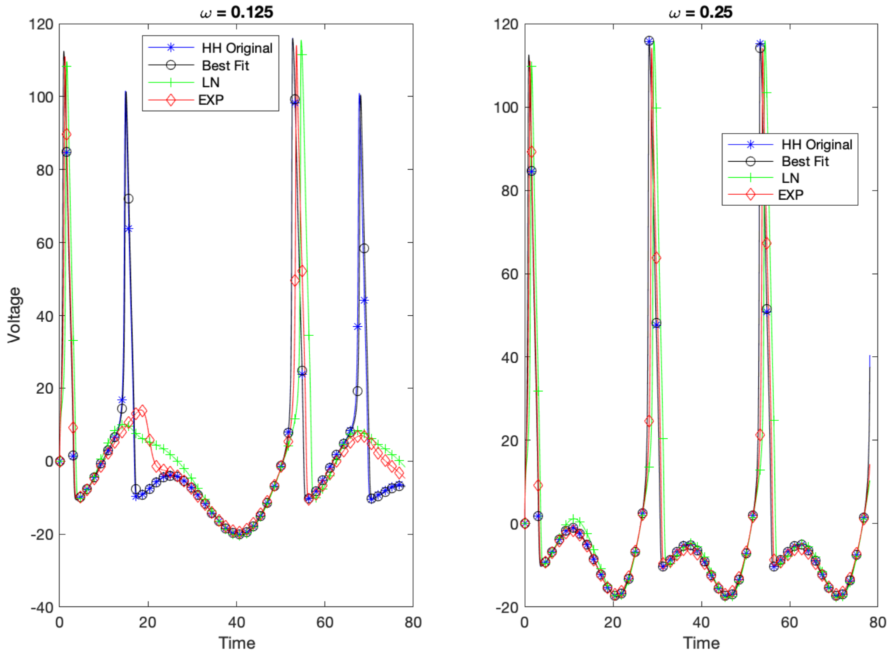

Figure 19: The IZ Voltages Under Two Different Sinusoidal Inputs.

Finally, we show how significantly different the results can be between the HH and Iz parameters under a high frequency input impulse. The high frequency input impulse $I_M(t) = 10\sin^2 t$ that generated Figures 20 through 22 shows that the voltage potentials can be unusually different and remarkably the same for the different function parameters. For the HH parameters all four voltage potentials are similar. However, for the Iz parameters the original and BF are the same, but much different than the HH and for the LN and EXP the Iz parameters are similar and similar to the HH parameter runs. In Figure 22 one can see the noticeable differences and similarities for the high frequency input impulse between the HH and IZ parameters and the four function parameters.

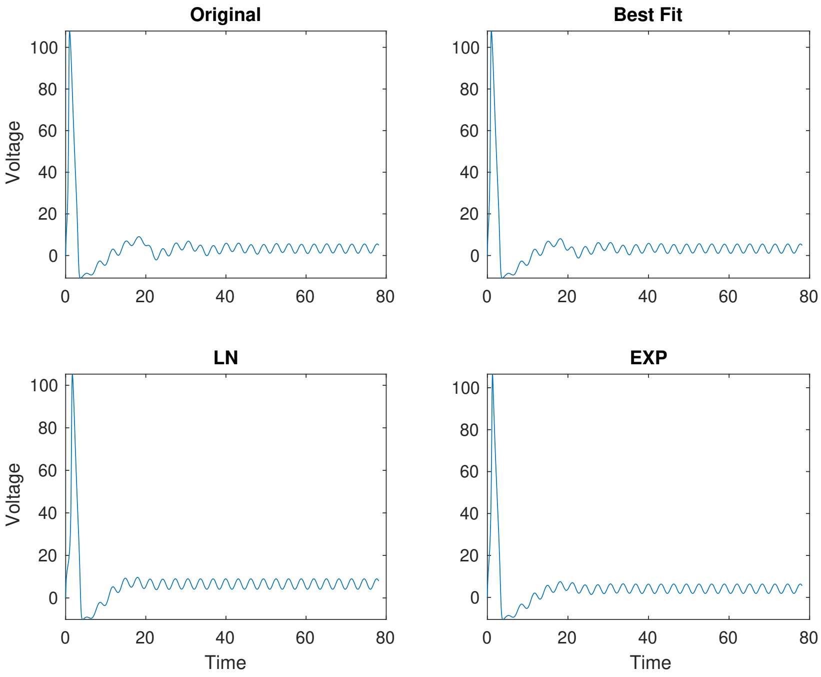

Figure 20: The HH Voltages for HH,BF,LN and EXP with a High Frequency Input.

3.6 HH PDE Examples

We now present the numerical results for a 1D, 2D and 3D array of neurons using the cable HH PDE and the approximations given above. All the numerical solutions presented in this section used DPSM8 (LWM8) with a time step of 2^{-8}. Again, the simulations are ran long enough for the voltage to reach equilibrium to verify that the numerical solutions hit the theoretical values. (The codes that generated the figures presented for the 1D, 2D and 3D arrays of neurons were written in c. The reader can email the author if they wish to obtain these codes.)

One should realize that if F_+ and F_- are set to 0 in the 1D case then we have a string of neurons that are independent of each other. If we give each neuron slightly different initial conditions then we can see how sensitive the HH ODEs are to initial conditions. This is demonstrated in Figures 23 and 24.

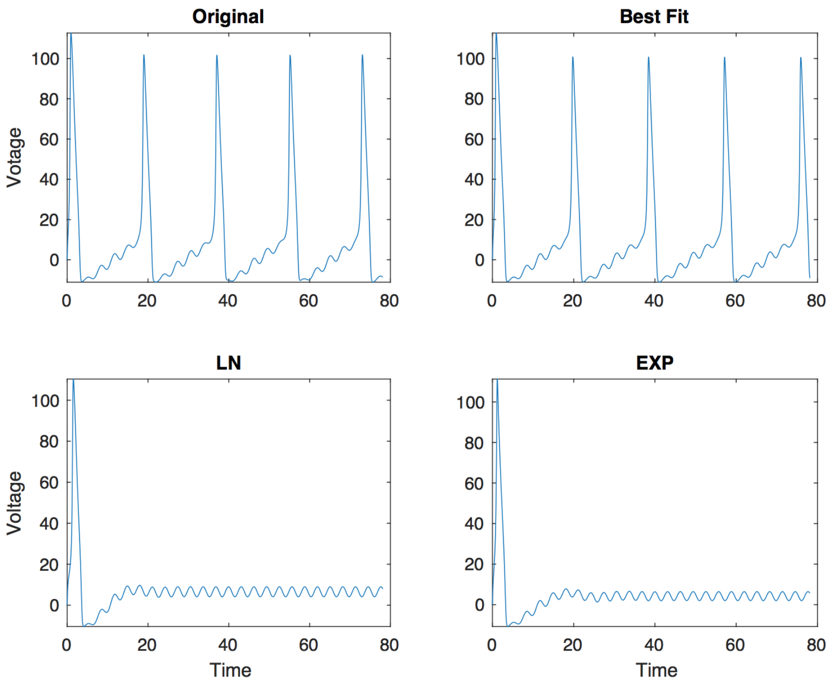

Figure 21: The Iz Voltages for HH,BF,LN and EXP with a High Frequency Input.

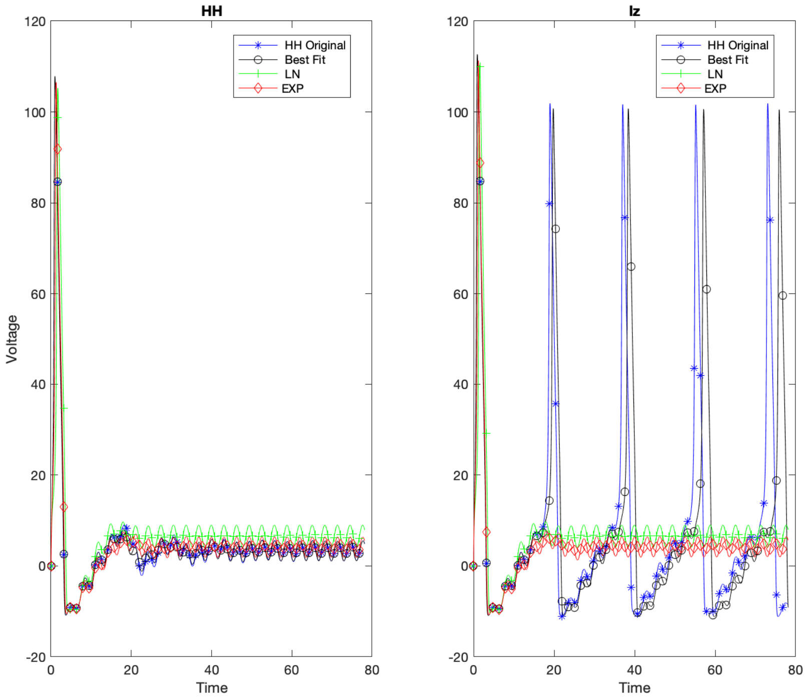

Figure 22: The Voltages for HH,BF,LN and EXP with a High Frequency Input.

In Figures 23 and 24 a string of 21 vertical neurons with V_0 starting at -10 (top) and incrementing by 1 to a value of 10 (bottom) with $n_0 = 0.5, m_0 = 0.25, h_0 = 0.25$ and the HH parameters for all the neurons is presented with $F_+ = 0 = F_-$ and with $F_+ = 0.25 = F_-$. There is no input current ($I_M(t) = 0$ for all 21 neurons). Figure 23 is a presentation of the two cases and the absolute value of the difference in the two cases over time as a firing plot. This means the brightest spots (near $t = 0$) are where the neurons fire (action potential). Figure 24 is a presentation of the same two cases and the absolute value of the difference in the two cases over time as a surface plot. That means the elevations represents the voltage. One can determine where the voltage potential is negative and where it is positive. One can see that for the positive initial voltages the voltages are about the same, but as the initial voltages go negative the outcome is quite different. One also notices the differences between $F_+ = 0 = F_-$ and $F_+ = 0.25 = F_-$. A stronger connection between the neurons can create significantly different results.

In Figures 25 and 26 a string of 21 vertical neurons all with $V_0 = 0, n_0 = 0.5, m_0 = 0.25, h_0 = 0.25$ and the HH parameters for all the neurons is presented with $F_+ = 0.25 = F_-$ and with $F_+ = 1 = F_-$. The nodes 10-12 are excited with the current $I_M(t) = 10 \sin 0.125t$. One can see how the input to nodes 10-12 propagates to the other neurons. There are noticeable differences between $F_+ = 0.25 = F_-$ and $F_+ = 1 = F_-$. Again showing that increasing the connection between the neurons gives observable changes. This is seen in Figures 25 and 26.

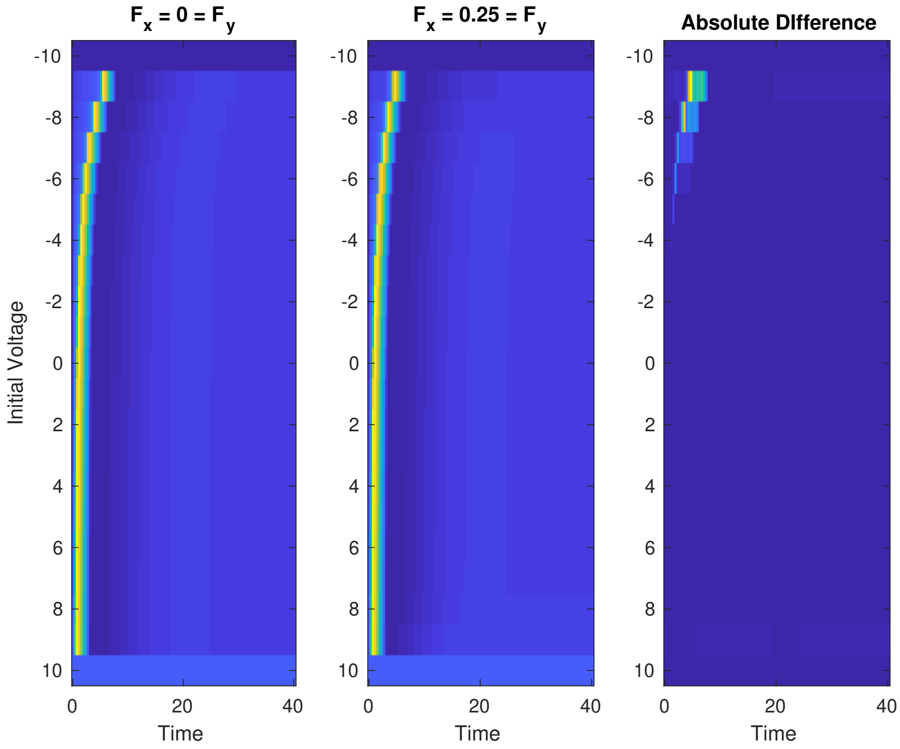

Figure 23: The Voltage Distribution over time for 21 Neurons with Different Initial Voltages and Different Connections Between the Neurons and Their Difference.

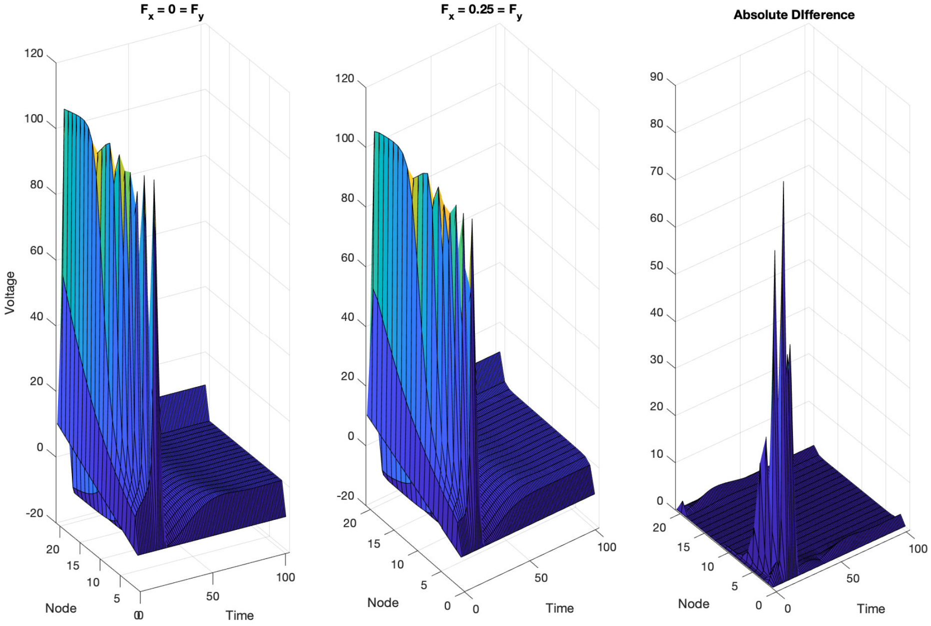

Figure 24: The Voltage Surface Across Time for the 21 Neurons with Different Connections Between the Neurons and Their Difference.

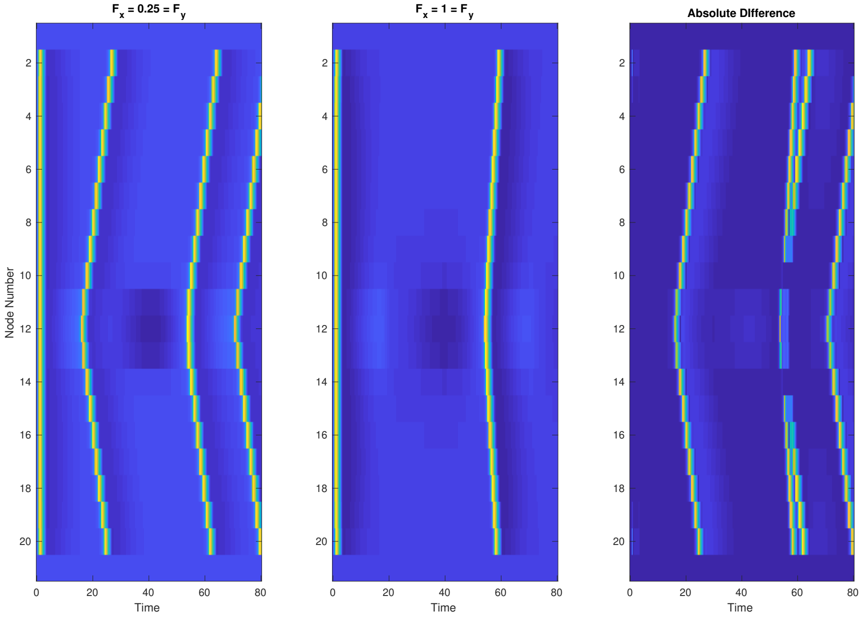

Figure 25: The Voltage Distribution Across Time for 21 Neurons with Different Connections and Middle Neurons Impulsed and Their Difference.

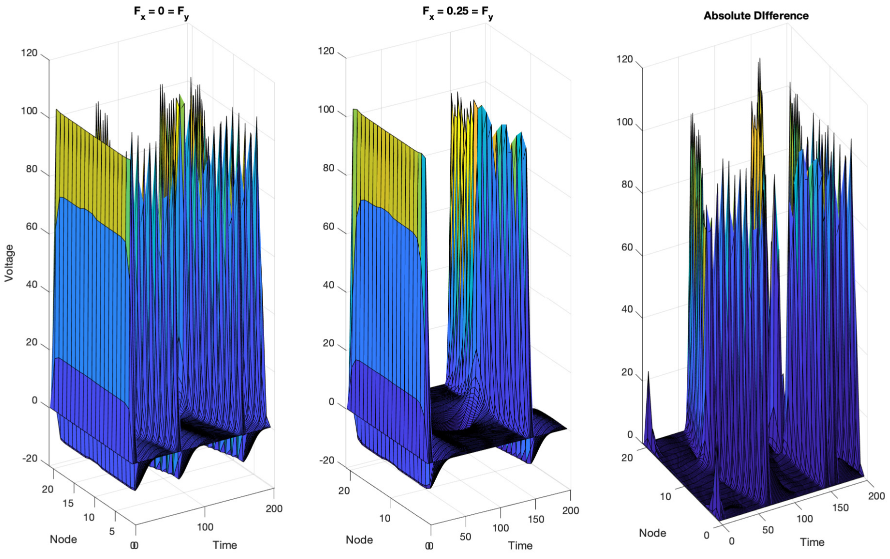

Figure 26: The Voltage Surface Across Time for 21 Neurons with Different Connections and Middle Neurons Impulsed and Their Difference.

Figure 27 presents the outcome of a 2D array of 21×25 neurons with the HH parameters, all initial conditions the same ($V_0 = 0, n_0 = 0.5, m_0 = 0.25, h_0 = 0.25$), with the center neurons under the input current $I_M(t) = 10 \sin 0.125t$ and with $F_x = 1 = F_y$ at nine different times as a surface plot with the vertical giving the voltage potential. At $t = 1.17$, one observes that all the interior neurons have achieved their action potential simultaneously. As time progresses, it can be seen that the voltage potential decreases to a negative value just as in the ODE case. The sinusoidal input starts to appear in the center of the neural array at $t = 55.1$. This impulse then begins to create an increasing voltage potential through the array of neurons in a radial fashion.

A 3D array of $11 \times 15 \times 17$ neurons under the HH parameters, with no impulse and with $F_x = F_y = F_z = 1$ at nine different times is shown in Figure 28. The initial conditions are given by $V_{i,j,k} = \exp(-0.0125((i-6)^2 + (j-8)^2 + (k-9)^2)), n_{i,j,k} = n_0 = 0.5$, $m_{i,j,k} = m_0 = 0.25$, $h_{i,j,k} = h_0 = 0.25$ for $i = 1..9, j = 1..13, k = 1..15$. Each neuron in the array is shown and the voltage potential is shown by color with green being the lowest, red the next and blue the largest. One can observe the action potentials of the neurons and see how the varying initial conditions creates voltage potential among the neurons in a rather unpredictable fashion.

If the reader wishes to view neurons firing in a 1D array, 2D array and 3D array, the reader can go to http://educ.jmu.edu/~sochacjs/NeuronAnimations/. The 1D animations show the firing of neurons along a cable of neurons. The 2D animations show the firing of the neurons in a 2D array through colors for the value of the voltage at each neuron. One can watch the firings move through this 2D network of neurons. The 3D animations shows the firing of voltages in two scenarios, one in a 3D array where colors show the values of the voltage at each neuron in the array and in a 3D CAVE architecture

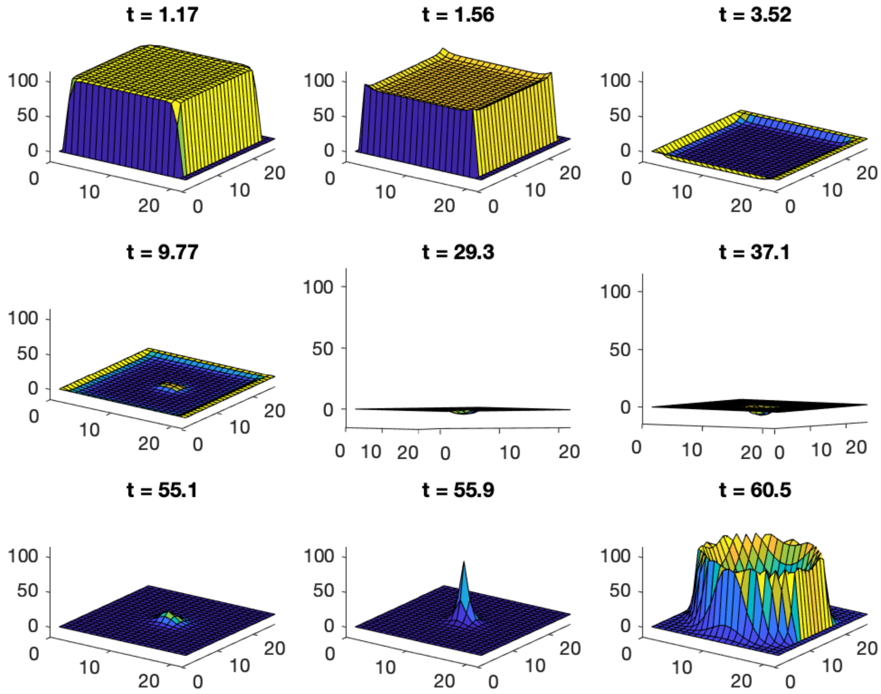

Figure 27: The Voltage Surface Across Time for a 2D Array of 21 × 25 Neurons with Middle Neurons Impulsed.

where lights show the firings. The document DescriptionOfNeuronAnimations.pdf gives a brief description of each animation at this site.

4 Conclusion

In Part I, the development of the HH ODEs were outlined and then analyzed to determine the equilibrium solutions. PSM8 was used to show what these solutions look like. Next the neuron was given three different input impulses to show that the dynamics of the neuron can become complicated yet still attain equilibrium solutions or solutions that become cyclical including close to periodic. Having numerical solutions allows one to obtain an accurate estimate of the equilibria and the shape of the cyclical nature of neurons.

In Part II, we considered the sensitive dependence of the HH ODEs and the removable singularities of α_n and α_m. We then extended the HH ODEs to PDEs as discussed in the 1952 Hodgkin Huxley paper and showed that with Matlab, one can now easily do what Hodgkin and Huxley only hoped to do.

Four methods were presented to address the removable singularities pointed out in Stewart and Bair when using PSM. Under a small t step, all four methods gave similar results and are similar to the numerical solutions given by RK4 and ODE45. However, it was shown that under a sinusoidal impulse, the results can vary among the four types of α presented and the parameters of the HH ODEs (HH vs Iz).

This demonstrates that having highly accurate numerical solvers for HH type models

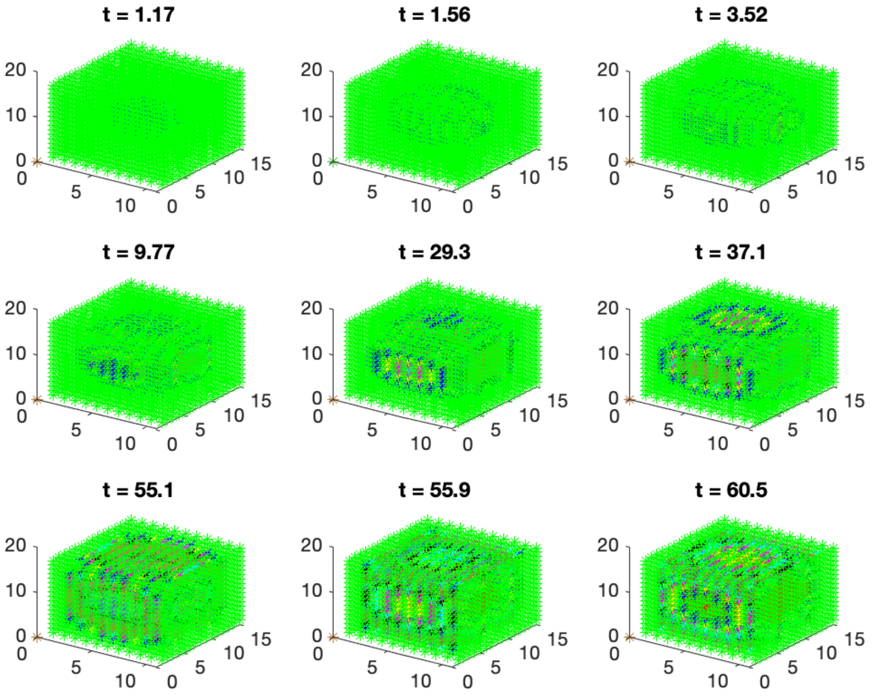

Figure 28: The Voltage Across Time for a 3D Array of 11x15x17 Neurons with Varying Initial Conditions

is necessary. Also, comparing numerical results amongst a variety of numerical solvers is a valuable task in the understanding of the complexity of neurons.

Under arrays of 1D, 2D and 3D of neurons, one can visualize the propagation of voltage through the neurons. Noticeable differences under a variety of initial conditions and impulses were also demonstrated.

Simulations of HH type neuron models under a variety of conditions and impulses using highly accurate methods will help in understanding the complexity of neuronal networks and lead to more accurate models. This will require better computational and visualization schemes that researchers will develop from the learning and understanding of the previous simulations.

Acknowledgments

I thank G.E. Parker for all the years we talked about the Parker Sochacki Method (PSM), the JMU PSM team (S. Lucas, R. Thelwell, A. Tongen, P. Warne) for all their encouragements and thoughts and all the other mathematicians I met because of PSM. I also thank all the undergraduate students who over the last 30 years have worked with me on PSM, with a congratulations to those undergraduates who were able to get their work published. Special thanks to my undergraduate research student Jeffrey Kopsick for help in writing the codes that made this paper possible.

References

[1] Hodgkin AL, Huxley AF. (1952) "A quantitative description of membrane current and its application to conduction and excitation in nerve". The Journal of Physiology. 117 (4): 500-544, August 1952.

[2] Cole, K. S., (1949), "Dynamic electrical characteristics of the squid axon membrane, " Arch. Sci. Physiol. 3, 253.

[3] Marmont, G., (1949), "Studies on the axon membrane. I. A new method, J. Cellular Comp. Physiol. 34, 351.

[4] Peterson, G. and Sochacki, J., (2001), *Linear Algebra and Differential Equations.* Boston: Pearson.

[5] Borisyuk, A., Ermentrout, G.B., Friedman, A., Terman, D.H., (2008), Tutorials in Mathematical Biosciences I, Springer.

[6] Izhikevich, E. M., (2007), Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting, MIT Press.

[7] Warne, P. G., Warne, D. A., Sochacki, J. S., Parker, G. E. and Carothers, D. C., (2006), Explicit a-priori error bounds and adaptive error control for approximation of nonlinear initial value differential systems, Comput. Math. Appl., 52 (2006), 1695-1710.

[8] Parker, G.E. and Sochacki, J.S., (1996), Implementing the Picard Iteration, Neural, Parallel & Scientific Computations, Vol 4, No 1, 97-112, 1996.

[9] Stewart, R.D. and Bair, W., (2009), Spiking neural network simulation: numerical integration with the Parker-Sochacki method, J. Comput. Neurosci., 27, 115-133, 2009.

[10] Stewart, R.D., Gurney, K.N., (2011), Spiking neural network simulation: memory-optimal synaptic event scheduling. J. Comput. Neurosci., 30, 721-728, 2011.

[11] Gofen, A., (2009), The ordinary differential equations and automatic differentiation unified, Complex Variables and Elliptic Equations, Vol. 54, No. 9, 828-854, 2009.

[12] Neidinger, R., (2010), Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. SIAM Review, 52(3), 545-563.

[13] Mirzaee, Farshid, (2011), Differential transform method for solving linear and nonlinear systems of ordinary differential equations, Appl. Math. Sci. 5 (70), 3465-3472.

[14] Pruett, C.D., Rudmin, J.W. and Lacy, J.M., (2003), An Adaptive N-Body Algorithm of Optimal Order. Journal of Computational Physics, 187, 298-317.

[15] Nurminskii, E. and Buryi, A., (2011), Parker-Sochacki Method for Solving Systems of Ordinary Differential Equations Using Graphics Processors. Numerical Analysis and Applications, 4, 223.

[16] Pruett, C.D., Ingham, W.H. and Herman, R.D., (2011), Parallel Implementation of an Adaptive and Parameter-Free n-Body Integrator. Computer Physics Communications, 182, 1187-1198.

[17] Szynkiewicz, P., (2016), A Novel GPU-Enabled Simulator for Large Scale Spiking Neural Networks. Journal Telecommunications and Information Technology, 2, 34-42.

[18] Yudanov, D., Shaaban, M., Melton, R. and Reznik, L., (2010), GPU-Based Simulation of Spiking Neural Networks with Real-Time Performance & High Accuracy. The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, 18-23 July 2010, 1-8.

[19] Money, J.H., (2006), Variational Methods for Image Deblurring and Discretized Picard's Method. PhD Thesis, University of Kentucky, Lexington.

[20] Lax, P.D., Wendroff, B., (1960), Systems of conservation laws. Commun. Pure Appl. Math. 13 (2): 217-237.

[21] Sochacki, J.S., (2010), Polynomial odes: examples, solutions, properties. Neural, Parallel Sci. Comput. 18, 3–4 (September-December 2010), 441-450.

[22] Thelwell, R.J, Warne, P.G., Warne, D.A., (2012), Cauchy-Kowalewski and Polynomial Ordinary Differential Equations, Electronic Journal of Differential Equations, Vol. 2012 No. 11, 1-8.

[23] Carothers, D. C., Parker, G. E., Sochacki, J. S. and Warne, P. G., (2005), Some properties of solutions to polynomial systems of differential equations, Electron. J. Differential Equations, 2005, 1-17.

[24] Fehlberg, E., (1964), Numerical integration of differential equations by power series expansions, illustrated by physical examples. Technical Report NASA-TN-D-2356, NASA, 1964.

[25] Ermentrout, G. Bard, Terman, David H., (2010), Mathematical Foundations of Neuroscience, Springer.