

# Sloppy Model Analysis of Buchholtz Neuron Model

Jonathan Daniel Treter

A capstone report submitted to the faculty of

Brigham Young University

In partial fulfillment of the requirements for the degree of

Bachelor of Science

Dr. Mark Transtrum, Advisor

Department of Physics and Astronomy

Brigham Young University

June 16, 2021

Copyright © 2021 Jonathan Daniel Treter

All Rights Reserved

## ABSTRACT

### Sloppy Model Analysis of Buchholtz Neuron Model

Jonathan Daniel Treter

Department of Physics and Astronomy

Bachelor of Science

Neurons are complex physical systems with many interacting components. The foundational model of neural behavior is the Hodgkin-Huxley model. It models the cell membrane as a capacitor and protein ion channels as voltage-dependent resistors. The membrane voltage responds to an applied current and is calculated as a system of differential equations using standard circuit analysis. The Hodgkin-Huxley model involves four dynamical variables and 26 parameters; however, previous work explicitly constructing a reduced-order approximation showed that many of these parameters are irrelevant. A more realistic model from Buchholtz et al. expands on the model of Hodgkin-Huxley and involves 14 dynamical variables and 68 parameters. We implement the Buchholtz model in the Julia programming language and conduct a “sloppy model” analysis of the parameters. We show that this model is sloppy, meaning the importance of parameters used to explain the model behavior is exponentially distributed. Most of this behavior can be explained by a reduced number of combinations of parameters, suggesting that the model can be approximated by a low-order, reduced model. This work lays the foundation for a future parameter reduction analysis to find a simplified version of the Buchholtz model.

Keywords: Buchholtz model; neuron; sloppiness; stiffness; singular value decomposition;  
model reduction; emergence

# Acknowledgments

---

I express my thanks to Dr. Mark Transtrum for his guidance and instruction that enabled me to work on this project and learn more about the geometry of mathematical models. I also thank my teammates Joshua Newey and Joshua Rasband for their help implementing and understanding the model. Finally, I thank the BYU department of Physics and Astronomy for providing research funding that helped to make this project possible.

# Table of Contents

---

<b>Table of Contents.....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>vii</b>
<b>List of Tables.....</b>	<b>viii</b>
<b>1 Introduction .....</b>	<b>1</b>
1.1 <i>Motivation</i> .....	1
1.1.1 Emergent Theories in Physics .....	1
1.1.2 Neurological Model Sloppiness .....	2
1.2 <i>Background</i> .....	3
1.3 <i>Previous Research at BYU</i> .....	4
1.4 <i>Application to New Model</i> .....	4
<b>2 Methods .....</b>	<b>5</b>
2.1 <i>The Buchholtz Model</i> .....	5
2.1.1 Overview .....	5
2.1.2 Dynamical Variables.....	6
2.1.3 Parameters.....	8
2.2 <i>Computational Formulation</i> .....	10
2.2.1 Target Behavior.....	10
2.2.2 Julia Module.....	12
2.3 <i>Analysis Methods</i> .....	12

2.3.1	Fisher Information .....	13
2.3.2	Singular Value Decomposition .....	13
<b>3</b>	<b>Results and Conclusions.....</b>	<b>15</b>
3.1	<i>Singular Values</i> .....	15
3.2	<i>Singular Vectors</i> .....	18
3.3	<i>Future Work</i> .....	21
3.4	<i>Conclusion</i> .....	23
	<b>Appendix .....</b>	<b>24</b>
	<i>Julia Module</i> .....	24
	<b>Index.....</b>	<b>35</b>
	<b>References.....</b>	<b>36</b>

# List of Figures

---

Figure 1 - Target Behavior.....	11
Figure 2 - Eigenvalues of FIM.....	16
Figure 3- Right Singular Vectors.....	17
Figure 4 – Left Singular Vector 1 .....	18
Figure 5 – Left Singular Vector 2 .....	19
Figure 6 – Left Singular Vector 3 .....	20
Figure 7 – Left Singular Vector 4 .....	20
Figure 8 – Left Singular Vector 68 .....	21

# List of Tables

---

Table 1 – Functions and dynamical variables .....	7
Table 2 - Current parameters .....	9
Table 3 - Additional Parameters .....	10



# 1 Introduction

---

## 1.1 Motivation

First principles neurological models can be extremely complex, relying on many physical processes at the cellular level to describe the neurons' behavior. It is common for models of single neurons to involve more than a half-dozen different ionic currents, each of which is defined by several parameters that are inferred from data. Combining these single-neuron models into models of neurological systems introduces more parameters to describe intercellular interactions, which results in computationally intensive models. Model reduction is therefore an important tool for computational neuroscience.

### 1.1.1 Emergent Theories in Physics

To physicists, the complexity of neurological models that arises from building up the system from numerous smaller components is reminiscent of statistical mechanics. Physicists

commonly face problems in which many interacting entities form a system that depends on a few new, emergent parameters rather than on the many parameters of its components. This process, known as emergence, is especially clear in kinetic theory. Given a large enough population of particles, the behavior of the system as a whole is no longer best described by the speeds, separations, and masses of the individual particles, but instead by the emergent temperature, total volume and pressure of the whole system. These new combined parameters allow the system, though extremely complex at a microscopic level, to be easily described at the macroscopic level with startling accuracy.

### 1.1.2 Neurological Model Sloppiness

Modern research into the statistical mechanics of inhomogeneous systems, such as neuron models, gives rise to the idea of model ‘sloppiness’. In this idea, parameters and their combinations within a model are described as ranging from *stiff* to *sloppy*. Sloppy parameters are combinations that can be changed many orders of magnitude while the model’s predictions remain mostly unaltered. Stiff parameters on the other hand have a strong effect on the behavior of the model. Continuing with the example of kinetic theory, temperature would be a stiff parameter, while the individual energies of the particles would be sloppy parameters. By forming a renormalized combination of sloppy parameters, i.e., averaging together the individual energies, a single stiff parameter, temperature, emerges. This new stiff parameter describes the system without needing to know the energies of the individual particles.

There are many reasons why models exhibit sloppiness. Stiff parameters like pressure and entropy are more convenient and relevant for describing a system of particles; however, when constructing a model from basic components they are not readily apparent. The properties of

separate particles are the clear basis to work in to construct the model even though they are not a phenomenological basis. As computers advance in memory and processing speed, it becomes increasingly practical to handle sloppy models with dozens to hundreds of parameters.

Neurological models have been shown in many instances to exhibit sloppiness (Ryan N Gutenkunst 2007). The goal of this paper is to perform a sloppy model analysis on a model proposed by Buchholtz et al. (F. Buchholtz 1992), identify the stiff parameter combinations and set the groundwork for future work to redefine the model in terms of these stiff combinations, i.e., reduce it to a minimal form.

## **1.2 Background**

Neurons primarily consist of a cell body (soma), dendrites, and an axon (Guy-Evans 2021). The role of a neuron is to transmit an electrical signal received at the dendrites away through the axon. While the cell body and dendrites are close together, the axon can reach from 0.1 mm to 1 m away from the cell to carry signals. These signals are called action potentials and are fluctuations in the electric potential across the neuron's cell membrane. These potentials are facilitated by ion channels and pumps that are distributed across the surface of the cell. The channels and pumps allow controlled flow of ions across the cell to detect and transmit potentials (Helen C. Lai 2006). The purpose of the conductance-based neuron model that will be analyzed in this paper is to describe the action potential by building it up as a system of differential equations representing the different ion channels in the cell by treating the channels as conductances, the pumps as batteries, and the membrane as a capacitor, and then solving the resulting electrical system for the voltage across the membrane.

## 1.3 Previous Research at BYU

The Transtrum research group at Brigham Young University studies sloppiness in multi-parameter models and model reduction (Mark K. Transtrum 2015). Previous work on the Hodgkin Huxley model (A. L. Huxley 1952), a foundational neuron model describing the giant squid axon that won the Nobel Prize in Physiology or Medicine in 1963, has shown it to exhibit sloppiness and that it can be approximated with fewer variables by using manifold boundary approximation to reduce the model to one based solely on the stiffest parameter combinations (Rasband 2021). In this paper, we will begin extending these results to more complex single-neuron models.

## 1.4 Application to New Model

The selected model for this new study is from Buchholtz et al. (F. Buchholtz 1992) in describes activation and deactivation of a neuron (stomatogastric ganglion in a rock crab). At 68 parameters and 14 dynamical variables it is over twice as complex as the original Hodgkin-Huxley model. The Buchholtz model is a standard model in computational neuroscience that is still extensively used in studies of neuronal systems (Fox 2017) (Stein n.d.) (Gorur-Shandilya 2020). As such, it is an excellent target system to study. The analysis in Chapter 2 will focus on the structure of the model, how it describes activation and inactivation of the neuron, and how we quantify its sloppiness.

## 2 Methods

---

This chapter describes our implementation of the Buchholtz model and the details of the sloppy model analysis. The model is formulated as a parameterized dynamical system. Dynamical variables represent physical quantities whose value change with time, such as voltage across the membrane. The dynamical variables constitute a system of 14 coupled first-order ordinary differential equations that describes the activation and inactivation of the neuron. Parameters are constant inputs to the model. The parameters and their importance for explaining model behavior are elucidated by sloppy model analysis and are the focus of the present study. The exact equations used for the dynamical variables and values of the parameters are presented in Tables 1-3.

### 2.1 The Buchholtz Model

#### 2.1.1 Overview

Buchholtz models the change in voltage across the neuron membrane as the sum of seven ionic currents: an external current, which is a control in laboratory conditions or an input from another neuron; three currents of  $K^+$ : a delayed rectifying current, an inward rectifying current, a calcium-activated current and an A-type current; a  $Ca^{2+}$  current; a  $Na^+$  current; and a leak current. Of these, the  $Na^+$  current is by far the largest. Each of these currents is in turn described by up to three dynamical variables representing activation and inactivation processes. The calcium and calcium-activated currents are unique in that they respectively control and depend on the  $Ca^{2+}$  concentration in the cell, which has its own dynamical variable depending on the calcium current. This gives a total of 14 dynamical variables in the model.

The dynamical system is parameterized by 68 different values. These parameters consist of the following: a maximum conductance for each current; a reversal potential to determine the direction of flow of each type of ion; a rate constant for the change of each dynamical variable; a half-maximum potential and step width for each of 17 voltage dependent quantities; the membrane capacitance; and several others that influence the calcium concentration and the sodium current. The dynamical system additionally depends on four physical constants: the gas constant  $R$ , temperature  $T$ , Faraday constant  $F$ , and the charge of  $\text{Ca}^{2+}$   $z$ .

While these are natural parameter choices for physical measurements and the mathematical model, they likely do not correspond to the model's natural phenomenological basis for the resulting behavior. We expect these many microscopic parameters can be renormalized together to determine a reduced basis of effective parameters. We also expect that this will remove dynamical variables.

### 2.1.2 Dynamical Variables

The dynamical variables of the model are presented in Table 1 as they were implemented in the model. The primary dynamical variable in the model is the voltage. We sample the predicted voltage of the model. We perform our sloppy model analysis with respect to this key behavior. The dynamics of the other 13 variables are given by linear differential equations that give a rate of change proportionate to the difference between the current value and a target steady state value. Both the target steady state and decay rates are nonlinear functions of voltage. The calcium concentration rate also includes a term  $c_{iCa}$  proportional to the calcium current  $I_{Ca}$ . Each of the steady-state-seeking variables depends on at least a rate constant parameter, a half-

maximum potential, and a step width. Several have additional half-maximum potentials and step widths to describe voltage-dependent rates or more complex steady states.

Table 1 – Functions and dynamical variables

*H* is used to represent the sigmoid function.

Voltage	
$\frac{dV}{dt} = (I_{ext} - I_d - I_{o(Ca)} - I_A - I_{Ca} - I_h - I_{Na}) \quad (1)$	
Delayed Rectifier Current	
$I_d = \bar{g}_d * n^4 * (V - E_k)$	
$n_\infty = H\left(\frac{V - V_{kn}}{s_n}\right) \quad (2)$	$\frac{dn}{dt} = (n_\infty - n)k_n \quad (3)$
Calcium Activated Outward Current	
$I_{o(Ca)} = \bar{g}_{o(Ca)} * a_o * b_o * (V - E_k)$	
$a_{o\infty} = H\left(\frac{V - V_{ao1} - f * [Ca]}{s_{ao1}}\right) \quad (4)$	$\frac{da_o}{dt} = (a_{o\infty} - a_o)k_{oa} \quad (5)$
$* H\left(\frac{V - V_{ao2} - f * [Ca]}{s_{ao2}}\right) \frac{[Ca]}{c1 + [Ca]}$	$\frac{db_o}{dt} = (b_{o\infty} - b_o)k_{ob} \quad (6)$
$b_{o\infty} = \frac{c_2}{c_3 + [Ca]} \quad (7)$	$\frac{d[Ca]}{dt} = -c_{ica} * I_{Ca} - k_{Ca} * ([Ca] - C_{ao}) \quad (8)$
A-type Current	
$I_A = \bar{g}_A * a_A^3 * b_A * (V - E_k)$	
$a_{A\infty} = H\left(\frac{V - V_A}{s_A}\right) \quad (9)$	$\frac{da_A}{dt} = (a_{A\infty} - a_o)k_A \quad (10)$
$b_{A\infty} = H\left(\frac{V - V_B}{s_B}\right) \quad (11)$	$\frac{db_{A1}}{dt} = (a_{o\infty} - b_{A1})k_{A1} \quad (12)$
$b_A = H\left(\frac{V - V_X}{s_X}\right) b_{A1}$	$\frac{db_{A2}}{dt} = (b_{A\infty} - b_{A2})k_{A2} \quad (14)$
$+ \left(1 - H\left(\frac{V - V_X}{s_X}\right)\right) b_{A2} \quad (13)$	
$k_{A2} = c_{A2} * H\left(\frac{V - V_{A2}}{s_{A2}}\right) \quad (15)$	
Calcium Current	
$I_{Ca} = (\bar{g}_{Ca1} * a_{Ca1} * b_{Ca1} + \bar{g}_{Ca2} * a_{Ca2}) * (V - E_{Ca})$	

$a_{Ca1\infty} = H\left(\frac{V - V_{aCa1}}{s_{aCa1}}\right)$	(16)	$\frac{da_{Ca1}}{dt} = (a_{Ca1\infty} - a_{Ca1})k_{aCa1}$	(17)
$b_{Ca1\infty} = H\left(\frac{V - V_{bCa2}}{s_{bCa1}}\right)$	(18)	$\frac{db_{Ca1}}{dt} = (b_{Ca1\infty} - b_{Ca1})k_{bCa1}$	(19)
$a_{Ca2\infty} = H\left(\frac{V - V_{aCa1}}{s_{aCa1}}\right)$	(20)	$\frac{da_{Ca2}}{dt} = (a_{Ca2\infty} - a_{Ca2})k_{aCa2}$	(21)
Inward Rectifier Current			
$I_h = \bar{g}_h * r * (V - E_h)$			
$r_\infty = H\left(\frac{V - V_r}{s_r}\right)$	(22)	$\frac{dr}{dt} = (r_\infty - r)k_r$	(23)
$k_r = c_r / H\left(\frac{V - V_{kr}}{s_{kr}}\right)$	(24)		
Fast Sodium Current			
$I_{Na} = \bar{g}_{Na} * m^3 * h * (V - E_{Na})$			
$m_\infty = \frac{a_m}{a_m + b_m}$	(25)	$\frac{dm}{dt} = (m_\infty - m)k_m$	(26)
$a_m = \frac{c_{am}(V - V_{am})}{1 - \exp\left(\frac{V - V_{am}}{s_{am}}\right)}$	(27)		
$b_m = c_{bm} \exp\left(\frac{V - V_{bm}}{s_{bm}}\right)$	(28)		
$h_\infty = \frac{a_h}{a_h + b_h}$	(29)	$\frac{dh}{dt} = (h_\infty - h)k_h$	(30)
$a_h = c_{ah} \exp\left(\frac{V - V_{ah}}{s_{ah}}\right)$	(31)		
$b_h = H\left(\frac{V - V_{bh}}{s_{bh}}\right)$	(32)		
Leak Current			
$I_l = \bar{g}_l * (V - E_l)$			

### 2.1.3 Parameters

The 68 parameters of the model are given in Table 2 and Table 3. They are organized by which dynamical variable they influence. For the purposes of analysis, the rate constants were



divided by 1 000 to convert them from Hz to kHz and give time values in milliseconds.

Otherwise, the parameters were retained as presented in Buchholtz's model (F. Buchholtz 1992).

Table 2 - Current parameters

Current	Maximum Conductance ( $\mu\text{S}$ )	Reversal Potential (mV)	Rate Constant (Hz)	Half-Maximum Potential (mV)	Step Width (mV)
Delayed Rectifier, $I_d$	$\bar{g}_d = 0.35$	$E_k = -80$	$c_n = 180$	$V_n = -25$ $V_{kn} = 10$	$s_n = -17$ $s_{kn} = -22$
Calcium-Activated Outward, $I_{o(\text{Ca})}$	$\bar{g}_{o(\text{Ca})} = 3.2$	$E_k$	$k_{oa} = 600$ $k_{ob} = 35$ $k_{Ca} = 360$	$V_{ao1} = 0$ $V_{ao2} = -16$	$s_{ao1} = -23$ $s_{ao2} = -5$
Transient A-type $I_A$	$\bar{g}_A = 2.2$	$E_k$	$k_A = 140$ $k_{A1} = 50$ $c_{A2} = 3.6$	$V_A = -12$ $V_B = -62$ $V_{A2} = -40$ $V_X = 7$	$s_A = -26$ $s_B = 6$ $s_{A2} = -12$ $s_X = -15$
$\text{Ca}^{2+}$ $I_{\text{Ca}}$	$\bar{g}_{Ca1} = 0.21$ $\bar{g}_{Ca2} = 0.047$	$E_{Ca} = \frac{RT}{zF} \log\left(\frac{13000}{[\text{Ca}^{2+}]}\right)$	$k_{aCa1} = 50$ $k_{bCa1} = 10$ $k_{aCa2} = 16$	$V_{aCa1} = -11$ $V_{bCa1} = -50$ $V_{aCa2} = 20$	$s_{aCa1} = -7$ $s_{bCa1} = 8$ $s_{aCa2} = -7$
Inward Rectifier $I_h$	$\bar{g}_h = 0.037$	$E_h = -10$	$c_r = 0.33$	$V_r = -70$ $V_{kr} = -110$	$s_r = 7$ $s_{kr} = -13$
Fast $\text{Na}^+$ $I_{\text{Na}}$	$\bar{g}_{Na} = 2300$	$E_{Na} = 50$	$k_m = 10\,000$ $k_n = 500$	$V_{am} = -6$ $V_{bm} = -34$ $V_{ah} = -39$ $V_{bh} = -40$	$s_{am} = -20$ $s_{bm} = -13$ $s_{ah} = -8$ $s_{bh} = -5$
Leak $I_l$	$\bar{g}_l = 0.1$	$E_l = -50$			

Table 3 - Additional Parameters

Additional Fast $\text{Na}^+$ ( $\text{mV}^{-1}$ )	$c_{am} = 0.11$ $c_{bm} = 15$ $c_{ah} = 0.08$
$[\text{Ca}^{2+}]$ Parameters ( $\text{mV}/\mu\text{M}$ )	$f = 0.6$
$[\text{Ca}^{2+}]$ Parameters ( $\mu\text{M}$ )	$c_1 = 0.08$ $c_2 = 0.08$ $c_3 = 0.08$ $[\text{Ca}^0] = 0.05$
$[\text{Ca}^{2+}]$ Parameters ( $\mu\text{M}/\text{nC}$ )	$c_{ica} = \frac{1}{z F Vol} = 300$
Membrane Capacitance ( $\text{nF}$ )	$C_m = 1.7$
Temperature	$T = 283 \text{ K}$

## 2.2 Computational Formulation

The model was implemented in the Julia environment so that it could be analyzed using libraries previously created by the Transtrum group (Transtrum 2021). The Julia model’s purpose is to evaluate the voltage of the model at multiple time points (i.e., the “target behavior” described below) for a given set of parameter values. It then calculates the sensitivity of these predictions to infinitesimal variations in these parameter values.

### 2.2.1 Target Behavior

The chosen behavior to sample was the voltage response to brief input currents of constant magnitude, as shown in Figure 7 of Buchholtz et al. and reproduced in Figure 1 below

using our implementation. Six different magnitudes of the applied input current were sampled ranging from 7 nA to -5 nA at 2.5 nA intervals, with positive current flowing into the neuron and

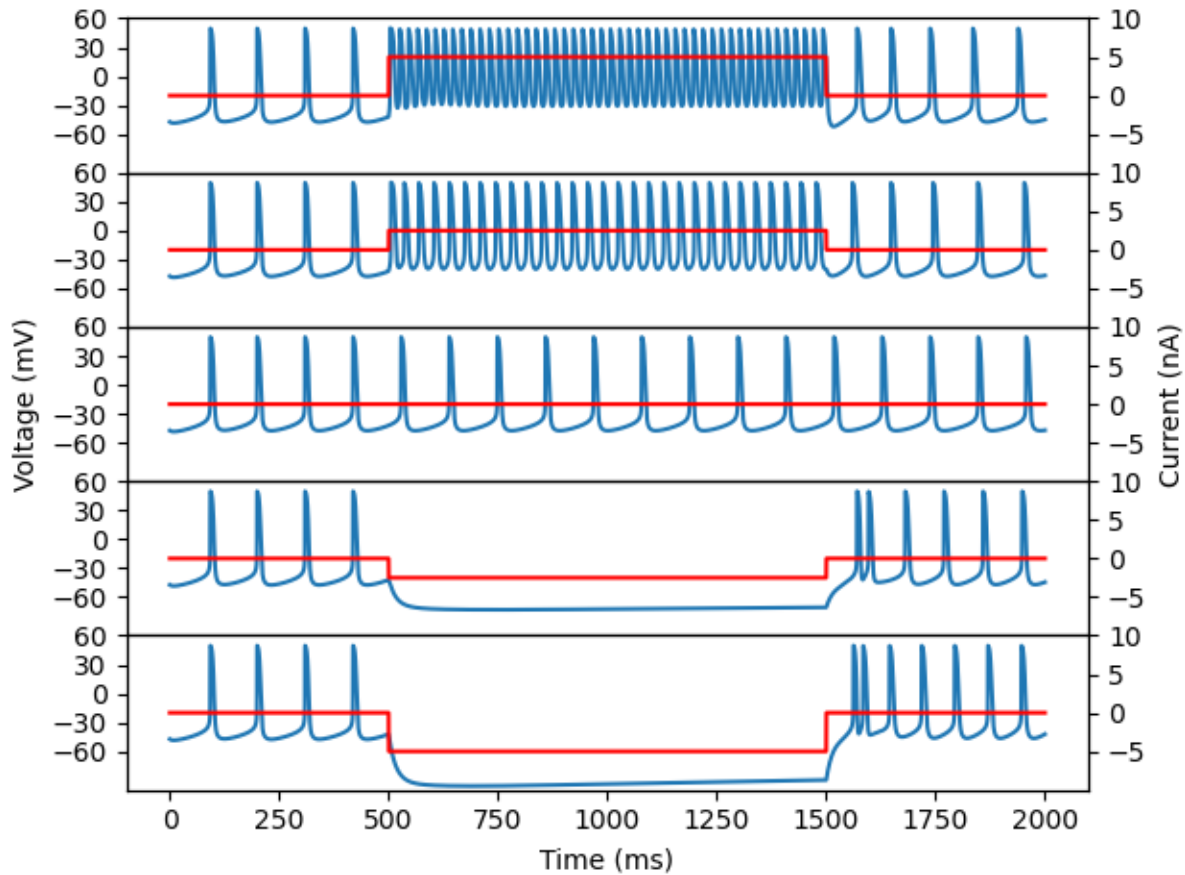


Figure 1 - Target Behavior

*The reactions of the voltage (blue) to different input currents (red). This is the behavior produced by the Julia model, and reflects closely the behavior shown in the paper by Buchholtz et al.*

negative flowing out. This allows us to study how the relative importance of parameters combinations depends on the magnitude of the input current.

There were several reasons for choosing the voltage spiking as the target behavior. First, it is most closely related to the primary physiological function of the cell, as it determines how it will fire when activated by an input current from another neuron. It is also the behavior most

thoroughly described by Buchholtz et al. in the original publication on the model. Further, it is the most complex behavior in the model since it depends on every parameter.

To determine the initial conditions of the model, we solved the dynamical system from generic initial conditions until it reached a limit cycle. We selected the values of the dynamical system from a point on this limit cycle as the initial conditions we use in our subsequent sloppy analysis. More specifically, we selected the values of the dynamical variables when the voltage was at a minimum of this limit cycle.

### **2.2.2 Julia Module**

The model was implemented as a Julia function (Appendix) that takes a set of parameter values, solves the corresponding system of differential equations, and returns the resulting voltage sampled for two thousand ms at 8.33 kHz for each of the six input currents. This implementation allows analysis of the model in parameter space in addition to solution space since the solution can be solved and sampled while varying the values of parameters. To ensure that parameters are only varied within their physical ranges we performed a parameter transformation. Parameters that are restricted to positive values were first log-transformed. A log-transform also removes dependence on units. For parameters that can take on any real value, we use an inverse sinh transform. The inverse sinh is log-like for large parameter values, so it also removes the effect of units.

## **2.3 Analysis Methods**

The core of the analysis method is to shift from a solution as a function of time to a solution as a function of parameters. This is facilitated by selecting the target behaviors,

described above, i.e., sampling voltage at predetermined time points. With this set of target behaviors, the model becomes a mapping from parameters to predictions. This view of the model as a mapping from parameters to solution enables us to study how the parameters affect the target behaviors and which combinations of parameters are the natural basis for controlling the target behaviors. The stiff parameter combinations are those that most strongly influence the solution and give the steepest gradients in parameter space.

### 2.3.1 Fisher Information

The behavior of the model in parameter space can be described by the Fisher information. This information is the matrix  $J^T J$  where  $J$  is the Jacobian matrix of the model function

$$J_{\mu\nu} = \frac{\partial y_\mu}{\partial \theta_\nu} \quad (33)$$

for  $\mathbf{y}$  being the time samples of the model and  $\boldsymbol{\theta}$  being the parameters. The Jacobian is therefore an  $M \times N$  matrix where  $M$  is the number of time samples and  $N$  the number of parameters.

The eigenvectors of the Fisher information matrix (FIM) form a basis in parameter space, with each vector being a combination of parameters. The eigenvalues give the relative importance of these combinations for controlling the target behavior, indicating their stiffness or sloppiness.

### 2.3.2 Singular Value Decomposition

To obtain this same information, the Jacobian of the model from the solution samples to the parameters was taken at each input current value. Next, the singular value decomposition (SVD) of the model was taken. The SVD separates the model into three parts as

$$J_{\mu\nu} = \sum_{\alpha} U_{\mu\alpha} S_{\alpha} V_{\nu\alpha} \quad (34)$$

where  $S$  is a diagonal matrix containing the singular values of  $J$  and  $\alpha$  are the singular values. These singular values are equal to the square roots of the eigenvalues of the FIM. The rows of  $V$  are called the right singular vectors and represent combinations of parameters or directions in parameter space, sharing this role with the eigenvectors of the FIM. The rows of  $U$  are the left singular vectors and are proportionate to the derivative of the solution with respect to the parameter combination in the corresponding right vector. The left singular vectors are additional information from the SVD over the FIM and reveal the roles of the new parameter combinations in the right vectors. The results drawn from studying these six SVDs are described below in Chapter 3.

## 3 Results and Conclusions

---

The analysis of the model showed that the Buchholtz model is indeed very sloppy. Only around a third as many parameters should be necessary to approximately describe the system. Future work may apply the Manifold Boundary Approximation Method (Mark K. Transtrum 2015) to explicitly construct a reduced-order model in terms of these parameters.

### 3.1 Singular Values

The squared singular values of the model, corresponding to the eigenvalues of the FIM, are shown in Figure 2. Each of these eigenvalues represents the stiffness of a particular combination of parameters as a value proportionate to the number of sample points squared, plotted as the purple dashed line. This number is the rough cutoff between stiff and sloppy directions. At the stiffest input current of 7.5 nA there are only 22 directions above this line. Theoretically then, the model could be refactored to create a remarkably close approximation using only about 22 new parameters.

An important feature of the following analysis of the left singular vectors is that the model was decomposed separately at each input current. This means that the highest eigenvalue for one current may not correspond to the same parameter combination as for another. Indeed, analysis of the right singular vectors shows that they do vary and that the stiffest parameters are therefore not the same at each current. These differences for each current are not just a mathematical quirk of the SVD, but rather indicate that different physical properties of the cell become more or less important to spiking depending on the input that the cell receives.

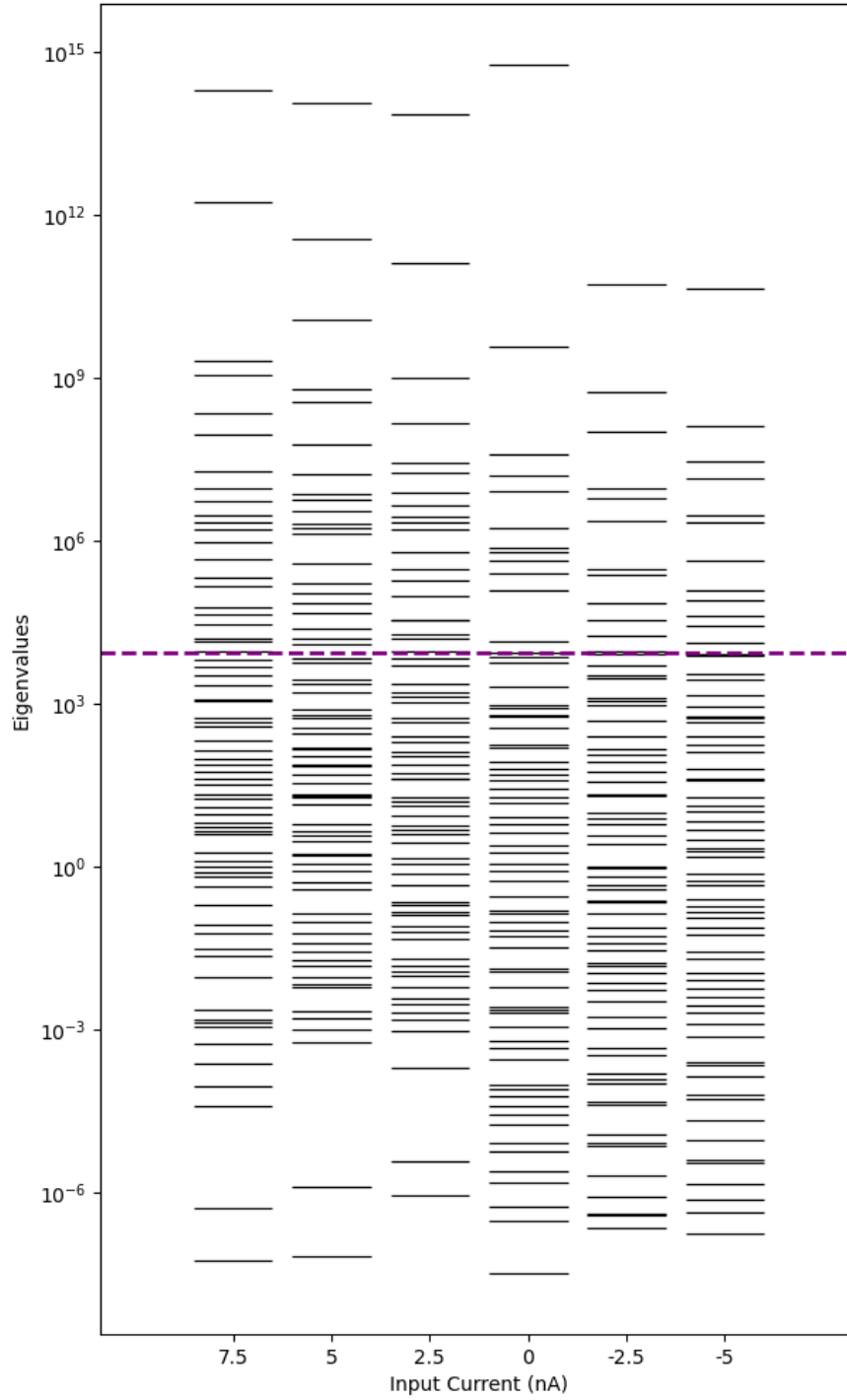


Figure 2 - Eigenvalues of FIM

*The dashed purple line indicates the rough cutoff between stiff and sloppy directions. Most directions fall under this line, in the sloppy region.*



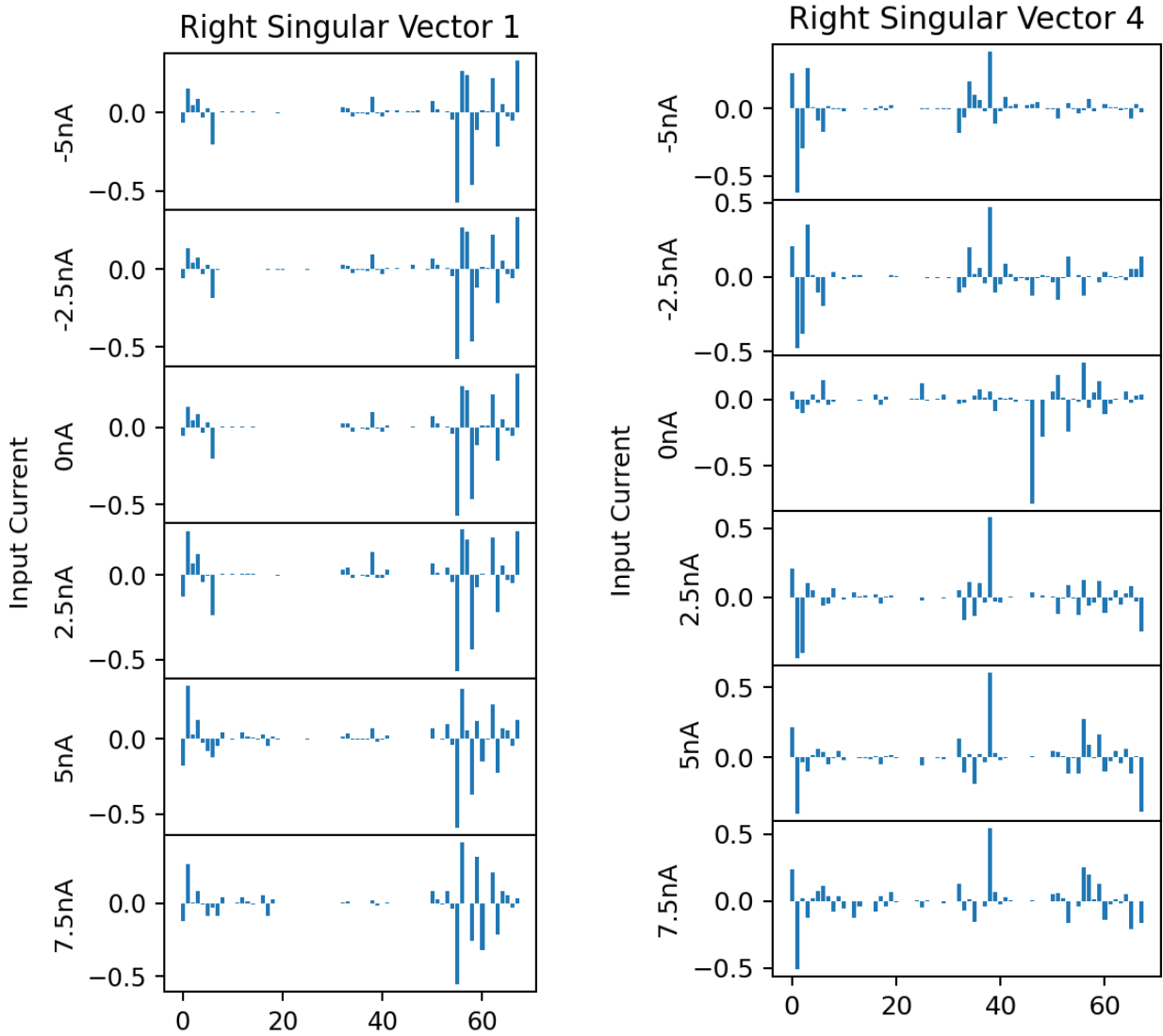


Figure 3- Right Singular Vectors

*These plots show the magnitude of each parameter in the right singular vectors for sample eigenvalues. The vectors correspond to the directions in parameter space for the first and fourth stiffest parameter combinations.*

*The first is roughly the same combination for each input current, but as early as the fourth vector the stiffest combinations vary and split into three distinct trends – one for negative current, one for positive and one in between them for zero.*

## 3.2 Singular Vectors

The singular vectors allow us to visualize how movement in parameter space affects the model. The left vector represents the strength of that point's model response to changes in parameters in the direction of the corresponding right singular vector. The singular value is the strength of the response in parameter space.

Figures 3 – 7 plot the model solution against

$$\mathbf{V}_{scaled} = 1\,000 * \mathbf{V} + \mathbf{y} \quad (35)$$

where  $\mathbf{V}$  is the left singular vector and  $\mathbf{y}$  is the solution. Peaks in the singular vector are places in the solution that would change the most if that combination of parameters were altered. The first few vectors shown here are the stiffest combinations corresponding to the largest singular values and give insight into how these emergent parameters describe the model.

In Figure 4 the first of the left singular vectors is shown. The increasing spikes at each peak indicate that this parameter combination controls the frequency of the solution: as the solution progresses in time, the phase difference from a change in frequency grows, and this manifests as growing peaks. Since the oscillation stops at

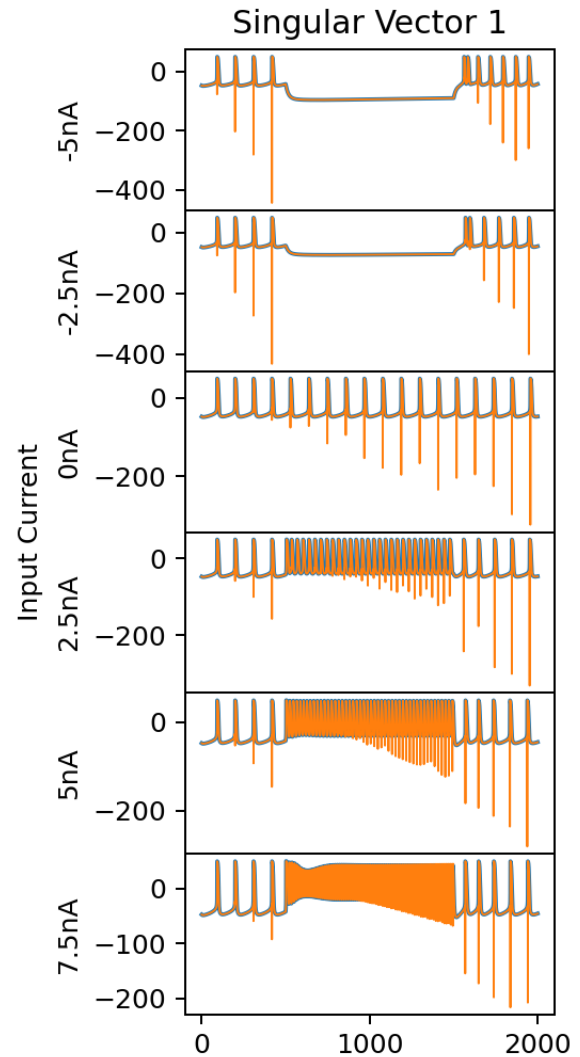


Figure 4 – Left Singular Vector 1

*The first left singular vector in orange and the original sampled values in blue*

*This vector controls the overall frequency.*

negative input currents, this offset disappears and only begins to grow again after the current is removed. For positive input currents the offset is able to build up during the faster oscillations and stays large even after the current is stopped. This vector has much less effect on the negative input currents because there are no oscillations for most of the solution time. This is reflected in the much lower first eigenvalue for these currents in Figure 2.

The second vector shown in Figure 5 is also related to frequency, as shown by the increasing magnitude of the spikes. This direction has a weaker effect during the rapid oscillations of positive input current and a stronger effect without the current, indicating that it primarily controls the resting frequency when the input current is zero.

The third vector in Figure 6 shows different effects for positive currents than negative currents. The stiffest parameter combinations are no longer the same, and the change in behavior here reflects that. For negative currents what is most important at this level is the reactivation after the current is removed, while for zero current it is the initial peak and for positive current it is the frequency before and after the current is applied. In the fourth vector, Figure 7, the differences are just as pronounced.

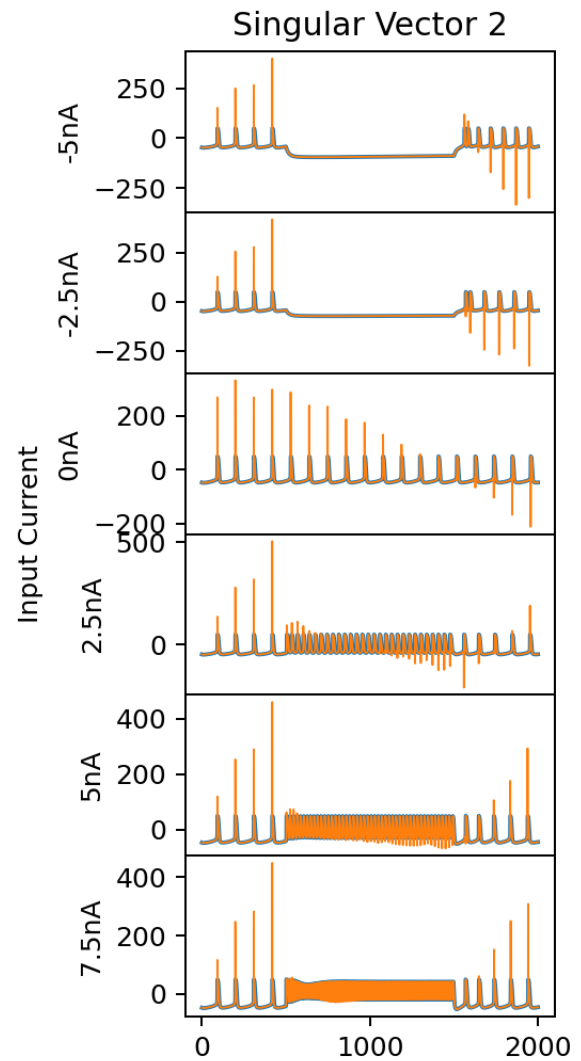


Figure 5 – Left Singular Vector 2

*This vector controls the frequency without input current.*

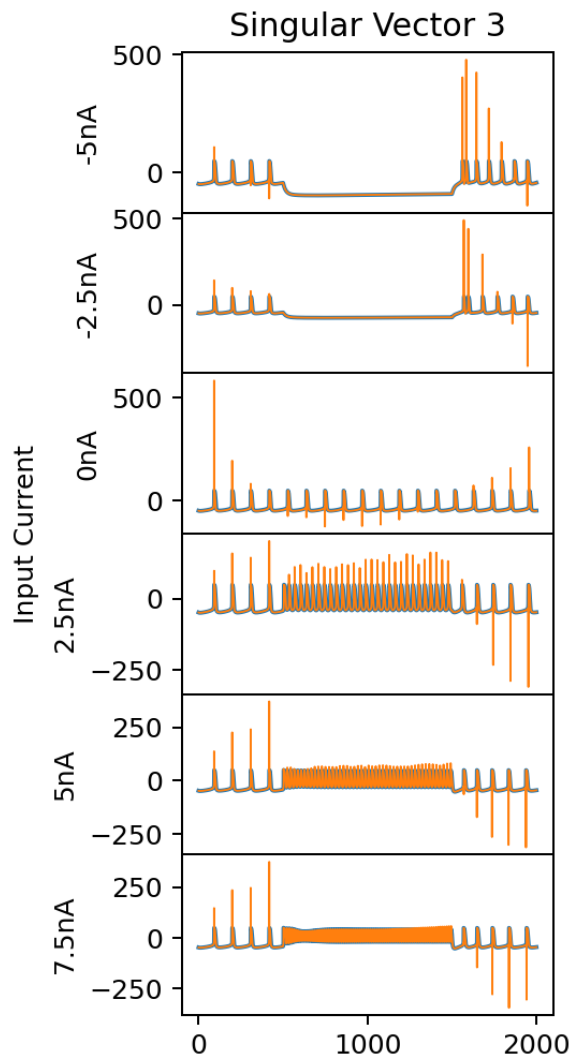


Figure 6 – Left Singular Vector 3

*This vector affects the amplitude of induced oscillations.*

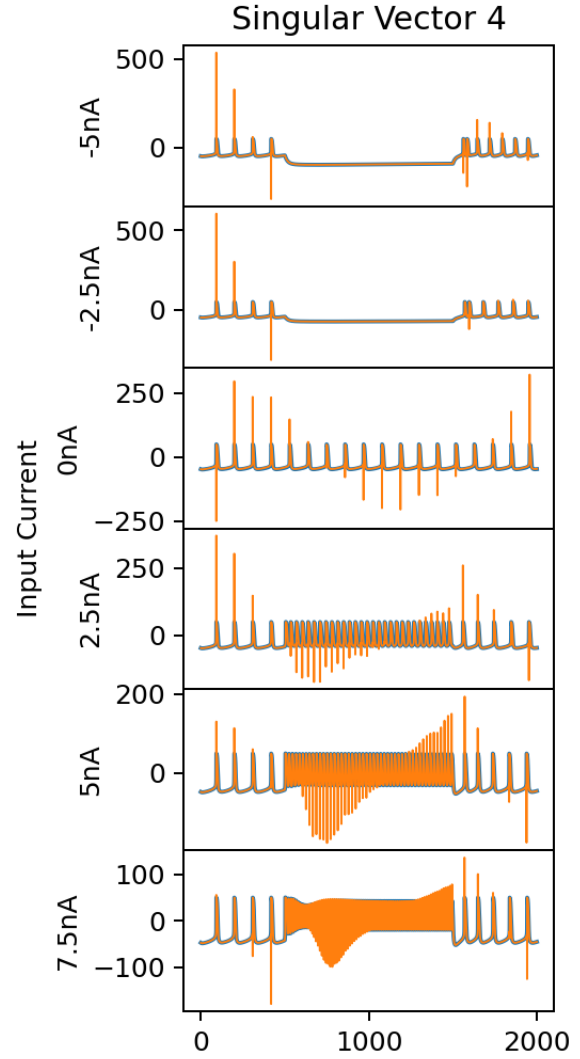


Figure 7 – Left Singular Vector 4

*This vector affects the frequency of induced oscillations.*

These stiffest directions' effects are clearly reflected in the behavior of the model – frequency responses, spiking during activation and deactivation, and total amplitude. These are traits that more naturally define the resulting model, though they do not have a clear relation to the individual currents and parameters (see the actual combinations in Figure 3 – they are very complex in terms of the original parameters of the model). Contrast the 68<sup>th</sup> singular vector - the

most sloppy combination - in Figure 8 where the effects are much smaller and are spread out across the entire solution. Most of the spikes are proportionate to the actual solution at that point. This combination has the weakest effect on the model's target behavior and does not correspond well to any particular defining trait of the solution.

It is clear from these analyses that the model exhibits high sloppiness but that there do exist certain parameter combinations that influence key properties of the solution voltage. These emergent parameters more clearly and simply describe the model. It is also clear that not all input currents are governed by the same set of these stiff parameters.

### 3.3 Future Work

The existence of these emergent properties in the singular vectors is encouraging as it demonstrates the possibility of simplifying the model to rely on these combinations of parameters to describe the model at a higher level. The task of redefining the model in terms of these new combinations and performing the reduction will require extensive future work using the manifold boundary approximation (Mark K. Transtrum 2015). This technique involves viewing as possible models as a manifold in n-dimensional parameter space. Since the manifold is thin along sloppy axes it

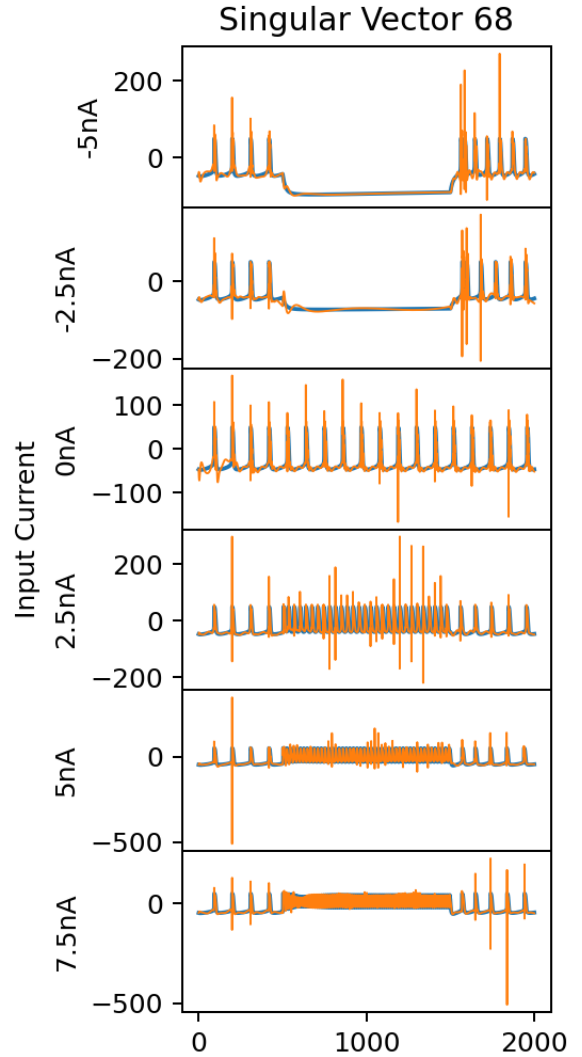


Figure 8 – Left Singular Vector 68

*This vector is included as an example of a sloppy parameter combination without a weaker and less clear effect.*

can be approximated by its boundary, removing that thin dimension, and fitting the resulting  $n-1$  dimensional model to the parameter again. Repetition of this process allows each sloppy dimension to be removed until a much stiffer approximation is reached. Julia code for constructing geodesics to find the manifold boundary and perform this process is available on the Transtrum group modeling repository (Transtrum 2021). Comparing the resultant model to previous work on reducing the model by other methods (David Golomb 1993) may show agreement on which parameters form a phenomenological basis.

The effects of the input current will likely be an important obstacle to overcome in performing this reduction, since the SVD yields different stiff parameters depending on the current. One potential solution could be parameterizing the input current rather than viewing it as constant and varying the input current through many values to get a single Jacobian rather than six. Another possible solution would be to sample the solution with a more complex function of input current, such as an input involving positive and negative gaussian pulses rather than a step function, so that the SVD would produce singular vectors that are stiff across the whole range of inputs.

Comparing the resulting model's parameters to the physical neuron should yield insights into what physical aspects of it are most important to its function. The traits that correspond to stiff parameters are likely the most strictly bound, while sloppier parameters would be able to vary widely and still effectively transmit action potentials. This could enhance understanding of why the neuron in this model is structured the way that it is.

With the goal of understanding physical effects of the stiff neuron attributes, future study of the model could also include parameterizing temperature and observing the effect on the

original and reduced models. Unfortunately, the data in the original paper from Buchholtz et al (F. Buchholtz 1992) was collected at 283 Kelvin and therefore the model will likely not represent the effects of temperature fluctuation. A different temperature-dependent version of the model would therefore be needed in order to treat temperature as anything but constant.

### **3.4 Conclusion**

The Buchholtz model exhibits high sloppiness, and the extra complexity from the sloppy parameters makes implementing the model cumbersome. While this sloppiness does not reduce the accuracy of the model, it does make the model difficult to understand and implement. Reducing sloppiness helps to improve understanding of the model, find emergent stiff parameter combinations and lower computation time when using it. Reducing this complexity will not only simplify computations of the model but will also help to identify the stiff parameter combinations that form a more natural basis for the model. This basis may also form a more natural means of describing other neurological models and help to advance our understanding of neurons.

# Appendix

---

## Julia Module

```
module Buchholtz_Model
import Models
using ParametricModels
import JSON

inputCurrent = 7.5

RelaxationRateUnit = 1/1000;
@parameterspace mutable struct Buchholtz
    R = 8.3145, constant
    z = 2, constant
    F = 96485.3364, constant
    T = 283, constant
    Cao = .05, constant
    V0 = -47.31873839514293, constant
    n0 = 0.7426884387628095, constant
    aA0 = 0.37055833789764714, constant
    bA10 = 0.026891249304544403, constant
    bA20 = 0.03974799915442218, constant
    aCa10 = 0.2716649180541391, constant
    aCa20 = 0.10131106656511829, constant
end
```



```

bCa10 = 0.24815067665480262, constant
Ca0 = 2.5565502265201756, constant
ao0 = 0.0015380660813823332, constant
bo0 = 0.44831495326463255, constant
r0 = 0.01713064244099982, constant
m0 = 0.015537113804810612, constant
h0 = 0.31037170208728354, constant
gd = .35, exp
EK = -80, sinh
cn = 180*RelaxationRateUnit, exp
Vn = -25, sinh
Vkn = 10, sinh1
sn = -17, sinh
skn = -22, sinh
gCa0 = 3.2, exp
koa = 600 * RelaxationRateUnit, exp
kob = 35 * RelaxationRateUnit, exp
kCa = 360 * RelaxationRateUnit, exp
Vao1 = 0, sinh
Vao2 = -16, sinh
sao1 = -23, sinh
sao2 = -5, sinh
f = .6, exp
c1 = 2.5, exp
c2 = .7, exp
c3 = .6, exp
ciCa = 300, exp

```

```

gA = 2.2, exp
kA = 140 * RelaxationRateUnit, exp
kA1 = 50 * RelaxationRateUnit, exp
cA2 = 3.6 * RelaxationRateUnit, exp
VA = -12, sinh
VB = -62, sinh
VA2 = -40, sinh
Vx = 7, sinh
sA = -26, sinh
sB = 6, sinh
sA2 = -12, sinh
sx = -15, sinh
gCa1 = .21, exp
gCa2 = .047, exp
kaCa1 = 50 * RelaxationRateUnit, exp
kbCa1 = 16 * RelaxationRateUnit, exp
kaCa2 = 10 * RelaxationRateUnit, exp
VaCa1 = -11, sinh
VbCa1 = -50, sinh
VaCa2 = 22, sinh
saCa1 = -7, sinh
sbCa1 = 8, sinh
saCa2 = -7, sinh
gh = .037, exp
Eh = -10, sinh
cr = .33 * RelaxationRateUnit, exp
Vr = -70, sinh

```

```

Vkr = -110, sinh
sr = 7, sinh
skr = -13, sinh
gNa = 2300, exp
ENa = 50, sinh
km = 10000 * RelaxationRateUnit, exp
kh = 500 * RelaxationRateUnit, exp
Vam = -6, sinh
Vbm = -34, sinh
Vah = -39, sinh
Vbh = -40, sinh
sam = -20, sinh
sbm = -13, sinh
sah = -8, sinh
sbh = -5, sinh
cam = .11, exp
cbm = 15, exp
cah = .08, exp
Cm = 1.7, exp
gl = .1, exp
El = -50, sinh
end

#Returns ICs
function ic(ps::Buchholtz{T}) where T <: Real
    return T[ps.V0, ps.n0, ps.ao0, ps.bo0, ps.Ca0, ps.aA0,
ps.bA10, ps.bA20, ps.aCa10, ps.aCa20, ps.bCa10, ps.r0, ps.m0,
ps.h0]

```

```

end

#Helper Functions
H(V,V2,s) = 1/(1+exp((V-V2)/s))

#External Current
Iext(t) = ((t > 500 && t < 1500) ? inputCurrent : 0)

#Actual diffeq
function rhs(ps::Buchholtz{TT}, t, u, du) where TT <: Real
    V,n,ao,bo,Ca,aA,bA1,bA2,aCa1,aCa2,bCa1,r,m,h = u

    #Physical constants

    R = 8.3145

    z = 2

    F = 96485.3364

    T = 283

    #Parameters

    gd = ps.gd

    EK = ps.EK

    cn = ps.cn

    Vn = ps.Vn

    Vkn = ps.Vkn

    sn = ps.sn

    skn = ps.skn

    gCaO = ps.gCaO

    koa = ps.koa

```

```
kob = ps.kob
kCa = ps.kCa
Vao1 = ps.Vao1
Vao2 = ps.Vao2
sao1 = ps.sao1
sao2 = ps.sao2
f = ps.f
c1 = ps.c1
c2 = ps.c2
c3 = ps.c3
ciCa = ps.ciCa
Cao = ps.Cao
gA = ps.gA
kA = ps.kA
kA1 = ps.kA1
cA2 = ps.cA2
VA = ps.VA
VB = ps.VB
VA2 = ps.VA2
Vx = ps.Vx
sA = ps.sA
sB = ps.sB
sA2 = ps.sA2
sx = ps.sx
gCa1 = ps.gCa1
gCa2 = ps.gCa2
kaCa1 = ps.kaCa1
```

```
kbCa1 = ps.kbCa1
kaCa2 = ps.kaCa2
VaCa1 = ps.VaCa1
VbCa1 = ps.VbCa1
VaCa2 = ps.VaCa2
saCa1 = ps.saCa1
sbCa1 = ps.sbCa1
saCa2 = ps.saCa2
gh = ps.gh
Eh = ps.Eh
cr = ps.cr
Vr = ps.Vr
Vkr = ps.Vkr
sr = ps.sr
skr = ps.skr
gNa = ps.gNa
ENa = ps.ENa
km = ps.km
kh = ps.kh
Vam = ps.Vam
Vbm = ps.Vbm
Vah = ps.Vah
Vbh = ps.Vbh
sam = ps.sam
sbm = ps.sbm
sah = ps.sah
sbh = ps.sbh
```

```

cam = ps.cam
cbm = ps.cbm
cah = ps.cah
Cm = ps.Cm
gl = ps.gl
El = ps.El

# Substitutions
kn = cn * H(V,Vkn,skn)
nf = H(V,Vn,sn)
IKd = gd * n^4 * (V - EK)
aof = H(V,Vao1-f*Ca,sao1) * H(V,Vao2-f*Ca,sao2) * (Ca / (c1
+ Ca))
bof = c2/(c3 + Ca)
ICaO = gCaO * ao * bo * (V - EK)
kA2 = cA2 * H(V, VA2, sA2)
aAf = H(V,VA,sA)
bAf = H(V,VB,sB)
bA = H(V,Vx,sx) * bA1 + (1 - H(V,Vx,sx)) * bA2
IA = gA * aA^3 * bA * (V - EK)
ECa = (R * T) / (z * F) * log(max((13000/Ca),0)) * 1000
aCa1f = H(V,VaCa1,saCa1)
aCa2f = H(V,VaCa2,saCa2)
bCa1f = H(V,VbCa1,sbCa1)
ICa = (gCa1 * aCa1 * bCa1 + gCa2 * aCa2) * (V - ECa)
rf = H(V,Vr,sr)
kr = cr / H(V,Vkr,skr)
Ih = gh * r * (V - Eh)

```

```

am = cam * (V - Vam) / (1 - exp((V - Vam)/sam))
bm = cbm * exp((V - Vbm)/sbm)
mf = am/(am + bm)
ah = cah * exp((V - Vah)/sah)
bh = H(V,Vbh,sbh)
hf = ah/(ah + bh)
INa = gNa * m^3 * h * (V - ENa)
Il = gl * (V - El)

#Voltage
du[1] = (Iext(t) - IKd - ICaO - IA - Ica - Ih - INa - Il)/Cm

#Delayed Rectifier Pottasium Current IKd
du[2] = (nf - n) * kn

#Calcium Activated Outward Pottasium Current ICaO
du[3] = (aof - ao) * koa
du[4] = (bof - bo) * kob
du[5] = -ciCa * Ica * RelaxationRateUnit - kCa * (Ca - Cao)

#Transient A-like Current IA
du[6] = (aAf - aA) * kA
du[7] = (bAf - bA1) * kA1
du[8] = (bAf - bA2) * kA2

#Calcium Current Ica
du[9] = (aCa1f - aCa1) * kaCa1
du[10] = (aCa2f - aCa2) * kaCa2
du[11] = (bCa1f - bCa1) * kbCa1

#Inwardly Rectifying Potassium Current Ih

```



```

    du[12] = (rf - r) * kr
    #Fast Tetrodotoxin-Sensitive Sodium INa
    du[13] = (mf - m) * km
    du[14] = (hf - h) * kh
    nothing
end
#Returns voltage from the solved problem
function obs(ps::Buchholtz{T}, _t, _x) where T <: Real
    return T[_x[1]]
end
function obs_states(ps::Buchholtz{T}, _t, _x) where T <: Real
    return _x
end
import Sundials
import DifferentialEquations
alg = Sundials.CVODE_BDF()
BuchholtzVoltage(ps::Buchholtz, t) =
ParametricModels.solve_ode(ps, ic, rhs, obs, t, alg, abstol= 1e-
8, reltol = 1e-8, stops = [500, 1500])
dt0 = 5.0
dt75 = .12
t = 0:dt75:2000
y = BuchholtzVoltage(Buchholtz(), t)
data = ParametricModels.OLSData("Buchholtz",
ParametricModels.ModelArgs(t), y)
pmodel = PModel(Buchholtz, parameter_transforms,
BuchholtzVoltage, data)
model = Models.Model(pmodel, "Buchholtz")
modelbare = Models.Model(pmodel, "Buchholtzbare"; bare = true)

```

```

xi = ParametricModels.xvalues(pmodel)

function setInputCurrent(current)
    global inputCurrent = current
    global y = BuchholtzVoltage(Buchholtz(), t)
    global data = ParametricModels.OLSData("Buchholtz",
ParametricModels.ModelArgs(t), y)
    global pmodel = PModel(Buchholtz, parameter_transforms,
BuchholtzVoltage, data)
    global model = Models.Model(pmodel, "Buchholtz")
    global modelbare = Models.Model(pmodel, "Buchholtzbare";
bare = true)
    global xi = ParametricModels.xvalues(pmodel)
    return y, xi
end

end # module

```

# Index

---

Buchholtz model, 5, 15

Emergence, 2

Model reduction, 4, 21

Neuron, 3, 5

Singular value decomposition, 13

Sloppiness, 2, 13, 21

# References

---

- Guy-Evans, O. (2021, Feb 15). What is a neuron? Function, parts, structure, and types. Simply Psychology. <https://www.simplypsychology.org/neuron.html>
- A. L. Huxley, A. F. Hodgkin. 1952. "A quantitative description of membrane current and its application to conduction and excitation in nerve." *The Journal of Physiology*.
- David Golomb, John Guckenheimer, Shay Gueron. 1993. "Reduction of a channel-based model for a stomatogastric." *Biological Cybernetics*.
- F. Buchholtz, J. Golowasch, I. R. Epstein, E. Marder. 1992. "Mathematical model of an identified stomatogastric ganglion neuron." *Journal of Neurophysiology*.
- Guy-Evans, O. 2021. "What is a Neuron? Function, Parts, Structure and Types." *Simply Psychology*. February 15. Accessed 4 21, 2021.  
<https://www.simplypsychology.org/neuron.html>.
- Helen C. Lai, Lily Y. Jan. 2006. "The distribution and targeting of neuronal voltage-gated ion channels." *Nature Reviews Neuroscience* 7: 548-562.
- Mark K. Transtrum, Benjamin B. Machta, Kevin S. Brown, Bryan C. Daniels, Christopher R. Myers, and James P. Sethna. 2015. "Perspective: Sloppiness and emergent theories in physics, biology, and beyond." *The Journal of Chemical Physics*.

Rasband, Joshua. 2021. *Two Reduced Models of Nerve Behavior*. Senior Thesis, Brigham Young University.

Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, James P Sethna. 2007. "Universally Sloppy Parameter Sensitivities in Systems Biology Models." *PLoS Computational Biology*.

Transtrum, Mark K. 2021. *Transtrum Group Modeling Library*. Accessed 4 2021.  
<https://git.physics.byu.edu/Modeling>.