Periodicities in the Hodgkin-Huxley Model and Versions of This Model with Stochastic Input

by
Kevin Endler

A thesis submitted in partial fulfillment of the requirements for the degree
of
Master of Sciences
Department of Physics, Mathematics, and Computer Science
Johannes Gutenberg University Mainz

Mainz, February 2012

Abstract

A field of computational neuroscience develops mathematical models to describe neuronal systems. The aim is to better understand the nervous system. Historically, the *integrate-and-fire model*, developed by Lapique in 1907, was the first model describing a neuron. In 1952 Hodgkin and Huxley [8] described the so called *Hodgkin-Huxley model* in the article "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve". The Hodgkin-Huxley model is one of the most successful and widely-used biological neuron models. Based on experimental data from the squid giant axon, Hodgkin and Huxley developed their mathematical model as a four-dimensional system of first-order ordinary differential equations. One of these equations characterizes the membrane potential as a process in time, whereas the other three equations depict the opening and closing state of sodium and potassium ion channels. The membrane potential is proportional to the sum of ionic current flowing across the membrane and an externally applied current. For various types of external input the membrane potential behaves differently. This thesis considers the following three types of input:

(i) Rinzel and Miller [15] calculated an interval of amplitudes for a constant applied current, where the membrane potential is repetitively spiking.

(ii) Aihara, Matsumoto and Ikegaya [1] said that dependent on the amplitude and the frequency of a periodic applied current the membrane potential responds periodically.

(iii) Izhikevich [12] stated that brief pulses of positive and negative current with different amplitudes and frequencies can lead to a periodic response of the membrane potential.

In chapter 1 the Hodgkin-Huxley model is introduced according to Izhikevich [12]. Besides the definition of the model, several biological and physiological notes are made, and further concepts are described by examples. Moreover, the numerical methods to solve the equations of the Hodgkin-Huxley model are presented which were used for the computer simulations in chapter 2 and chapter 3. In chapter 2 the statements for the three different inputs (i), (ii) and (iii) will be verified, and periodic behavior for the inputs (ii) and (iii) will be investigated. In chapter 3 the inputs are embedded in an Ornstein-Uhlenbeck process to see the influence of noise on the results of chapter 2.

Zusammenfassung

Ein Bereich der Computational Neuroscience beschäftigt sich mit der mathematischen Modellierung neuronaler Systeme. Das Ziel ist es, ein besseres Verständnis des Nervensystems zu erhalten. Historisch gesehen ist das *Integrate-and-Fire Modell*, entwickelt von Lapique im Jahre 1907, das erste Modell, das ein Neuron beschrieben hat. Im Jahre 1952 beschreiben Hodgkin und Huxley [8] in ihrer Arbeit "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve" eines der erfolgreichsten und am meisten genutzten biologischen Neuronenmodelle, das sogenannte *Hodgkin-Huxley Modell*. Auf der Grundlage von Experimenten am Riesenaxon des Tintenfischs wurde das Hodgkin-Huxley Modell als vierdimensionales System gewöhnlicher Differentialgleichungen erster Ordnung entwickelt. Eine dieser Gleichungen stellt das Membranpotential dar, wohingegen die restlichen drei Gleichungen das Öffnen und Schließen der Natrium- und Kaliumionenkanäle wiedergibt. Dabei ist das Membranpotential proportional zur Summe der Ionenströme und einem externen Strominput. Für verschiedene Typen von externem Input reagiert das Membranpotential unterschiedlich. In dieser Arbeit beschäftigen wir uns mit den folgenden drei Typen von Input:

(i) Rinzel und Miller [15] berechneten numerisch, dass für einen konstanten Input ein Intervall von Amplituden existiert, für welches das Membranpotential mit einer gewissen Frequenz wiederholend Aktionspotenziale erzeugt.

(ii) Aihara, Matsumoto und Ikegaya [1] beschreiben, dass abhängig von der Amplitude und der Frequenz eines periodischen Inputs das Membranpotential periodisch zum Input erwidert.

(iii) Izhikevich [12] erklärt, dass kurze Impulse von positivem, sowie negativem Strom mit unterschiedlichen Amplituden und Frequenzen zu periodischem Verhalten des Membranpotentials führen kann.

In Kapitel 1 wird eine Einführung in das Hodgkin-Huxley Modell gegeben, analog zu Izhikevich [12]. Dabei werden neben der Definition des Modells einige Bemerkungen über den biologischen und physiologischen Hintergrund gegeben und weitere Begriffe anhand von Beispielen eingeführt. Außerdem werden die numerischen Methoden angegeben, die zur Lösung der Gleichungen des Hodgkin-Huxley Modells mit Computersimulationen in Kapitel 2 und Kapitel 3 verwendet werden. In Kapitel 2 werden die Aussagen über die drei verschiedenen Inputtypen (i), (ii) und (iii) verifiziert. Außerdem werden die Periodizitäten in (ii) und (iii) genauer untersucht. In Kapitel 3 werden die Inputtypen in einen Ornstein-Uhlenbeck Prozess eingebettet, um den Einfluss von Rauschen auf die Ergebnisse aus Kapitel 2 zu untersuchen.

Contents

List of Figures

List of Tables

Chapter 1

The Hodgkin-Huxley Model

This chapter is divided into two sections. The first section is a mathematical summary of the Hodgkin-Huxley model based on Izhikevich's book [12]. Izhikevich [12] is more modern than the pioneering article [8] of Hodgkin and Huxley describing the conductances responsible for the generation of action potentials in the squid giant axon. This section hardly touches on the biological and physiological background. In the second section the methods used for the numerical simulations are provided.

1.1 Mathematical Description

1.1 Definition: (i) The functions $\alpha_n, \alpha_m, \alpha_h, \beta_n, \beta_m \colon \mathbb{R} \to \mathbb{R}_+$ and $\beta_h \colon \mathbb{R} \to (0,1)$ defined as

$$\begin{aligned}
\alpha_n(V) &:= \begin{cases} \frac{10-V}{100\left(\exp\left(\frac{10-V}{10}\right)-1\right)} & \text{, if } V \neq 10 \\ \frac{1}{10} & \text{, else} \end{cases}, \qquad \beta_n(V) := \frac{1}{8}\exp\left(\frac{-V}{80}\right), \\
\alpha_m(V) &:= \begin{cases} \frac{25-V}{10\left(\exp\left(\frac{25-V}{10}\right)-1\right)} & \text{, if } V \neq 25 \\ 1 & \text{, else} \end{cases}, \qquad \beta_m(V) := 4\exp\left(\frac{-V}{18}\right), \\
\alpha_h(V) &:= \frac{7}{100}\exp\left(\frac{-V}{20}\right), \qquad \beta_h(V) := \frac{1}{\exp\left(\frac{30-V}{10}\right)+1}
\end{aligned}$$

are called the *rating functions*.

(ii) Let

$$E_K = -12, \quad E_{Na} = 120, \quad E_L = 10.6 \tag{1.1}$$

be the *equilibrium potentials*, $\bar{g}_K = 36$, $\bar{g}_{Na} = 120$, $\bar{g}_L = 0.3$ be the *maximum conductances* and $C = 1$ be the *membrane capacity*. Further let

$$I \colon \mathbb{R}_+ \to \mathbb{R},\ t \mapsto I(t) \tag{1.2}$$

be a given *input function*. The *Hodgkin-Huxley model* consists of a four-dimensional system of first-order ordinary differential equations

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I(t) - \bar{g}_K n^4 \cdot (V - E_K) - \bar{g}_{Na} m^3 h \cdot (V - E_{Na}) - \bar{g}_L \cdot (V - E_L) \tag{1.3}$$

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V) \cdot (1 - n) - \beta_n(V) \cdot n \tag{1.4}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m(V)\cdot(1-m) - \beta_m(V)\cdot m \tag{1.5}$$
$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h(V)\cdot(1-h) - \beta_h(V)\cdot h. \tag{1.6}$$

The process $(V(t))_{t\geq 0}$ is called the *membrane potential*, the functions $n(t)$, $m(t)$ and $h(t)$ are called the *gating variables*.

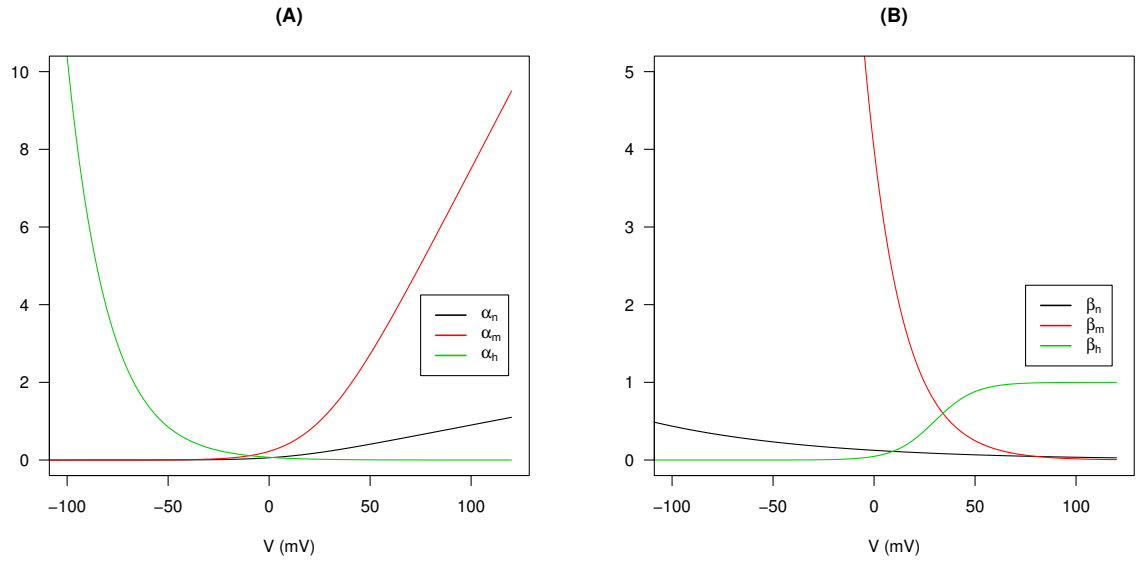


Figure 1.1: Rating functions α_n, α_m, α_h **(A)** and β_n, β_m, β_h in **(B)** for $V \in [-120, 120]$.

1.2 Proposition: The rating functions $\alpha_n, \alpha_m, \alpha_h, \beta_n, \beta_m, \beta_h$ are analytic, i.e. the functions are locally given by a convergent power series. Especially the rating functions are infinitely differentiable (see [3], 2.3 Definition on page 34 and 2.5 Proposition on page 35).

Proof Let

$$f(z) \;:=\; \begin{cases} \frac{z}{\exp(z)-1} & \text{, if } z \neq 0 \\ 1 & \text{, else.} \end{cases}$$

There is a neighborhood of 0, so that f can be expanded into a power series of z:

$$f(z) = \sum_{k=0}^{\infty} B_k \frac{z^k}{k!},$$

where B_k are the Bernoulli numbers (see [14], page 289). To proof that f is analytic, consider

$$g(z) := \frac{1}{f(z)} = \frac{\exp(z)-1}{z}.$$

Also g can be expanded into a power series of z in a neighborhood of 0:

$$g(z) = \sum_{k=0}^{\infty} \frac{z^k}{(k+1)!}.$$

Further, g is absolutely convergent for all $z \in \mathbb{R}$ using d'Alembert's ratio test (see [14], page 65):

$$\lim_{n\to\infty} \left| \frac{\frac{z^{n+2}}{(n+2)!}}{\frac{z^{n+1}}{(n+1)!}} \right| = \lim_{n\to\infty} \frac{z}{n+2} = 0 \quad \forall z \in \mathbb{R}.$$

Now conclusion 1 in [14] on page 287 is used: *Let* $p(z) = \sum_{n=0}^{\infty} a_n z^n$ *with a positive radius of convergence. If* $p(0) = a_0 \neq 0$*, then* $\frac{1}{p}$ *can be expanded into a power series of* z *in a neighborhood of* 0*:*

$$\frac{1}{p(z)} = \sum_{n=0}^{\infty} b_n z^n.$$

In this case $p = g$ and $\frac{1}{p} = f$. Therefore f is analytic. Defining

$$\omega_1(V) := \frac{10 - V}{10} \text{ and } \omega_2(V) := \frac{25 - V}{10},$$

which are analytic for all $V \in \mathbb{R}$, it follows that the compositions (see [3], 2.4 Chain Rule on page 34)

$$\tilde{\alpha}_n = f \circ \omega_1 \text{ and } \alpha_m = f \circ \omega_2$$

are analytic for all $V \in \mathbb{R}$ and finally

$$\alpha_n = \frac{1}{10}\tilde{\alpha}_n$$

is analytic for all $V \in \mathbb{R}$.
β_h is analytic for all $V \in \mathbb{R}$, because the denominator $\exp(\frac{30-V}{10}) + 1$ is analytic for all $V \in \mathbb{R}$. In the end β_n, β_m and α_h are analytic for all $V \in \mathbb{R}$, because the exponential functions are analytic for all $V \in \mathbb{R}$. □

1.3 Remark: (i) The units for the membrane potential (mV), for the time (ms), for the equilibrium potentials (mV), for the maximum conductances (mS/cm^2), for the membrane capacity (μF/cm^2), and for the input function (μA/cm^2) are not always mentioned in the thesis. With regard to the biological point of view, the units are of minor importance for the mathematical approach.

(ii) The functions of 1.1 Definition (i) were established by Alan L. Hodgkin and Andrew F. Huxley, and were published in their pioneering article [8]. To this day these functions have not been changed in further examinations besides shifting by the resting potential V_{rest} ($V \mapsto V + V_{rest}$). Hodgkin and Huxley [8] named these functions the rating constants, whereas in this thesis they are called the rating functions because they are not constant. Since the rating functions correspond to the membrane potential shifted by 65 mV, the resting potential is at $V_{rest} \approx 0$. They describe the transition rates between open and closed states of the channels, according to Izhikevich [12]. Hodgkin and Huxley [8] determined that the squid axon carries three types of ionic currents:

sodium (Na^+) current, $I_{Na} = \bar{g}_{Na} m^3 h \cdot (V - E_{Na})$, with three activation gates and one inactivation gate (the term $m^3 h$, where m and h are probabilities between 0 and 1 describing the state of Na^+ activation and Na^+ inactivation, respectively), potassium (K^+) current, $I_K = \bar{g}_K n^4 \cdot (V - E_K)$, with four activation gates (the term n^4, where n is a probability between 0 and 1 describing the state of K^+ activation), and current that Hodgkin and Huxley [8] named leakage current, $I_L = \bar{g}_L \cdot (V - E_L)$, which consists of mostly chloride (Cl^-) ions.

(iii) The values given in 1.1 Definition (ii) are the values according to Izhikevich [12]. Hodgkin and Huxley [8] used different values for the equilibrium potentials:

$$E_K = -12, \quad E_{Na} = 115, \quad E_L = 10.613. \tag{1.7}$$

Note that the maximum conductances chosen by Hodgkin and Huxley (see [8], table 3) are still up to date in further examinations.

(iv) Throughout the thesis equations (1.4), (1.5) and (1.6) are used in the standard form

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \frac{n_\infty(V) - n}{\tau_n(V)}, \tag{1.8}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{m_\infty(V) - m}{\tau_m(V)}, \tag{1.9}$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{h_\infty(V) - h}{\tau_h(V)}, \tag{1.10}$$

where

$$\tau_n = \frac{1}{\alpha_n + \beta_n}, \quad n_\infty = \frac{\alpha_n}{\alpha_n + \beta_n},$$
$$\tau_m = \frac{1}{\alpha_m + \beta_m}, \quad m_\infty = \frac{\alpha_m}{\alpha_m + \beta_m},$$
$$\tau_h = \frac{1}{\alpha_h + \beta_h}, \quad h_\infty = \frac{\alpha_h}{\alpha_h + \beta_h}.$$

τ_n, τ_m, τ_h are called *time functions* (Hodgkin and Huxley [8] called them the time constants) and n_∞, m_∞, h_∞ are called the *steady-state (in)activation functions* (see figure 1.2). Due to 1.2 Proposition the time functions and steady-state (in)activation functions are analytic.

(v) The equilibrium potentials can be calculated, and they are given by the Nernst equation (see [12], equation (2.1)):

$$E_{ion} = \frac{RT}{zF} \ln\left(\frac{[ion]_{out}}{[ion]_{in}}\right), \tag{1.11}$$

where $[ion]_{in}$ and $[ion]_{out}$ are concentrations of the ions inside and outside the cell respectively, $R = 8,315$ mJ/Kmol is the universal gas constant, T is temperature in Kelvin (K $= 273.16 +$ °C), $F = 96,480$ C/mol is Faraday's constant and z is the valence of the ion ($z = 1$ for Na^+, K^+ and $z = -1$ for Cl^-). The equilibrium potentials (1.1) can be obtained by using the concentrations of the ions in table 1.1 for $T = 20$ °C and shifting the obtained Nernst equilibrium potentials by approximately 65 mV (see [12], Chapter 2, exercise 1).

	Inside (mM)	Outside (mM)
K^+	430	20
Na^+	50	440
Cl^-	65	560

Table 1.1: These are the concentrations of the ions inside and outside the cell which Izhikevich used to obtain the equilibrium potentials by the Nernst equation (1.11) (see [12], Chapter 2, exercise 1).

Another way to obtain the equilibrium potentials is by experimenting, and comparing the results to real data. Hodgkin and Huxley [8] did that for E_{Na} and E_K. The equilibrium potential E_L is the exact value chosen to make the total ionic current zero at the resting potential $V_{rest} = 0$, i.e.

$$E_L = \frac{-\bar{g}_K \cdot n_\infty^4(0) \cdot E_K - \bar{g}_{Na} \cdot m_\infty^3(0) \cdot h_\infty(0) \cdot E_{Na}}{\bar{g}_L}.$$

This is $E_L = 10.5989$ which is slightly different from the value of Hodgkin and Huxley ($E_L = 10.613$ in (1.7) or see [8], table 3).

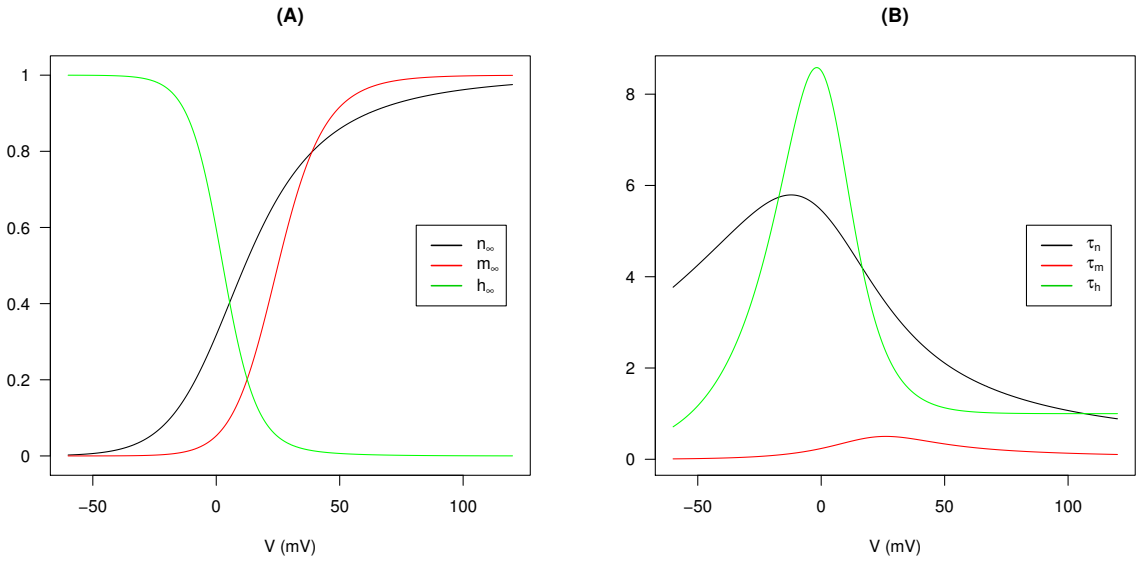


Figure 1.2: Steady-state (in)activation functions n_∞, m_∞, h_∞ **(A)** and time functions τ_n, τ_m, τ_h **(B)** for $V \in [-60, 120]$.

1.4 Example: For the input function

$$I(t) = 5 \cdot \mathbb{1}_{[2,3]}(t) + 15 \cdot \mathbb{1}_{[10,11]}(t), \quad t \geq 0,$$

a typical time course of an action potential in the Hodgkin-Huxley model can be observed in figure 1.3 using the equilibrium potentials (1.1) (see [12], figure 2.15 using 0.5 ms pulses of current). For $t = 13.38$ the membrane potential $V(t)$ has a relative maximum of 107.0419. Such a relative maximum is named an *action potential* or *spike*.

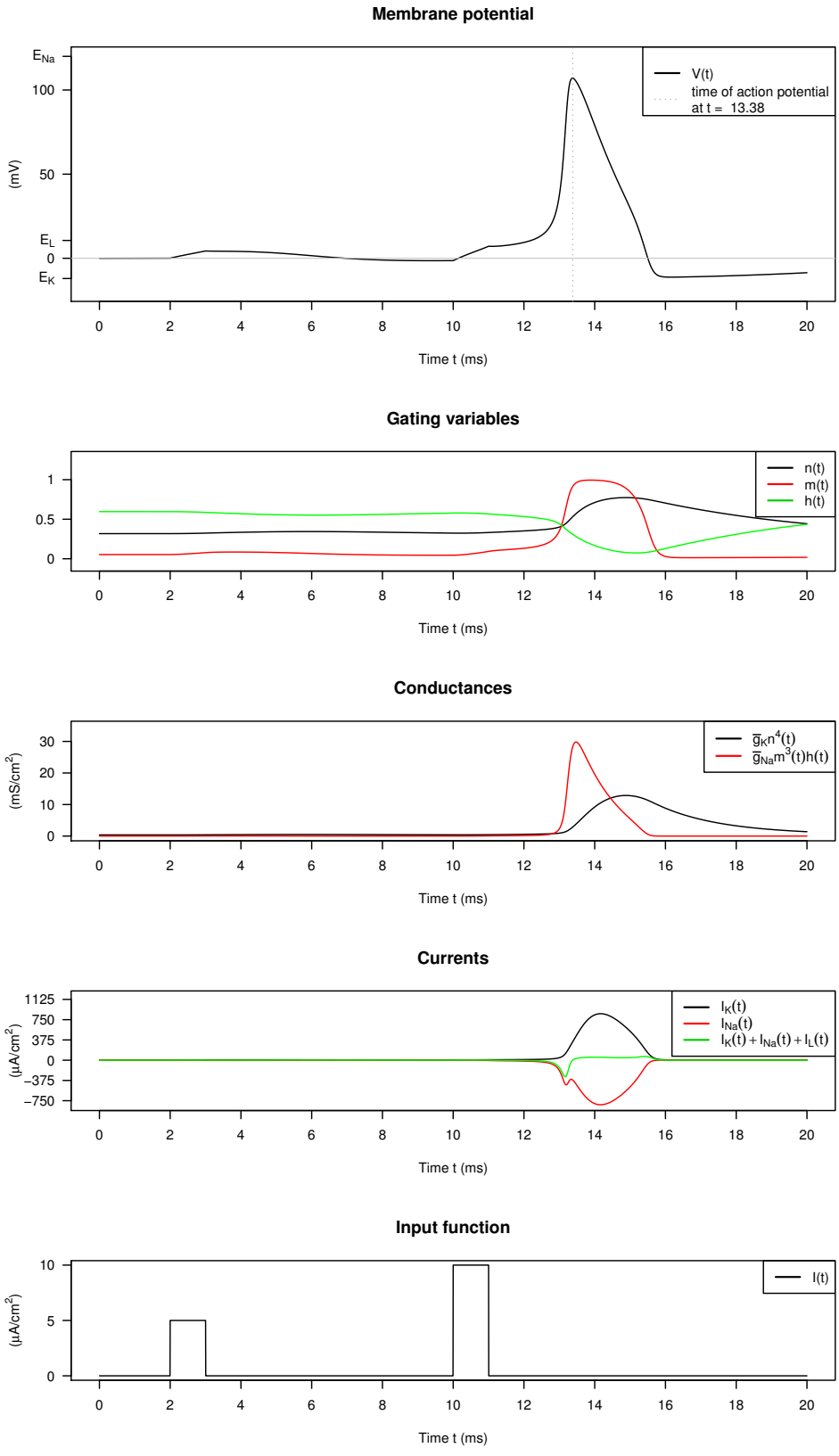


Figure 1.3: Action potential in the Hodgkin-Huxley model

1.5 Example: The Hodgkin-Huxley model was stimulated with 1 ms pulses of current. For

$$I(t) = a \cdot \mathbb{1}_{[1,2]}(t), \quad t \geq 0,$$

the threshold to generate a spike is $a = 6.41$ when $t = 7.93$ (see figure 1.4). The period during which the Hodgkin-Huxley system cannot initiate another spike is called the *absolute refractory*. Further, the period during which the Hodgkin-Huxley system is able to generate an action potential - if the applied current is strong enough - is called the *relative refractory*. To analyze these periods a second action potential was generated varying the latency for the initialization of a second pulse. For these latencies the minimum pulse amplitude a_p can be calculated which is needed to evoke a second spike (see figure 1.5):

$$I(t) = 6.41 \cdot \mathbb{1}_{[1,2]}(t) + a_p \cdot \mathbb{1}_{[t_p, t_p+1]}(t), \quad t \geq 0,$$

where $t_p > 7.93$. For $a_p = 6.41$ the time $t_p = 22.02$ is the minimum to induce a second spike at $t = 29.715$. There is no second spike, if $I(t) = 6.41$ for $t \in [1, 2] \cup [22.01, 23.01]$, although the initialization times differ only by 0.01 (see figure 1.6). Here, the minimum time to initialize a second action potential is 14.09 ms after the first spike for pulses with the same amplitude of 6.41.

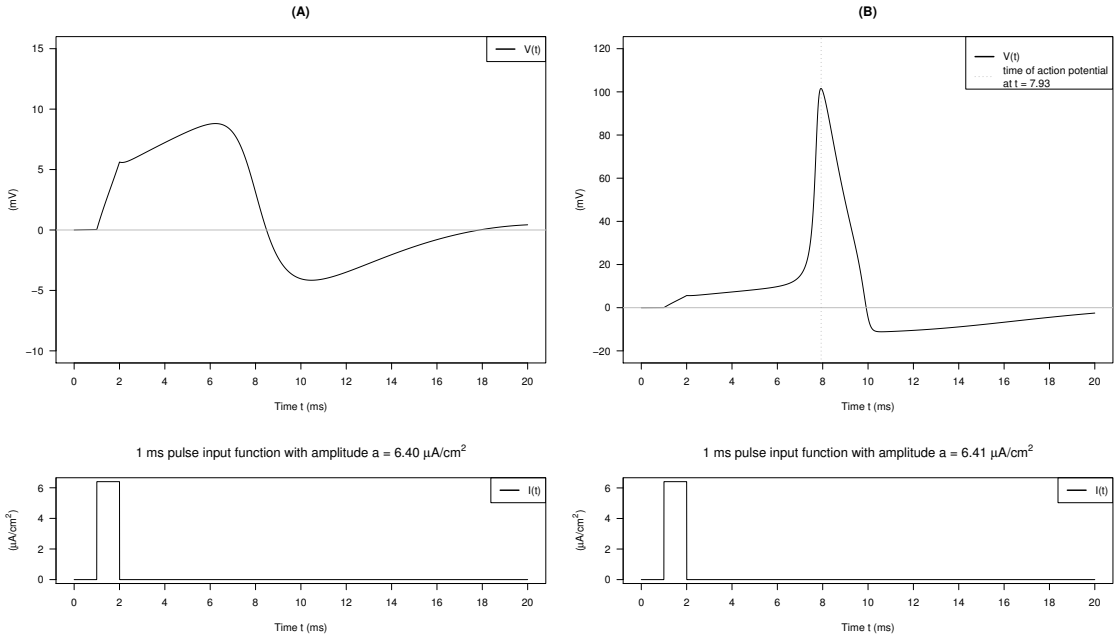


Figure 1.4: Generation of the first action potential for a 1 ms pulse initialized at time $t = 1$ with amplitude $a = 6.40$ **(A)** and $a = 6.41$ **(B)**. In **(B)** a spike can be observed at time $t = 7.93$, whereas in **(A)** no action potential is generated, although the input amplitudes differ only by 0.01.

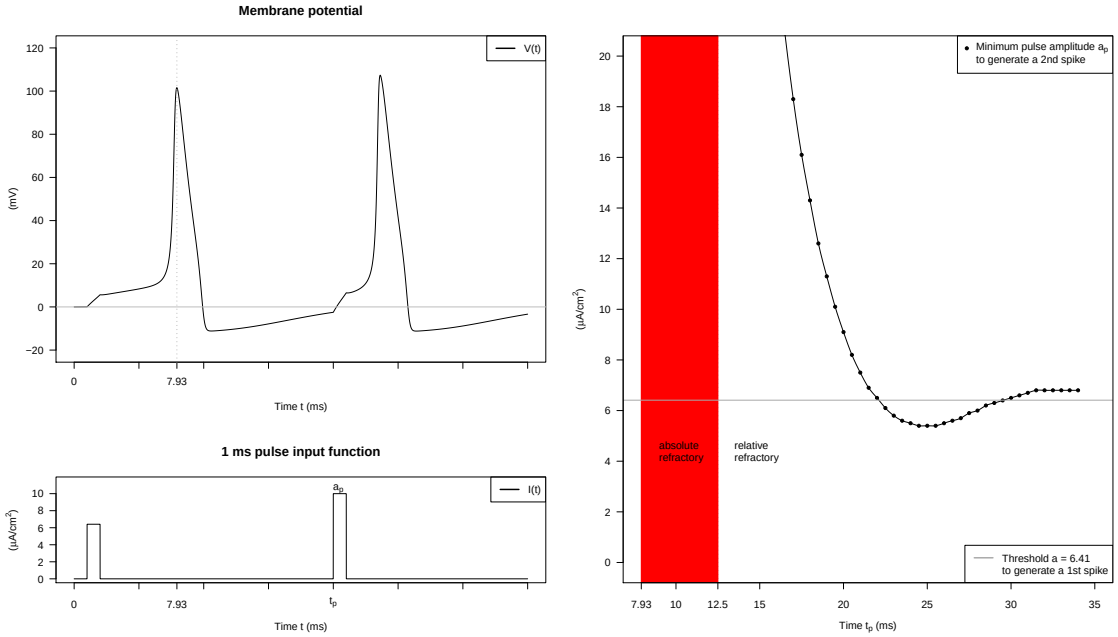


Figure 1.5: Refractory period for 1 ms pulses with an amplitude for the first pulse of 6.41. The second pulse is initiated at $t = t_p > 7.93$ with amplitude a_p. In the right figure the curve for the minimum amplitudes to generate a second spike can be observed. The approximate absolute refractory is for $t_p \in (7.93, 12.5]$ and the approximate relative refractory is for $t_p \in (12.5, \infty)$.

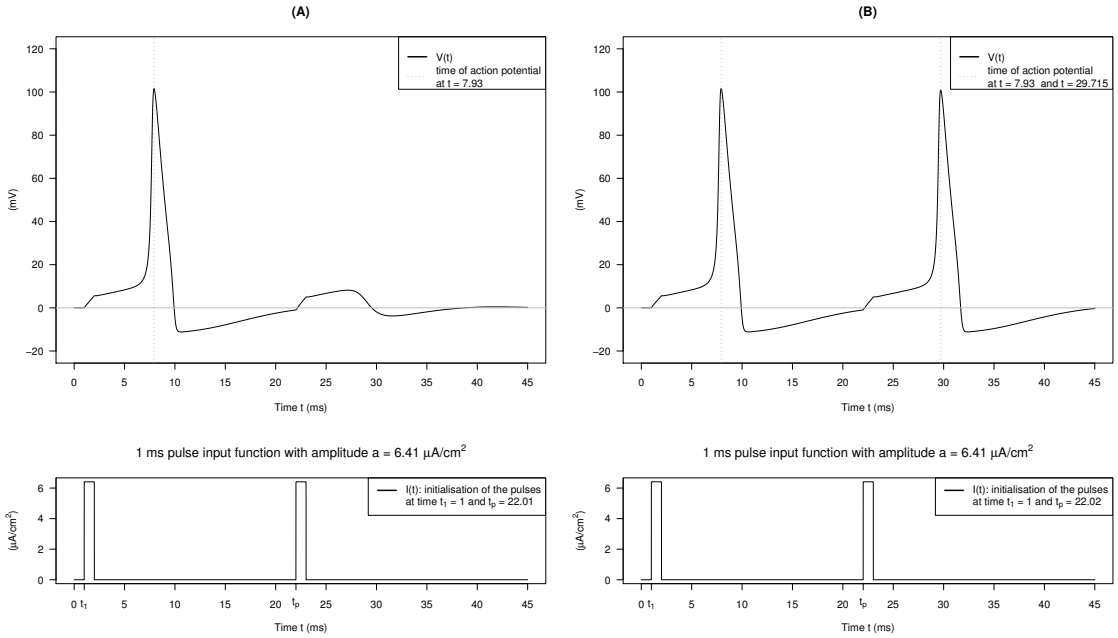


Figure 1.6: Generation of the second action potential for 1 ms pulses with amplitude $a = 6.41$. The second pulse is initiated at $t_p = 22.01$ **(A)** and $t_p = 22.02$ **(B)**. In **(B)** the second spike can be observed at time $t = 29.715$, whereas in **(A)** no second action potential is generated.

1.2 Implementation

This section is a description of the implementations given in appendix A. During the work on this thesis the code changed. At the beginning programming language R was used for the implementation of the Hodgkin-Huxley equations. R is a free software environment for statistical computing and graphics (see `http://www.r-project.org/`). For numerical purposes R is really slow. Therefore the simulations took a lot of time. Markus Tacke, a computer expert of the University of Mainz, advised me to rewrite the code in programming language C. Indeed, the optimized C code was more than 30 times faster than the code in R. In the C code the calculated values are directly written into a .txt-file. After the simulations the text files could be read and analyzed with R, e.g. to find the times when the membrane potential $(V(t))_{t\geq 0}$ has an action potential:

```
spikeindex <- intersect(which(diff(sign(diff(V)))==-2)+1,which(V>75));
spiketimes <- spikeindex*s;
```

where V is the data of the simulated membrane potential and s is the step size of the numerical method. Here, an action potential is defined as an amplitude over 75 mV.
First, the formulae of 1.1 Definition (i) were implemented (see A.1 lines 5-79). For the implementation of equations (1.3), (1.8), (1.9) and (1.10) see A.1 lines 81-100. Note that for some of the simulations both choices of the equilibrium potentials (1.1), (1.7) are used (see lines 94 and 96 in A.1).
Second, the main part was implemented. Depending on what type of input function (1.2) is used, the main part slightly differs. The simulations always start at time $t_0 = 0$ ms. The initial value problem can be specified as follows:

$$y' = f(t, y), \quad y(t_0) = y_0,$$

where

$$y \coloneqq \begin{pmatrix} V \\ n \\ m \\ h \end{pmatrix}$$

and

$$f(t, y) \coloneqq \begin{pmatrix} I(t) - 36n^4(V + 12) - 120m^3h(V - 120) - 0.3(V - 10.6) \\ (n_\infty(V) - n)/\tau_n(V) \\ (m_\infty(V) - m)/\tau_m(V) \\ (h_\infty(V) - h)/\tau_h(V) \end{pmatrix},$$

with initial conditions given as the steady state when the membrane potential is zero:

$$y(0) = \begin{pmatrix} V(0) \\ n_\infty(0) \\ m_\infty(0) \\ h_\infty(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 4/(5\exp(1) - 1) \\ 5/(8\exp(\frac{5}{2}) - 3) \\ (7\exp(3) + 7)/(7\exp(3) + 107) \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0.3176769 \\ 0.0529325 \\ 0.5961208 \end{pmatrix}. \tag{1.12}$$

For the following two propositions see [2], page 157. In the first proposition the note in 1.3 Remark (ii) is proved that n, m and h are probabilities between 0 and 1.

1.6 Proposition: Under the initial conditions (1.12), the solutions $n(t)$, $m(t)$, $h(t)$ satisfy for all $t \geq 0$ the inequalities

$$0 < n(t) < 1, \quad 0 < m(t) < 1, \quad 0 < h(t) < 1. \tag{1.13}$$

Proof Suppose $n(t) = 0$ for $t > 0$. Since $n(0) > 0$ and n is continuous (due to 1.2 Proposition), there must be a first such time $\tilde{t}$. Then $n(\tilde{t}) = 0$ and $n(t) > 0$ for $t \in [0, \tilde{t})$. It follows that

$$\begin{aligned} \frac{\mathrm{d}n}{\mathrm{d}t}(\tilde{t}) &= \alpha_n(V(\tilde{t})) \cdot (1 - n(\tilde{t})) - \beta_n(V(\tilde{t})) \cdot n(\tilde{t}) \\ &= \alpha_n(V(\tilde{t})) > 0, \end{aligned}$$

because $\alpha_n(V) > 0$. Therefore n is strictly increasing in $\tilde{t}$. That means for $t \in (\tilde{t}-\delta, \tilde{t}+\delta) \cap (0, \tilde{t})$ it follows that $n(t) < n(\tilde{t}) = 0$, a contradiction to $n(t) > 0$ for $t \in [0, \tilde{t})$. The proof for $m(t)$ and $h(t)$ is equal.
Now suppose $n(t) = 1$ for $t > 0$. Since $n(0) < 1$ and n is continuous (due to 1.2 Proposition), there must be a first such time $\bar{t}$. Then $n(\bar{t}) = 1$ and $n(t) < 1$ for $t \in [0, \bar{t})$. It follows that

$$\begin{aligned} \frac{\mathrm{d}n}{\mathrm{d}t}(\bar{t}) &= \alpha_n(V(\bar{t})) \cdot (1 - n(\bar{t})) - \beta_n(V(\bar{t})) \cdot n(\bar{t}) \\ &= -\beta_n(V(\bar{t})) < 0, \end{aligned}$$

because $\beta_n(V) > 0$. Therefore n is strictly decreasing in $\bar{t}$. That means for $t \in (\bar{t}-\delta, \bar{t}+\delta) \cap (0, \bar{t})$ it follows that $n(t) > n(\bar{t}) = 1$, a contradiction to $n(t) < 1$ for $t \in [0, \bar{t})$. The proof for $m(t)$ and $h(t)$ is equal. □

1.7 Remark: Note that it follows by the definition of the steady state (in)activation functions that

$$0 < n_\infty < 1, \quad 0 < m_\infty < 1, \quad 0 < h_\infty < 1.$$

1.8 Proposition: Under the initial conditions (1.12), the solution $V(t)$ satisfy for all $t \geq 0$ the inequality

$$-12 < V(t) < 120,$$

as long as $-6.78 < I(t) < 32.82$.

Proof Suppose $V(t) = -12$ for $t > 0$. Since $V(0) = 0$ and V is continuous (due to 1.2 Proposition), there must be a first such time $\tilde{t}$. Then $V(\tilde{t}) = -12$ and $V(t) > -12$ for $t \in [0, \tilde{t})$. It follows by (1.3) and (1.13) that

$$\begin{aligned} \frac{\mathrm{d}V}{\mathrm{d}t}(\tilde{t}) &= I(\tilde{t}) - 36n^4(\tilde{t}) \cdot (V(\tilde{t}) + 12) - 120m^3(\tilde{t})h(\tilde{t}) \cdot (V(\tilde{t}) - 120) - 0.3 \cdot (V(\tilde{t}) - 10.6) \\ &> I(\tilde{t}) + 6.78. \end{aligned}$$

If $I(t) > -6.78$, then $\frac{\mathrm{d}V}{\mathrm{d}t}(\tilde{t}) > 0$. Therefore V is strictly increasing in $\tilde{t}$. That means for $t \in (\tilde{t} - \delta, \tilde{t} + \delta) \cap (0, \tilde{t})$ it follows that $V(t) < V(\tilde{t}) = -12$, a contradiction to $V(t) > -12$ for $t \in [0, \tilde{t})$.
Now suppose $V(t) = 120$ for $t > 0$. Since $V(0) = 0$ and V is continuous (due to 1.2 Proposition), there must be a first such time $\bar{t}$. Then $V(\bar{t}) = 120$ and $V(t) < 120$ for $t \in [0, \bar{t})$. It follows by (1.3) and (1.13) that

$$\begin{aligned} \frac{\mathrm{d}V}{\mathrm{d}t}(\bar{t}) &= I(\bar{t}) - 36n^4(\bar{t}) \cdot (V(\bar{t}) + 12) - 120m^3(\bar{t})h(\bar{t}) \cdot (V(\bar{t}) - 120) - 0.3 \cdot (V(\bar{t}) - 10.6) \\ &< I(\bar{t}) - 32.82. \end{aligned}$$

If $I(t) < 32.82$, then $\frac{\mathrm{d}V}{\mathrm{d}t}(\bar{t}) < 0$. Therefore V is strictly decreasing in $\bar{t}$. That means for $t \in (\bar{t} - \delta, \bar{t} + \delta) \cap (0, \bar{t})$ it follows that $V(t) > V(\bar{t}) = 120$, a contradiction to $V(t) < 120$ for $t \in [0, \bar{t})$. □

Hodgkin and Huxley [8] solved their equations numerically by the method of Hartree (see [8], page 523), and showed that the action potentials, threshold for generating action potentials and refractory period compared to their recorded data approximately match. For the numerical calculations in chapter 2 the classical forth-order Runge-Kutta method was used. The assumption for f being a sufficiently differentiable function is satisfied by the input functions used in chapter 2. The formulae of the classical Runge-Kutta method are as follows (see [6], chapter XIV, section 76):

$$y(t+s) = y(t) + \frac{s}{6}(k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4),$$

where

$$\begin{aligned} k_1 &= f(t, y(t)), \\ k_2 &= f(t + \frac{s}{2}, y(t) + \frac{s}{2}k_1), \\ k_3 &= f(t + \frac{s}{2}, y(t) + \frac{s}{2}k_2), \\ k_4 &= f(t + s, y(t) + s \cdot k_3). \end{aligned}$$

Since the input in chapter 3 is stochastic, the assumption for f being a sufficiently differentiable function is no longer satisfied. Therefore, the Euler method is used for the simulations. The formula of the Euler method is as follows (see [6], chapter XIV, section 74):

$$y(t+s) = y(t) + s \cdot f(t, y(t)).$$

The accuracy of the classical Runge-Kutta method is better than the accuracy of the Euler method ($\mathcal{O}(s^4)$ and $\mathcal{O}(s)$, respectively, where s is the step size, see [6], chapter XIV, section 74 and 76). The step size for both methods was set to be $s = 0.005$.

Chapter 2

Phenomena of the Hodgkin-Huxley Model

Due to the choice of which type of input function I (1.2) is used, the reaction of the Hodgkin-Huxley equations are different. In this chapter three diverse types are analyzed.

2.1 Constant Input

Abstract

Rinzel and Miller [15] calculated an interval (I_1, I_2) for a deterministic constant input $I(t)$, where the solution of the Hodgkin-Huxley equations is periodic. In [15] the equilibrium potentials (1.7) were used. The intention of this section is to approve the results in [15]. In addition, such an interval will be calculated using the equilibrium potentials (1.1).

Introduction and Method

The Hodgkin-Huxley equations were analyzed for a constant input

$$I(t) \equiv a, \quad a \geq 0,\, t \geq 0. \tag{2.1}$$

For the examinations three different levels for the amplitude a are needed:

(i) a_{fs}, the amplitude level to generate a first spike,

(ii) a_{lrs}, the lowest amplitude level to generate regular spiking, and

(iii) a_{hrs}, the highest amplitude level to generate a regular solution (later it is going to be clear why the word "solution" is more appropriate than "spiking").

Cole, Antosiewicz and Rabinowitz [2] calculated the value $a_{fs} = 2.2302$. Due to the so called SEAC error (see [4]), this value was corrected to $a_{fs} = 2.24097466$. For numerical simulation they used the classical Runge-Kutta method with a step size of $s = 0.01$. Compared to the step size $s = 0.005$ for the simulations of A, the results should be more accurate. Rinzel and Miller [15] and Hassard [7] analyzed that an input $a \in (a_{lrs}, a_{hrs}) = (6.2647, 154.53)$ leads to a regular solution. Rinzel and Miller [15] used a rth order difference schema with a truncation error $\mathcal{O}((\frac{2\pi}{N})^r)$, where $r = 5$ and $N \in \{75, 200, 300, 400, 600\}$.

To verify these results the code in appendix $A.1$ and $A.2$ was used for the simulations. The numerical simulation was done for time $t \in [0, 100]$ (see $A.2$, line 11) to analyse a_{fs}. Afterwards the simulations were done for time $t \in [0, 15000]$ for analyzing regular solutions of the Hodgkin-Huxley equations in a neighborhood of $a_{lrs} = 6.2647$ and $a_{hrs} = 154.53$. The upper boundary a_{hrs} plays a less significant role because of the unrealistically high magnitude of the input value. The resulting interval should be slightly different because of the different numerical methods and step sizes used in the simulations of A and the simulations of Rinzel and Miller [15], Hassard [7]. The numerical simulations were done for both choices of the equilibrium potentials (1.1) and (1.7).

Results

Results for Equilibrium Potentials (1.7)

For $0 \leq a \leq 2.23677297$ no spikes could be observed in the simulations of A. Therefore $a_{fs} = 2.23677298$, although the amplitudes differ only by 10^{-8} (see figure 2.1). As mentioned before, the value slightly differs from the result $a_{fs} = 2.24097466$ in [4] because the simulation of A has a higher accuracy.

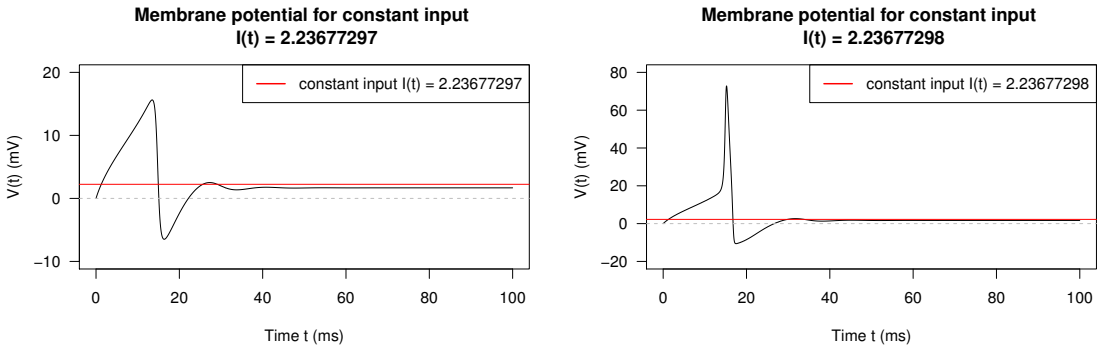


Figure 2.1: Simulation of constant input (2.1) using equilibrium potentials (1.7): $I(t) = 2.23677297$ on the left and $I(t) = 2.23677298$ on the right.

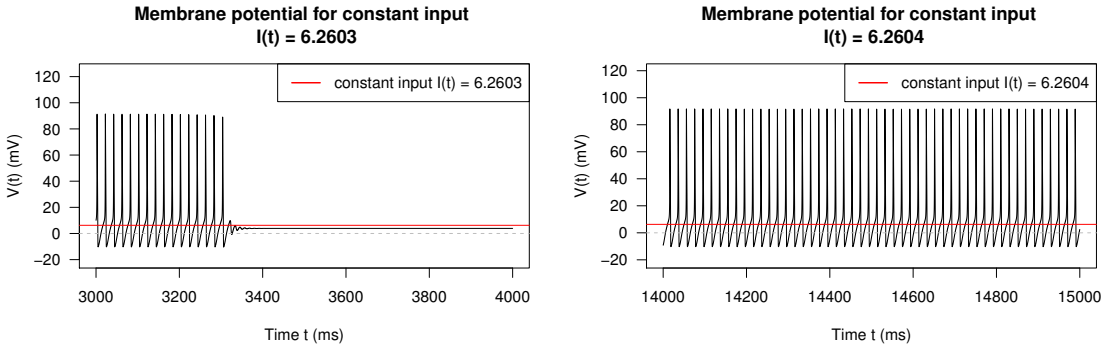


Figure 2.2: Simulation with constant input (2.1) using equilibrium potentials (1.7): different time scales for values $I(t) = 6.2603$ on the left and $I(t) = 6.2604$ on the right.

Due to figure 2.2, my result of the simulation $a_{lrs} = 6.2604$ marginally differs from the result $a_{lrs} = 6.2647$ of Rinzel and Miller [15]. It is observable that for $I(t) = 6.2603$ the Hodgkin-Huxley system stops spiking, whereas it seems that the system repetitively generates action potentials for $I(t) = 6.2604$ differing the input only by 10^{-3}. The frequency of the action potentials is approximately 50 Hz.
There are two reasons why it is not possible to calculate the highest amplitude level to generate regular spiking. First, the level to declare whether the membrane potential has a spike or not is arbitrary. Second, the higher the amplitude is in the interval $(a_{lrs}, a_{hrs}) = (6.2647, 154.53)$, the lower the spike amplitude is (see figure 2.3).

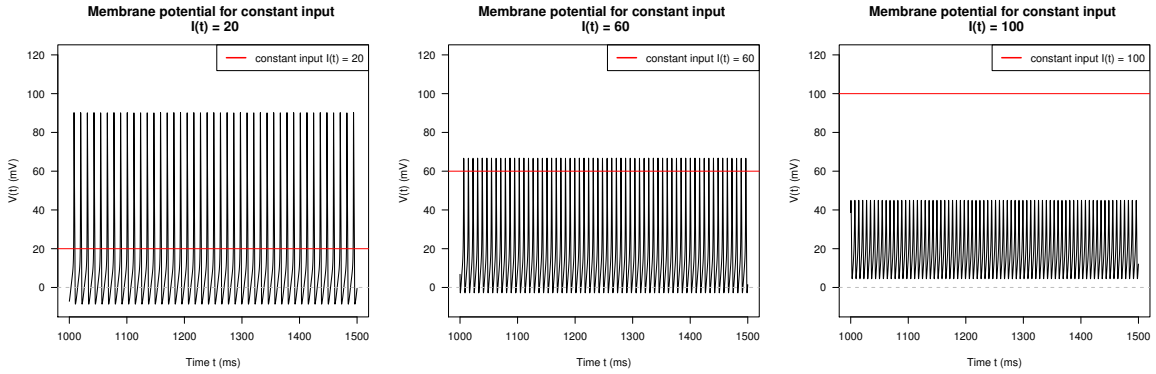


Figure 2.3: Simulation with constant input (2.1) using equilibrium potentials (1.7): $I(t) = 20$ on the left, $I(t) = 60$ in the middle and $I(t) = 100$ on the right.

It is possible to calculate a_{hrs}, the highest amplitude level for which the membrane potential is not constant (with an accuracy of 10^{-8}). The simulations of A calculated $a_{hrs} = 154.82$. In figure 2.4 the membrane potential for $a_{hrs} = 154.53$ calculated by Hassard [7] can be seen. The membrane potential still has a regular solution.

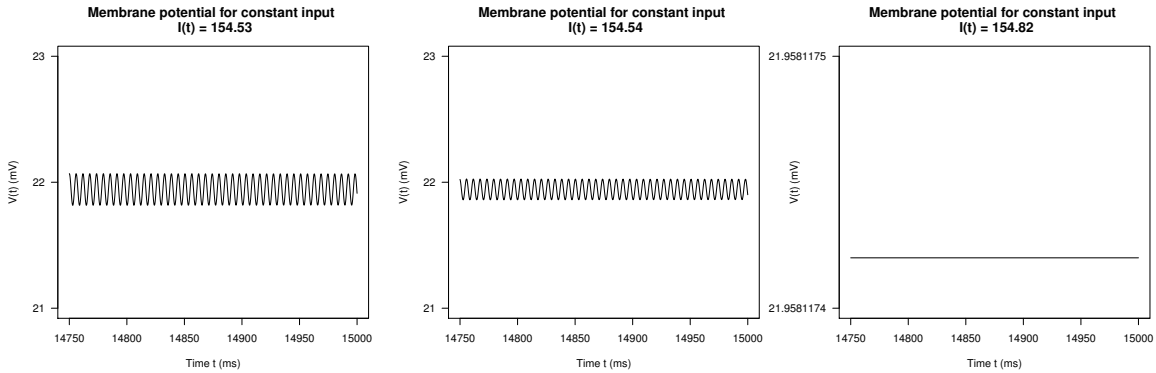


Figure 2.4: Simulation with constant input (2.1) using equilibrium potentials (1.7): $I(t) = 154.53$ on the left, $I(t) = 154.54$ in the middle and $I(t) = 154.82$ on the right.

Taking everything into account, the following can be determined:
The distinctive feature that the Hodgkin-Huxley equations have a regular solution for a constant input (2.1) in the interval (a_{lrs}, a_{hrs}) can be maintained.

Results for Equilibrium Potentials (1.1)

The simulations of A calculated $a_{fs} = 2.02775076$ (see figure 2.5) to be the amplitude level to generate a first spike. No action potentials could be observed for $0 \leq a \leq 2.02775075$.

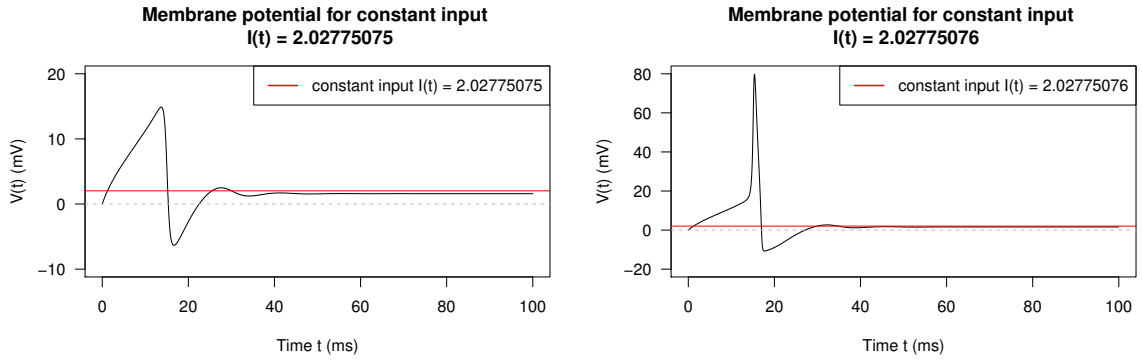


Figure 2.5: Simulation of constant input (2.1) using equilibrium potentials (1.1): $I(t) = 2.02775075$ on the left and $I(t) = 2.02775076$ on the right. On the left no spike is generated, whereas on the right an action potential can be observed.

Furthermore, for the lowest amplitude level to generate regular spiking $a_{lrs} = 5.2653$ was calculated (see figure 2.6). The frequency generating action potentials is approximately 49 Hz.

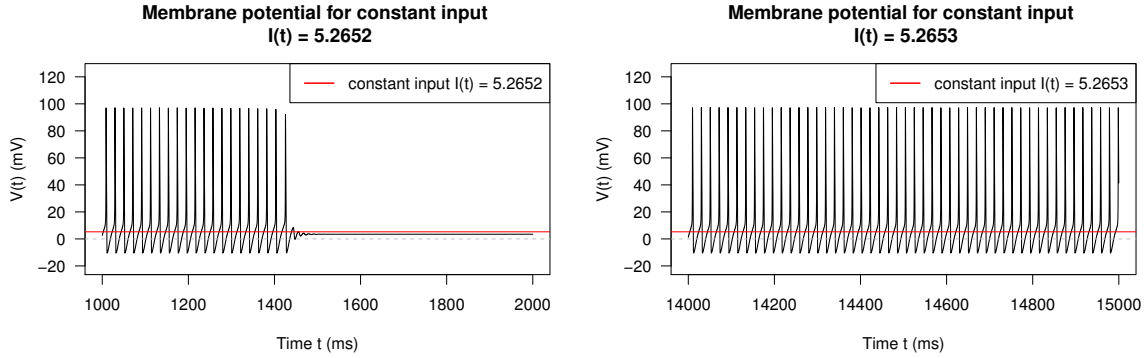


Figure 2.6: Simulation with constant input (2.1) using equilibrium potentials (1.1): different time scales for values $I(t) = 5.2652$ on the left and $I(t) = 5.2653$ on the right. On the left the spike train stops, whereas on the right the train of action potentials is probably infinite.

It is visible in figure 2.7 for $a_{hrs} = 154.53$ calculated by Hassard [7] that the membrane potential still has a regular behavior and that $a_{hrs} = 163.64$ was calculated for the highest amplitude level to generate a regular solution.
Still, with Izhikevich's equilibrium potentials (1.1) the results of Rinzel and Miller [15] could be approved.

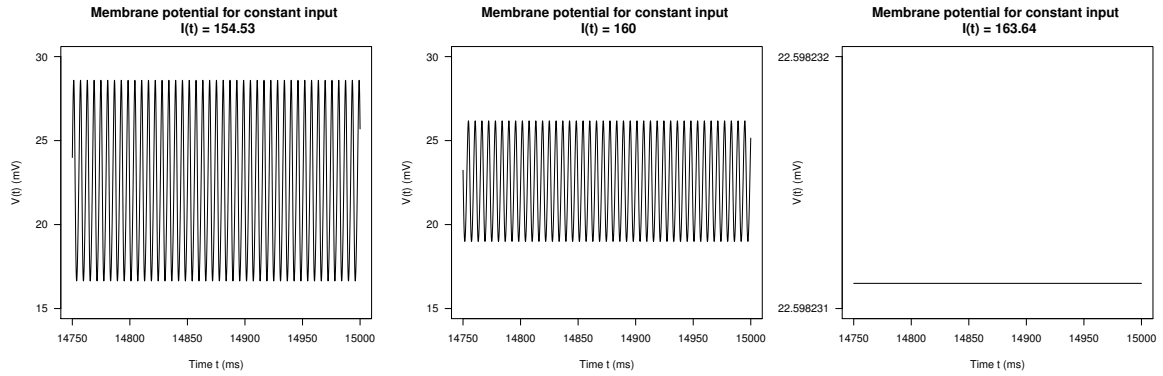


Figure 2.7: Simulation with constant input (2.1) using equilibrium potentials (1.7): $I(t) = 154.53$ on the left, $I(t) = 160$ in the middle and $I(t) = 163.64$ on the right.

Comparing the calculated levels a_{fs}, a_{lrs} and a_{hrs} for the equilibrium potentials (1.1) and (1.7), it is noticeable that a_{fs} and a_{lrs} for (1.1) are lower than a_{fs} and a_{lrs} for (1.7), and a_{hrs} for (1.1) is higher than a_{hrs} for (1.7).

2.2 Periodic Input

Abstract

In this section the phenomena described by Aihara, Matsumoto and Ikegaya [1] is explored. Dependent on the amplitude of a deterministic periodic input there is an interval $F = (f_\alpha, f_\beta)$ of frequencies so that the input with frequency $f \in F$ leads to a periodic solution of the Hodgkin-Huxley equations. The aim of this section is to verify this statement with the sinusoidal input given in [5], and to analyze the resulting periodicities.

Introduction and Method

The Hodgkin-Huxley equations were analyzed by Fohlmeister, Adelman and Poppele [5] for a periodic input

$$I(t) = I_0(1 + \sin(2\,\pi\, f_s\, t)), \quad t \geq 0, \tag{2.2}$$

with an amplitude I_0 and a frequency f_s in Hz (for implementation see $A.3$, lines 1-12). The results showed that the sinusoidal input for $10 \leq I_0 \leq 20$ with frequency $50 \leq f_s \leq 120$ leads to a so called phase-lock. Holden [9] described phase-lock as a ratio $R_{bio} = M/N$, where M is the number of action potentials of the membrane potential V occurring in N periods of the input I. Even larger values up to $I_0 = 88$ and $f_s = 200$ were analyzed but again the unrealistically high magnitudes make these values less interesting (see section 2.1). With the knowledge of section 2.1 it is more interesting to analyze the Hodgkin-Huxley equations for a sinusoidal input with smaller amplitudes. Amplitude levels or frequency levels for which the Hodgkin-Huxley equations generate the first one-to-one phase-lock (i.e. $R_{bio} = 1$) are not provided in [5]. In [1] the Runge-Kutta-Gill method with double precision, in [5] a library Runge-Kutta routine was used for numerical calculation.
The simulations were done with the code in $A.1$ and $A.3$. After simulations for $t \in [0, 200]$ to get an overview, which values make sense, the range for the parameters was set to $1 \leq I_0 \leq 5$ ($A.3$, lines 30-34) and $1 \leq f_s \leq 150$ ($A.3$, lines 35-39). For the simulations I_0 was

incremented by 0.1, f_s was incremented by 1, and $t \in [0, 2500]$ (see $A.3$, line 24). Just the data for $t \in [500, 2500]$ were analyzed because some phenomena at the beginning of the solution occurred which seems to depend on the initial conditions (1.12). For $t \geq 500$ the solution was engaged. Besides R_{bio} another ratio $R_{math} = L/N$ is needed, where L is the number of periods of the membrane potential V occurring in N periods of the input I. If the membrane potential is not periodic and accordingly not regular, R_{math} is set to be 0 (e.g. see 2.8). The ratio R_{math} makes it possible to see whether the periodicity of the input can be transfered to the Hodgkin-Huxley equations or not.

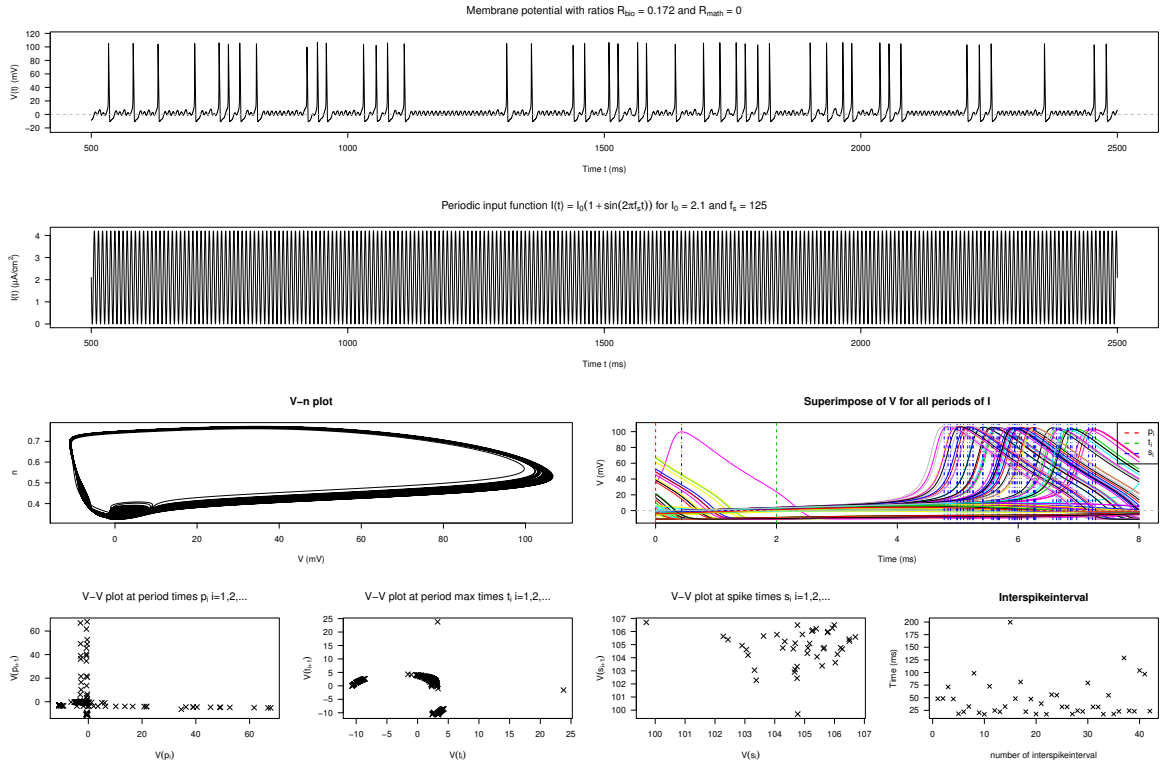


Figure 2.8: Simulation with sinusoidal input (2.2): $I_0 = 2.1$ and $f_s = 125$. For these values a non-periodic membrane potential V is generated. Therefore $R_{math} = 0$, whereas $R_{bio} = 0.172$.

Unfortunately, it was not possible to extract periodic behavior of the membrane potential data by calculation. I tried to work with discrete Fourier transformation and correlation but the results were not satisfying. Therefore graphical devices were used to decide whether the membrane potential is regular or not. Again this is dissatisfying because it is not possible to define what is regular and what is not regular. But it is a possibility to extract approximate values for the ratio R_{math}. The six graphics at the bottom two rows in figure 2.8 and figure 2.9 are different types of graphical devices which makes it possible to talk about the periodicities of the membrane potential:

(i) *V-n plot*: This is a projection from the four-dimensional space (V, n, m, h) onto the $V - n$ plane, where the flow is anticlockwise. A solid curve is a first indicator for

periodic behavior of the Hodgkin-Huxley equations. Note that there is a equilibrium at

$$\begin{pmatrix} V \\ n \\ m \\ h \\ I \end{pmatrix} = \begin{pmatrix} 0 \\ n_\infty(0) \\ m_\infty(0) \\ h_\infty(0) \\ I(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0.3176769 \\ 0.0529325 \\ 0.5961208 \\ I_0 \end{pmatrix}.$$

due to the initial conditions (1.12).

(ii) *Superimpose of V for all periods of I*: Here, the membrane potential V is divided, appropriate to the period times of the input, in periods with distance $1000/f_s$ and these periods are superimposed in one plot. Laying all periods on top of another resulting in an image like one period of the membrane potential, then it is a sign for periodicity.

(iii) *V-V plot at period times p_i, $i = 1, 2, ...$*: The period times p_i, $i = 1, 2, ...$, are the times when the periods of the input function (2.2) start at the value I_0. The distance between two successive period times is $1000/f_s$, for f_s being the frequency of the input. In this plot the membrane potential V at two successive period times is plotted against each other ($V(p_i)$ versus $V(p_{i+1})$). An indication for periodicity would be, if the graphic shows solid points, e.g. just one point, if $R_{math} = 1$, and two points, if $R_{math} = \frac{1}{2}$ and so on.

(iv) *V-V plot at period max times t_i, $i = 1, 2, ...$*: Let t_i, $i = 1, 2, ...$, be the times t when the input function (2.2) has a local maximum of $I(t) = 2 \cdot I_0$. Again the distance between two successive period max times is $1000/f_s$, for f_s being the frequency of the input. In this plot the membrane potential V at two successive period max times is plotted against each other ($V(t_i)$ versus $V(t_{i+1})$). The interpretation is the same as in (iii).

(v) *V-V plot at spike times s_i, $i = 1, 2, ...$*: Let s_i, $i = 1, 2, ...$, denote the times t when the membrane potential $V(t)$ spikes. As mentioned in section 1.2, a spike needs to have an amplitude over 75, otherwise it is not declared as a spike. The choice of this level is arbitrary. In this plot the membrane potential V at two successive spike times is plotted against each other ($V(s_i)$ versus $V(s_{i+1})$). The graphics (iii) and (iv) are better for analyzing periodicities because the times p_i and t_i are independent of the action potentials.

(vi) *Interspikeinterval*: An interspikeinterval is the time between two successive action potentials. This graphic is a simple plot of the successive occurring interspikeintervals as points. If the graphic shows any patterns of points, the assumption can be made that the membrane potential is periodic.

Another thing that was done was to calculate the f_s local maxima of the membrane potential and examine these maxima, if any patterns can be observed. This was sometimes helpful, too. Throughout this section the equilibrium potentials (1.1) were used. The reason is that in [1] a modified Hodgkin-Huxley model was used, and in [5] the input amplitudes I_0 were to big and it is not clear whether the original or a strongly adapted version of the Hodgkin-Huxley model was used. Therefore it was not possible to compare my results to the results of [1] or [5].

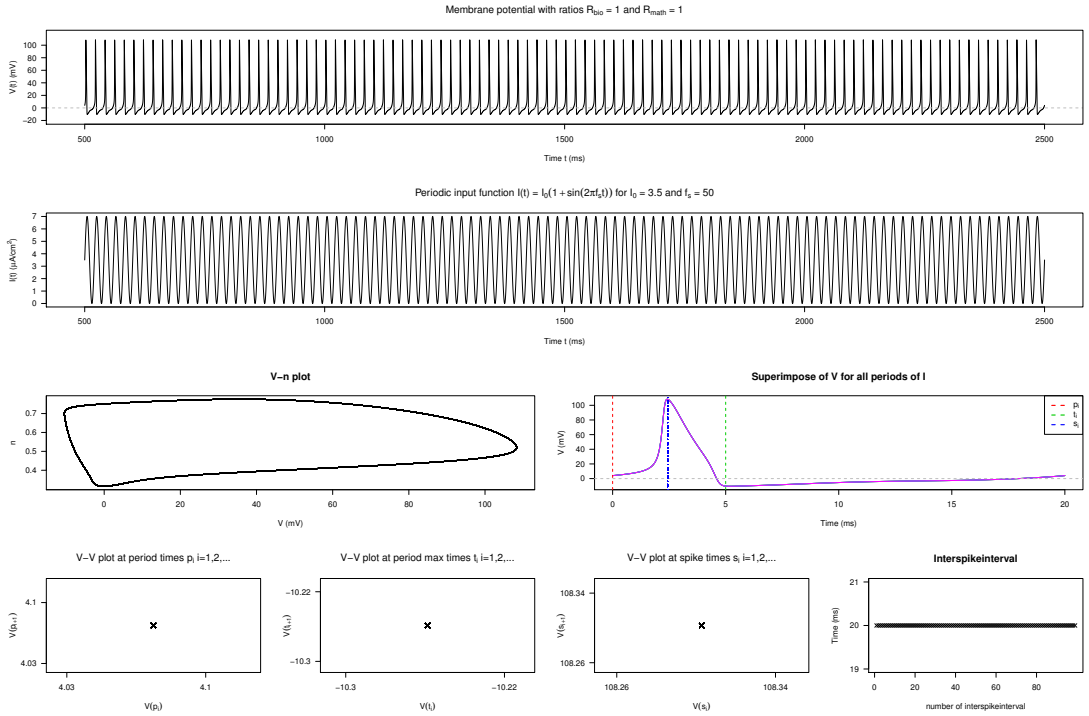


Figure 2.9: Simulation with sinusoidal input (2.2): $I_0 = 3.5$ and $f_s = 50$. All analyzing graphics let us assume that the solution is periodic with $R_{math} = 1 = R_{bio}$.

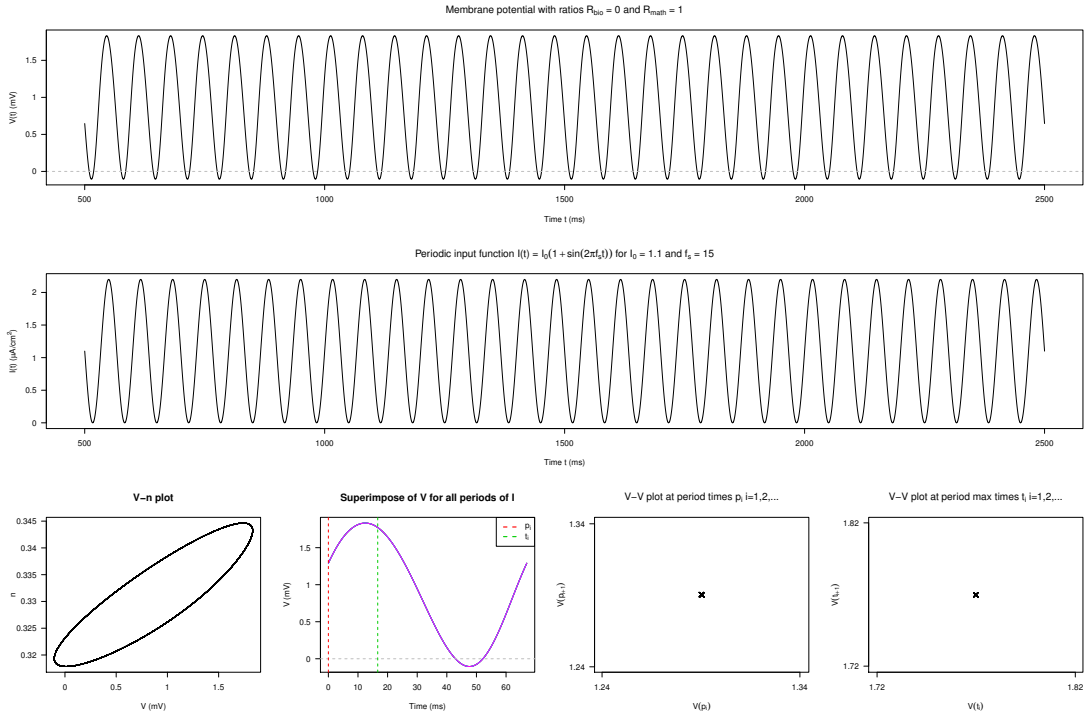
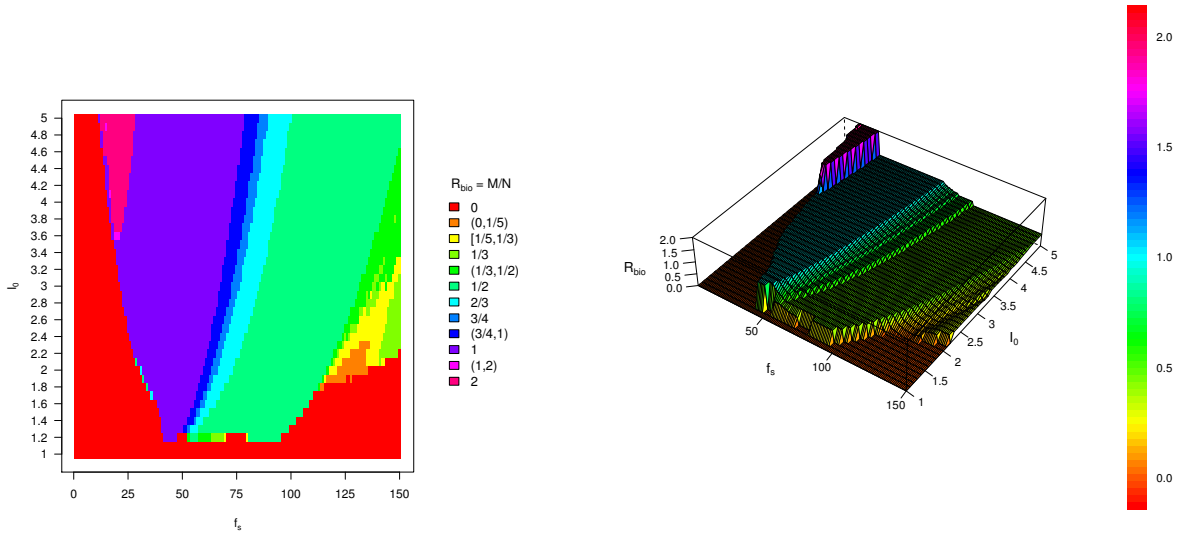


Figure 2.10: Simulation with sinusoidal input (2.2): $I_0 = 1.1$ and $f_s = 15$. It is visible that no action potentials are generated. Therefore $R_{bio} = 0$, whereas $R_{math} = 1$.

Results

For $I_0 = 1$ and $I_0 = 1.1$, $t \geq 500$, no spikes could be observed for any frequency $1 \leq f_s \leq 150$. But the membrane potential adopts the periodicity of the input function (2.2). In figure 2.10 all graphical devices let us assume this adoption. For the two amplitudes $I_0 = 1$ and $I_0 = 1.1$ the Hodgkin-Huxley equations behave periodically for all frequencies $1 \leq f_s \leq 150$. Here, the big difference between R_{bio} and R_{math} is that the biological ratio depends on the number of spikes generated, but if $R_{bio} = 0$, it does not mean that the membrane potential is not periodic. For example, in figure 2.10 the membrane potential is periodic with $R_{math} = 1$.
The biological ratio can also be misinterpreted to analyze periodicities when the membrane potential is not periodic. Still, it is possible to calculate the biological ratio, but since there is no periodicity, it is not meaningful. For example, in figure 2.8 the membrane potential is not periodic ($R_{math} = 0$), even though $R_{bio} = 0.172$.



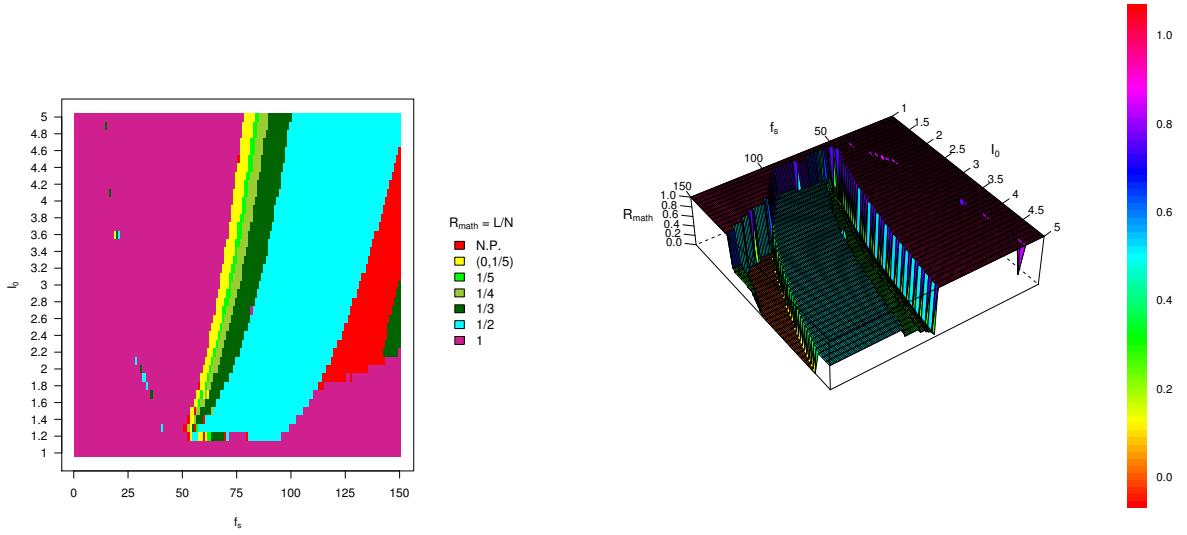
(A) **(B)**

Figure 2.11: Ratios R_{bio} of the sinusoidal input (2.2) for $1 \leq I_0 \leq 5$ and $1 \leq f_s \leq 150$. **(A)** shows the 2-dimensional image graphic of the ratios and **(B)** shows the ratios in a 3-dimensional wireframe graphic.

Figure 2.11 shows the ratios R_{bio} of the simulations in a 2-dimensional image graphic **(A)** and in a 3-dimensional wireframe graphic **(B)**. Some of the ratios did not appear frequently and not all solutions for the membrane potential show periodic behavior (see figure 2.8), therefore these ratios were combined to an interval. I.e. in the interval $(3/4, 1)$ are mainly the ratios $4/5$, $7/8$ and $9/10$; in the interval $(1, 2)$ are mainly the ratios $5/4$, $3/2$ and $5/3$; in the intervals $(0, 1/5)$, $[1/5, 1/3)$ and $(1/3, 1/2)$ there are no frequently appearing, specific ratios, and talking about R_{math} it will be in evidence that in these intervals the membrane potential is not periodic. The red part in 2.11 shows that for small frequency ranges no action potentials are generated. The lower the amplitude I_0, the higher the frequency f_s necessary to generate a first phase lock. It is visible in figure 2.11 that after the appearance of the first phase lock for the membrane potential the fraction R_{bio} decreases as the frequency f_s increases, but remains

a simple rational number over most of the frequency range.
Figure 2.12 shows the ratios R_{math} of the simulations in a 2-dimensional image graphic **(A)** and in a 3-dimensional wireframe graphic **(B)**. If $R_{bio} = 0$, then the periodicity of sinusoidal input is almost always transfered to the membrane potential as in figure 2.10, and therefore $R_{math} = 1$. Only in ranges, where a phase-lock is changed over to $R_{bio} = 0$, it can happen that $R_{math} \neq 1$. If $R_{bio} \in [1/2, 1)$, it is observable that $R_{math} = 1 - R_{bio}$ (see figure 2.13). If $R_{bio} = 2$, then $R_{math} = 1$ (see figure 2.14). For $R_{bio} \in (1, 2)$ it is noticeable that almost always $R_{math} \in \{1/3, 1/2\}$. As mentioned before non periodic membrane potentials are generated for the most part when $R_{bio} \in (0, 1/5) \cup [1/5, 1/3) \cup (1/3, 1/2)$. There is a band of non-periodic behavior starting at $I_0 = 1.8$ for high frequencies $f_s \geq 113$ (see figure 2.8). An interesting fact is that after this band of non-periodic behavior another phase-lock occurs starting at $I_0 = 2.2$ and high frequencies $f_s \geq 143$ (see figure 2.15). Due to the small step size used for the simulations, the irregular behavior is not based on the error of numerical calculation.
Figure 2.16 shows the minimum interspikeintervals of the simulations in a 2-dimensional image graphic **(A)** and in a 3-dimensional wireframe graphic **(B)**. If less than two spikes are generated, the minimum interspikeinterval was set to be 0. It is observable that depending on the magnitude of the input amplitude I_0, if the minimum interspikeinterval reaches a critical value, then afterwards the minimum interspikeinterval raises. This critical value is the transition to the phase lock $R_{bio} = 1/2$.

Taking everything into account, the following can be determined:
The distinctive feature that the Hodgkin-Huxley equations have periodic solutions for a periodic input (2.2) can be maintained for most of the amplitude and frequency range.



(A) **(B)**

Figure 2.12: Ratios R_{math} of the sinusoidal input (2.2) for $1 \leq I_0 \leq 5$ and $1 \leq f_s \leq 150$. **(A)** shows the 2-dimensional image graphic of the ratios and **(B)** shows the ratios in a 3-dimensional wireframe graphic. Note that "N.P." in the legend stands for "non periodic".

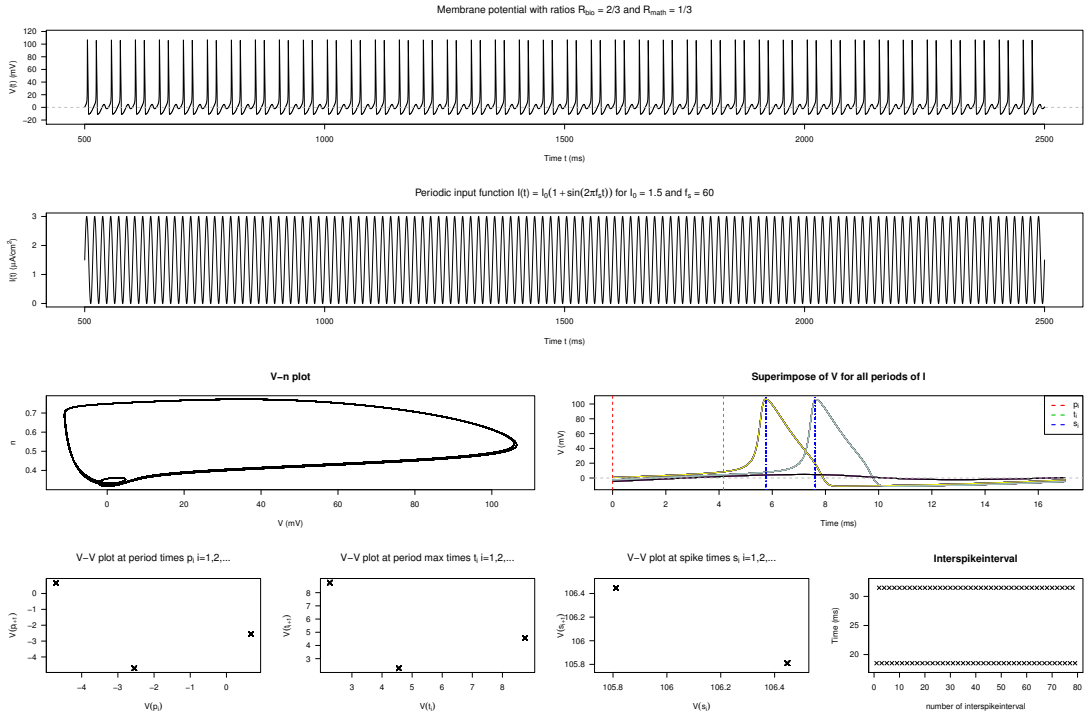


Figure 2.13: Simulation with sinusoidal input (2.2): $I_0 = 1.5$ and $f_s = 60$. It is visible that all graphical devices let us assume that $R_{bio} = 2/3$, whereas $R_{math} = 1/3$.

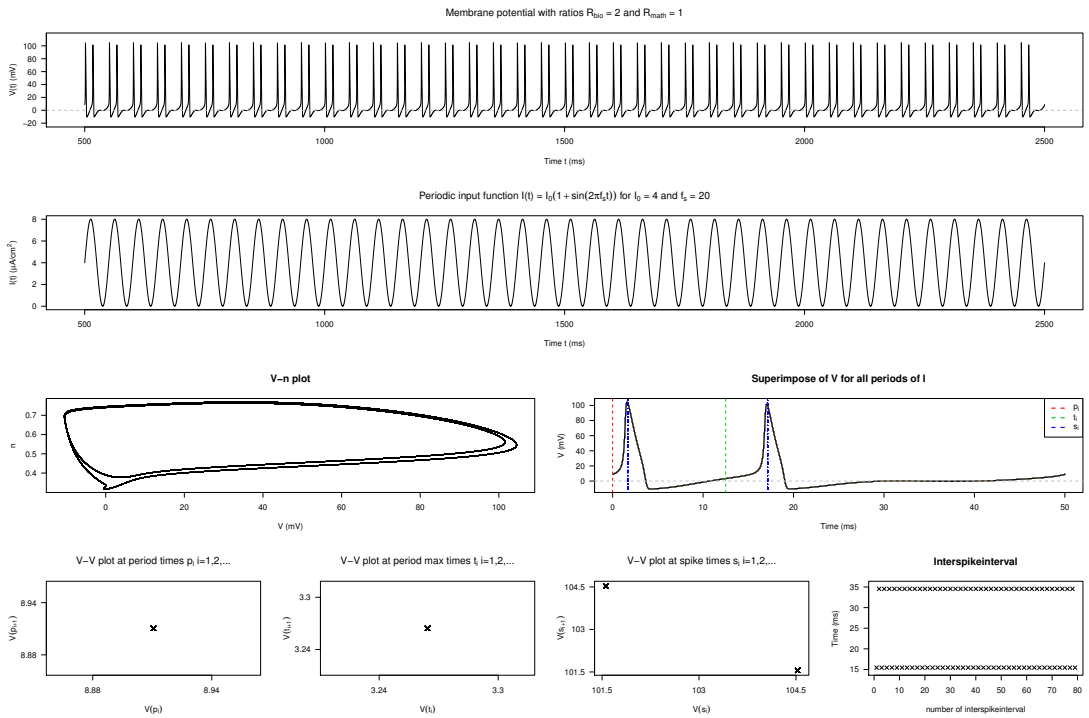


Figure 2.14: Simulation with sinusoidal input (2.2): $I_0 = 4$ and $f_s = 20$. It is visible that all graphical devices let us assume that $R_{bio} = 2$, whereas $R_{math} = 1$.

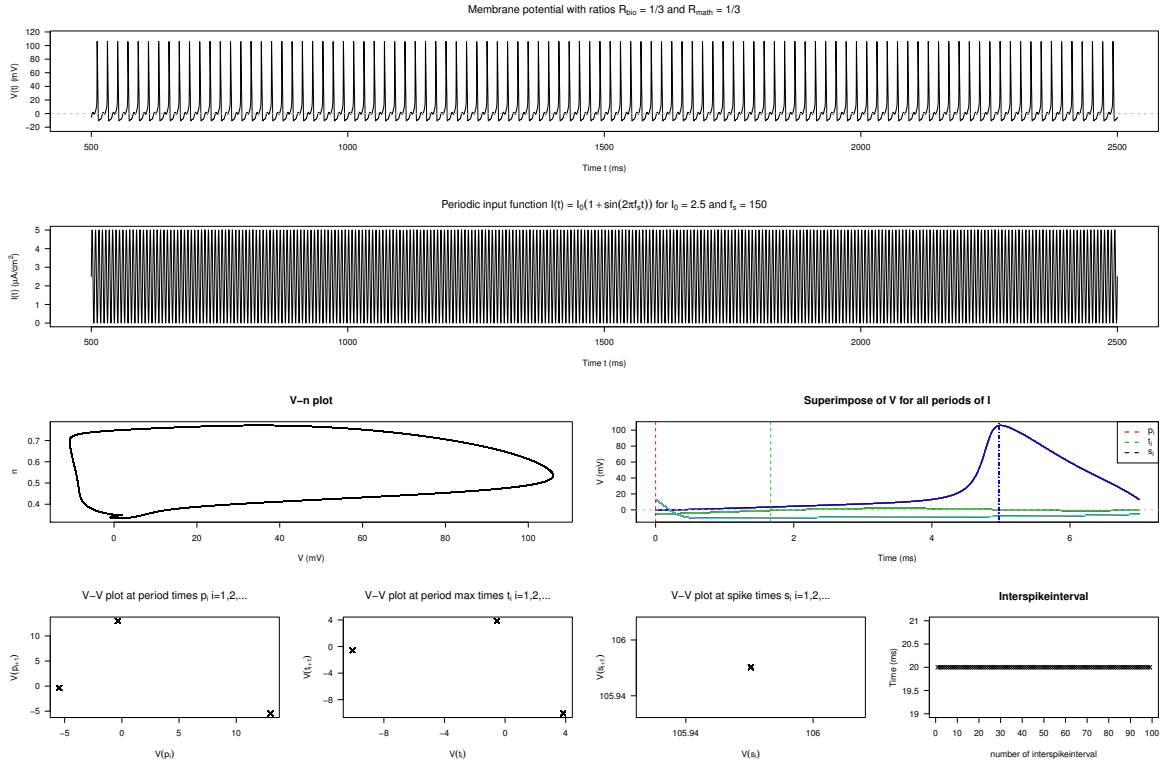
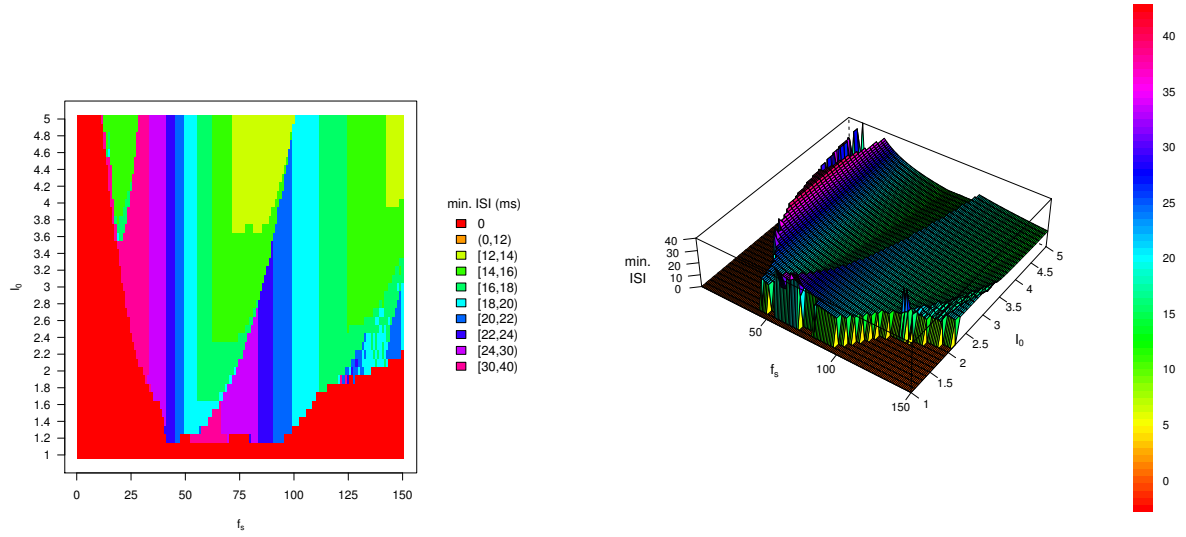


Figure 2.15: Simulation with sinusoidal input (2.2): $I_0 = 2.5$ and $f_s = 150$. It is visible that all graphical devices let us assume that $R_{bio} = 1/3$ and $R_{math} = 1/3$.



(A) **(B)**

Figure 2.16: Minimum interspikeintervals of the sinusoidal input (2.2) for $1 \leq I_0 \leq 5$ and $1 \leq f_s \leq 150$. **(A)** shows the 2-dimensional image graphic of the minimum interspikeinterval and **(B)** shows the minimum interspikeinterval in a 3-dimensional wireframe graphic.

2.3 1 ms Pulse Input

Abstract

In the last section of this chapter the phenomena depicted by Izhikevich in [12] is analyzed. Izhikevich [12] stated that depending on the input amplitude of brief pulses there are frequencies $f \in F$, $F = (f_\alpha, f_\beta)$, which lead to a periodic solution of the Hodgkin-Huxley equations. Again the ambition is to confirm this statement, and to analyze the resulting periodicities as it was done in section 2.2.

Introduction and Method

In [12] on page 5 Izhikevich described the behavior of the Hodgkin-Huxley equations for an input of brief pulses. In this section a 1 ms pulse input

$$I(t) = \mathbb{1}_{(1,\infty)}(t) \sum_{m=0}^{f-1} a \cdot \mathbb{1}_{\left[m\frac{1000}{f}, m\frac{1000}{f}+1\right]}(t \bmod 1000), \quad t \geq 0, \tag{2.3}$$

with amplitude a and frequency f in Hz was analyzed (for implementation see $A.4$, lines 1-24). $I(t) = 0$ for $0 \leq t \leq 1$ to avoid that the simulation starts with a pulse. Note that the distance between the initialization times of the pulses is $1000/f$. Izhikevich [12] noted that even negative amplitudes a can lead to spiking. It is not possible to compare my results because in [12] no ranges for the amplitudes and frequencies are mentioned.
The simulations were done with the code in $A.1$ and $A.4$. The first simulations were done to find out which ranges for the amplitudes are reasonable. The range for the amplitudes was set to $5.5 \leq a \leq 8$ and $-18.5 \leq a \leq -15.5$ ($A.4$, lines 42-49), whereas the range of frequencies $1 \leq f \leq 150$ is the same as in section 2.2 ($A.4$, lines 50-54). For the simulations a was incremented by 0.1, f was incremented by 1, and $t \in [0, 2500]$ (see $A.4$, line 36). Again only the data for $t \in [500, 2500]$ were analyzed to eliminate the behavior of the Hodgkin-Huxley equations which is dependent on the initial conditions (1.12). For $t \geq 500$ the solution was engaged.

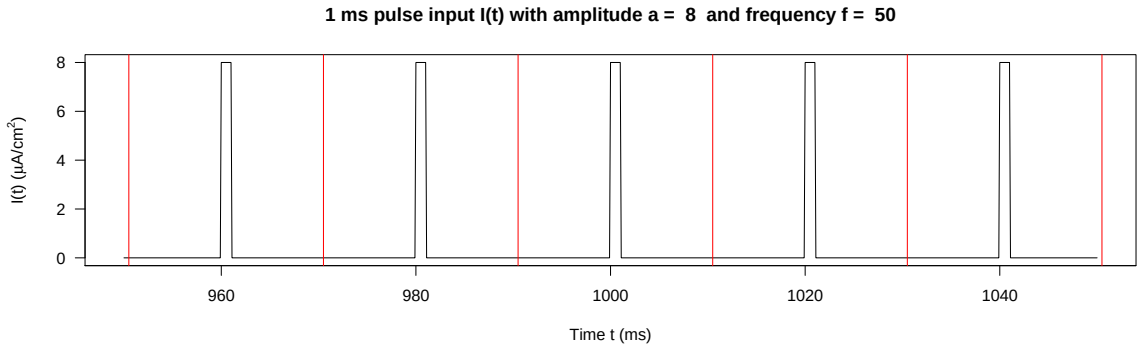


Figure 2.17: 1 ms pulse input (2.2) for $a = 8$ and $f = 50$. It is visible that the distance between the initialization of two successive pulses is $1000/f = 20$. The red lines are the period times of the pulse input.

Two different ratios are defined as it was done in section 2.2: a biological ratio $P_{bio} = M/N$, where M is the number of action potentials occurring in N periods of the input I, and a

mathematical ratio $P_{math} = L/N$, where L is the number of periods occurring for N periods of the input I. To make clear how a period for the input (2.3) is defined see figure 2.17. If the membrane potential is not periodic and accordingly not regular, P_{math} is set to be 0 (e.g. see 2.18).

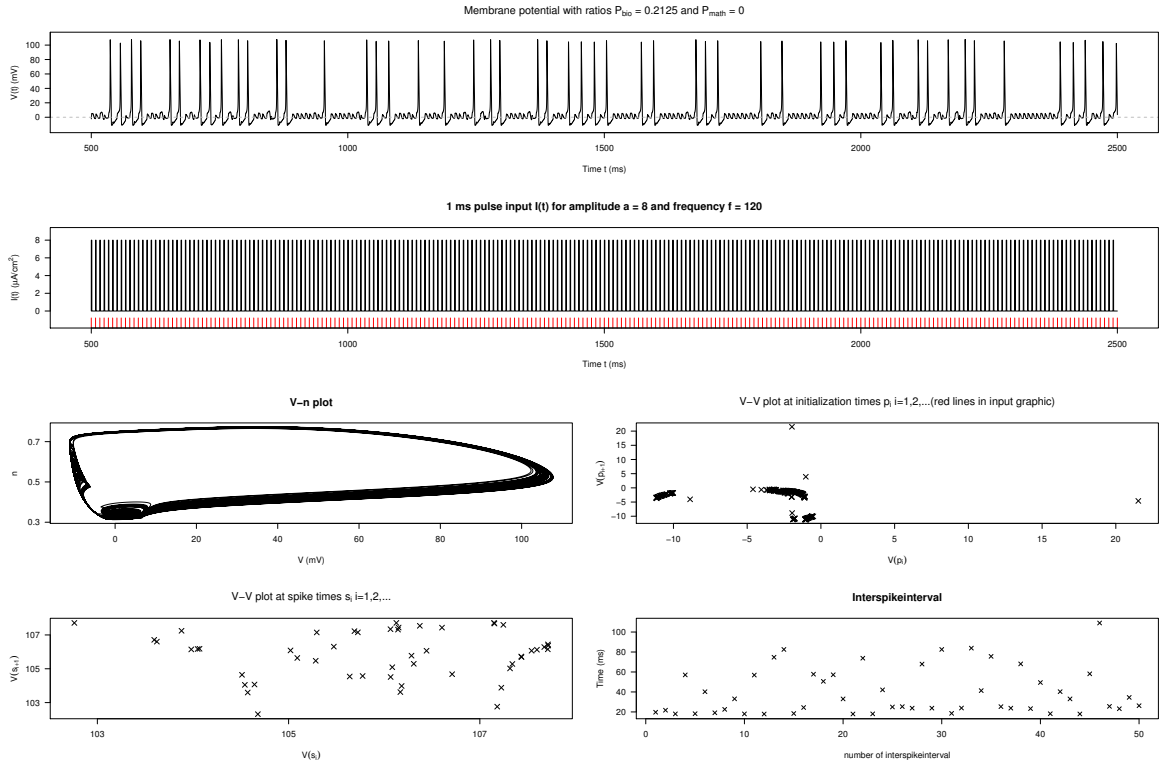


Figure 2.18: Simulation with 1 ms pulse input (2.2): $a = 8$ and $f = 120$. For these values a non-periodic membrane potential is generated. Therefore $P_{math} = 0$, whereas $P_{bio} = 0.2125$.

The following four graphical devices make it possible to talk about the periodicities of the membrane potential:

(i) *V-n plot*: See section 2.2.

(ii) *V-V plot at initialization times p_i, $i = 1, 2, ...$*: The initialization times p_i, $i = 1, 2, ...$, are the times when the pulses of the input function (2.3) start at the value a. In this plot the membrane potential V at two successive initialization times is plotted against each other ($V(p_i)$ versus $V(p_{i+1})$). An indication for periodicity would be, if the graphic shows solid points, e.g. just one point, if $P_{math} = 1$, and two points, if $P_{math} = \frac{1}{2}$ and so on.

(iii) *V-V plot at spike times s_i, $i = 1, 2, ...$*: See section 2.2.

(iv) *Interspikeinterval*: See section 2.2.

Throughout this section the equilibrium potentials (1.1) were used.

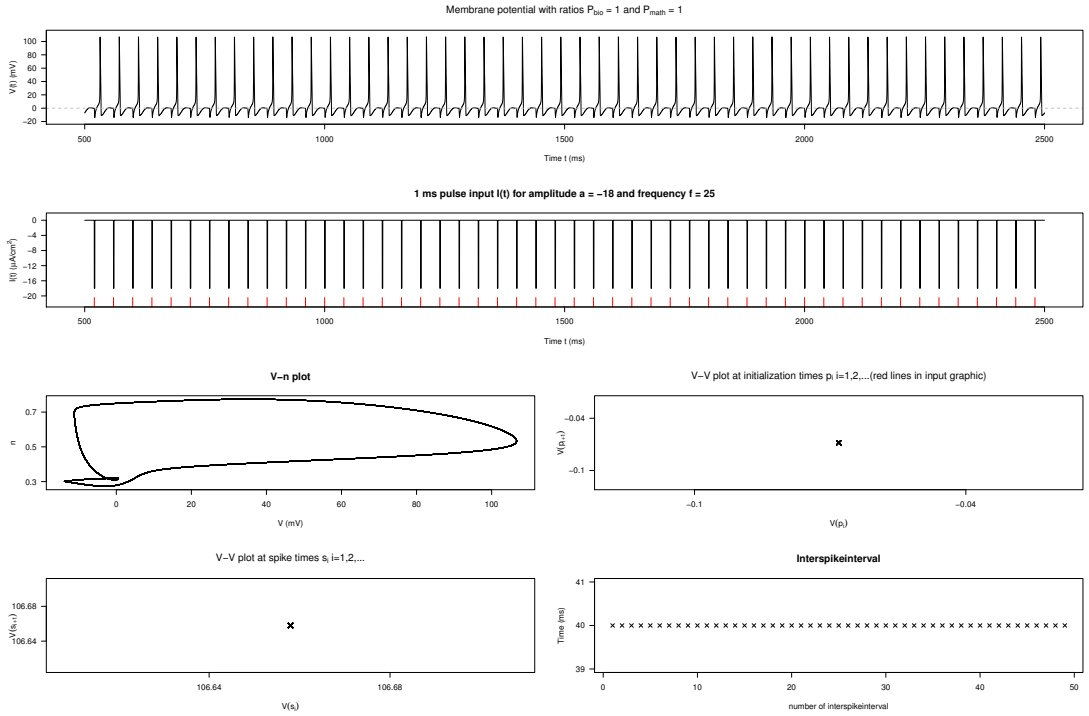


Figure 2.19: Simulation with 1 ms pulse input (2.2): $a = -18$ and $f = 25$. It is visible that all graphical devices let us assume that $P_{bio} = 1$ and $P_{math} = 1$.

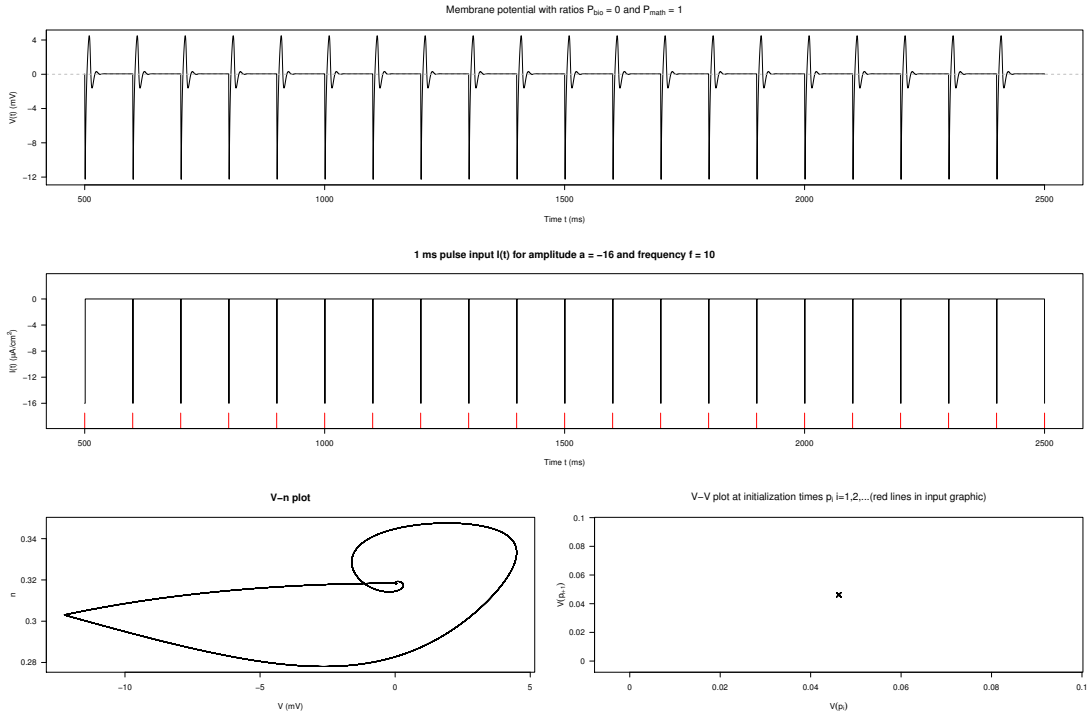
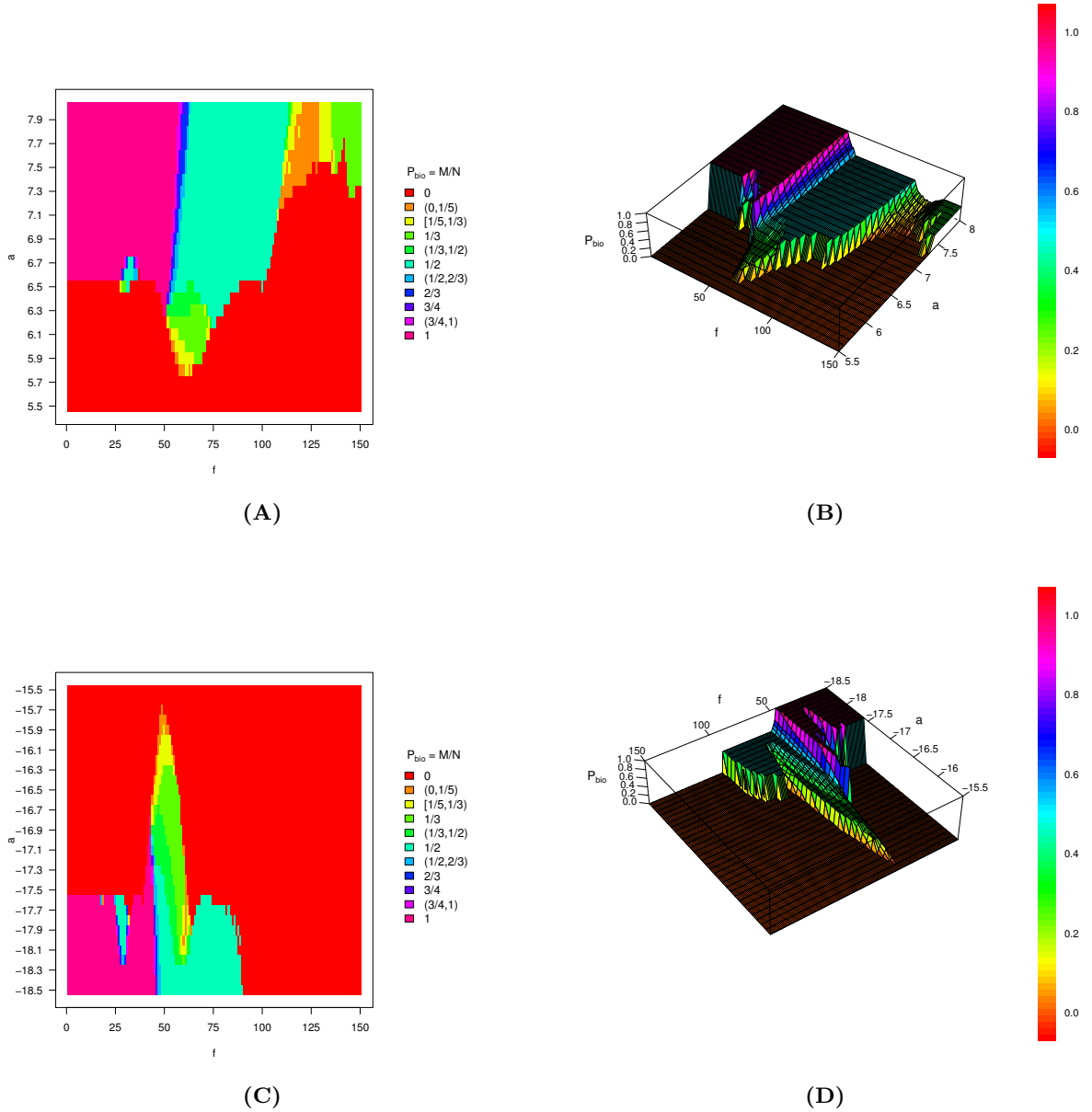


Figure 2.20: Simulation with 1 ms pulse input (2.2): $a = -16$ and $f = 10$. It is visible that all graphical devices let us assume that $P_{bio} = 0$, whereas $P_{math} = 1$.

Results

For $a \in \{-15.6, -15.5, 5.5, 5.6, 5.7\}$, $t \geq 500$, no spikes could be observed for any frequency $1 \leq f_s \leq 150$. But like in section 2.2 the membrane potential adopts the frequency of the input function (2.3) for all frequencies $1 \leq f_s \leq 150$. For amplitudes $-15.5 < a < 5.5$ the membrane potential probably behaves equal. Again, the difference between P_{bio} and P_{math} is observable in figure 2.20. It is visible that the membrane potential is periodic, even though $P_{bio} = 0$.



(A) **(B)**

(C) **(D)**

Figure 2.21: Ratios P_{bio} of the 1 ms pulse input (2.3) for $5.5 \leq a \leq 8$ **(A)**, $-18.5 \leq a \leq -15.5$ **(C)** and $1 \leq f \leq 150$. **(A)** and **(C)** show the 2-dimensional image graphic of the ratios, whereas **(B)** and **(D)** show the ratios in a 3-dimensional wireframe graphic.

Figure 2.21 shows the ratios P_{bio} of the simulations in 2-dimensional image graphics **(A)**, **(C)** and in 3-dimensional wireframe graphics **(B)**, **(D)**. The choice of ratios and combination of ratios to intervals is the same as in section 2.2, besides adding the interval $(1/2, 2/3)$, and leaving out the interval $(1, 2)$ and the ratio 2 because these ratios did not occur. If a is positive and bigger than a certain threshold, then the membrane potential starts for $f = 1$ in a one-to-one phase-lock. In figure 2.21 **(A)**, **(B)** it is observable that above this threshold the fraction P_{bio} decreases as the frequency f increases, but remains a simple rational number over most of the frequency range. If figure 2.21 **(C)**, **(D)** would be mirrored and deformed for $5.5 \leq a \leq 7$ leaving out the part for $7 < a \leq 8$, then the resulting images would look like figure 2.21 **(C)**, **(D)**. Therefore it is assumable that the behavior of the membrane potential for negative pulses is similar to the behavior of the membrane potential for positive pulses.

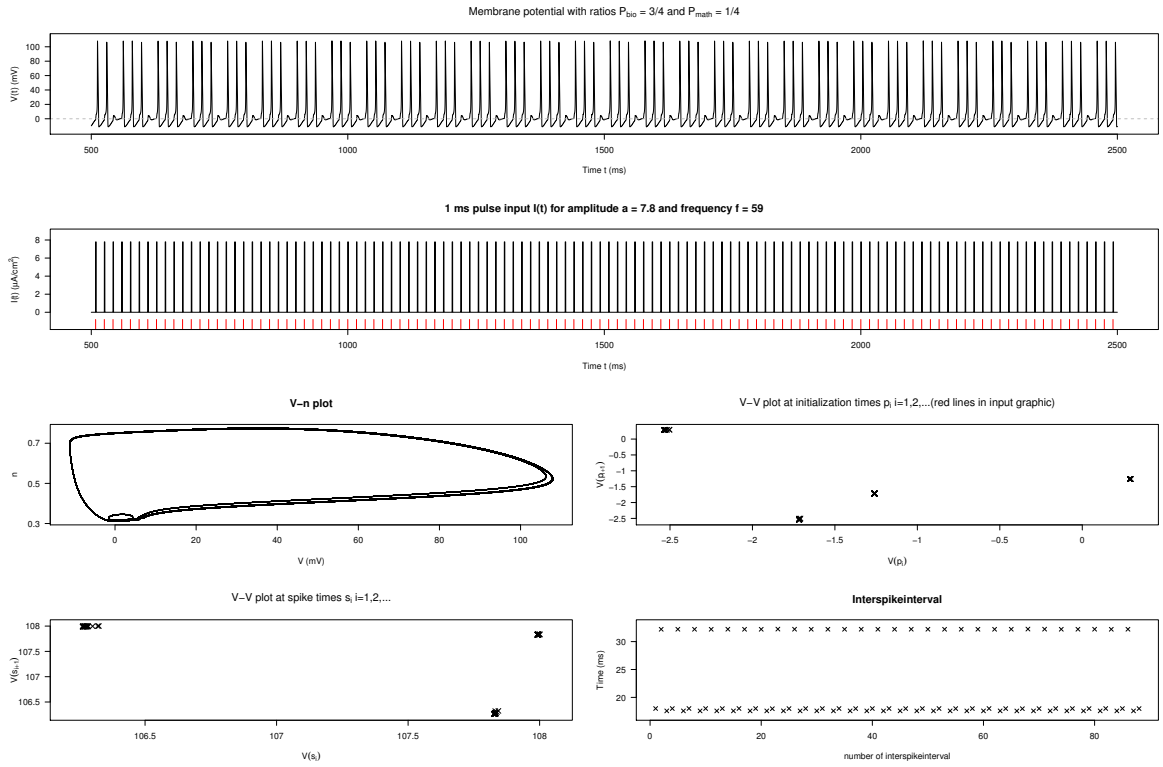


Figure 2.22: Simulation with 1 ms pulse input (2.2): $a = 7.8$ and $f = 59$. It is visible that all graphical devices let us assume that $P_{bio} = 3/4$, whereas $P_{math} = 1/4$.

Figure 2.23 shows the ratios P_{math} of the simulations in 2-dimensional image graphics **(A)**, **(C)** and in 3-dimensional wireframe graphics **(B)**, **(D)**. As in section 2.2, if $P_{bio} = 0$, then almost always the periodicity of the pulse input is transfered to the membrane potential as in figure 2.20, and therefore $P_{math} = 1$. Again it is observable that $P_{math} = 1 - P_{bio}$, if $P_{bio} \in [1/2, 1)$ (see figure 2.22). As seen in section 2.2, non periodic membrane potentials are generated for the most part when $P_{bio} \in (0, 1/5) \cup [1/5, 1/3) \cup (1/3, 1/2) \cup (1/2, 2/3) \cup (3/4, 1)$. For $a > 0$ there is a band of non-periodic behavior starting at $a = 7.1$ for high frequencies $f \geq 108$ (see figure 2.18). Again it is observable that after this band of non-periodic behavior another phase-lock occurs starting at $a = 7.3$ and high frequencies $f_s \geq 139$ (see figure 2.24). Compared to section 2.2 there is a lot more non-periodic behavior. Due to the small step size used for

the simulations, the irregular behavior is not based on the error of numerical calculation. The reason is probably the smoothness of the input (2.3) compared to the smoothness of the input (2.2). Maybe if the input is smoothed relative to the center of the pulse like $a \cdot \exp(-\frac{1}{2}x^{20})$, the results look more like in section 2.2.

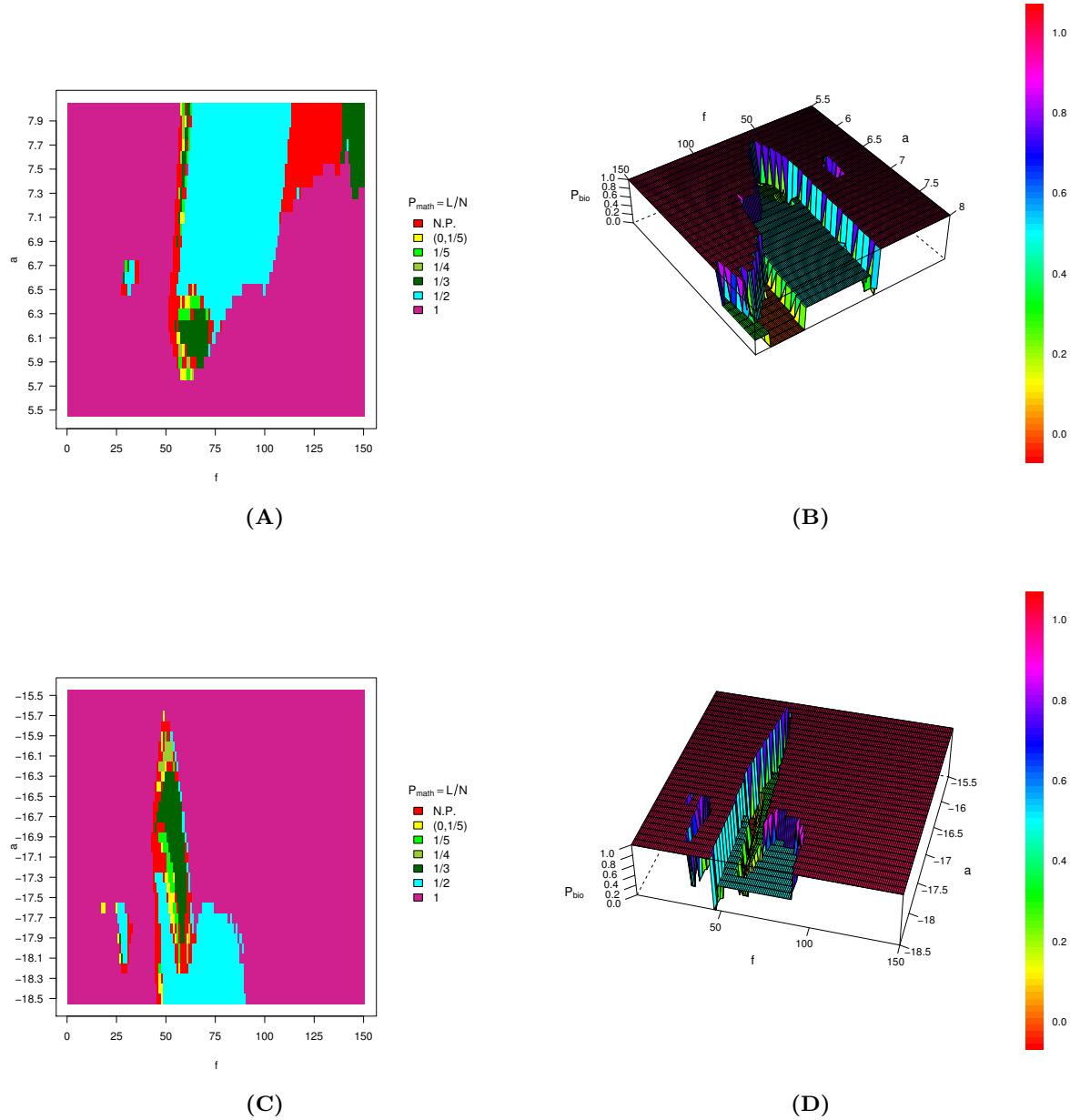


Figure 2.23: Ratios P_{math} of the 1 ms pulse input (2.3) for $5.5 \leq a \leq 8$ **(A)**, $-18.5 \leq a \leq -15.5$ **(C)** and $1 \leq f \leq 150$. **(A)** and **(C)** show the 2-dimensional image graphic of the ratios, whereas **(B)** and **(D)** show the ratios in a 3-dimensional wireframe graphic.

Taking everything into account, the following can be determined:
The distinctive feature that the Hodgkin-Huxley equations have periodic solutions for a pulse

input (2.3) can be maintained for the most part of the amplitude and frequency range. Interestingly even for negative amplitudes it is possible to generate action potentials. But compared to the positive amplitude, the negative amplitude has to be much stronger.

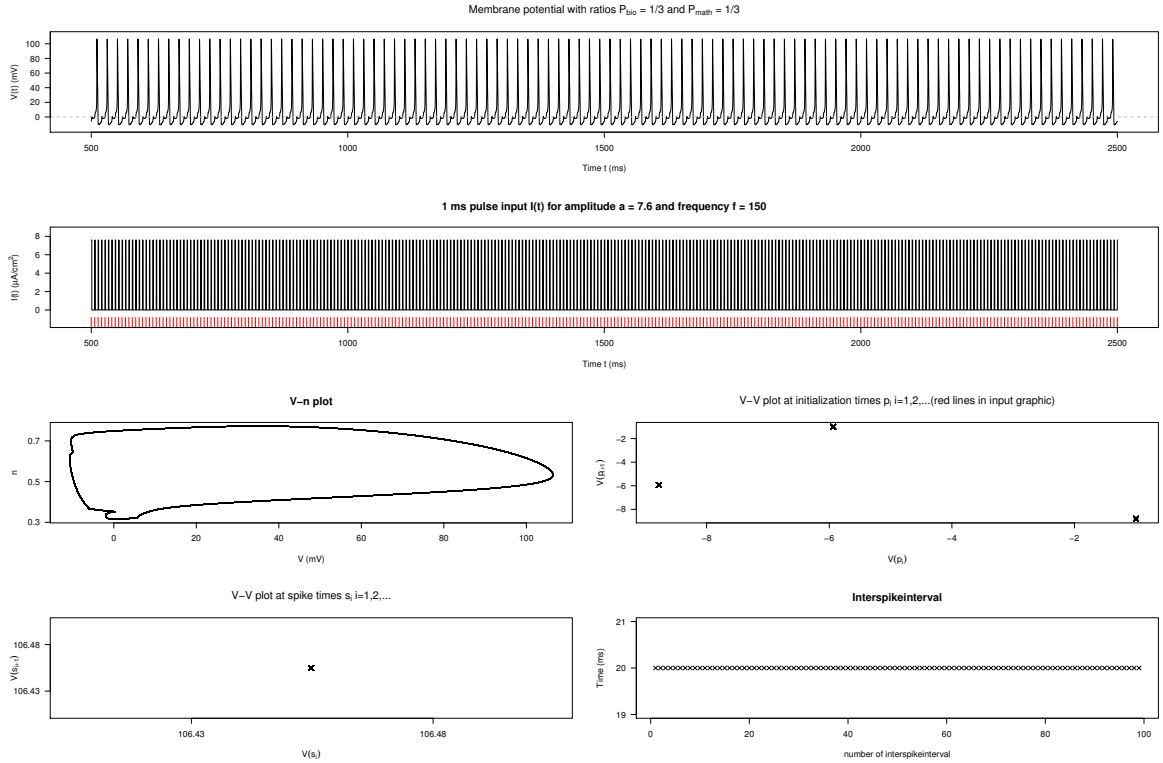


Figure 2.24: Simulation with 1 ms pulse input (2.2): $a = 7.6$ and $f = 150$. It is visible that all graphical devices let us assume that $P_{bio} = 1/3$ and $P_{math} = 1/3$.

Chapter 3

Hodgkin-Huxley Model with Stochastic Input

3.1 Stochastic Differential Equations

This section is a short introduction without any proofs to stochastic differential equations as part of the lecture "Stochastische Analysis" at the university of Mainz in 2011. For a similar introduction see [13].

3.1 Definition: Let $(\Omega, \mathcal{A}, P)$ be a probability space, $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ a filtration in $\mathcal{A}$. $(\Omega, \mathcal{A}, \mathbb{F}, P)$ is called a *stochastic basis*. A stochastic basis satisfies the *usual hypotheses*, if the following conditions hold

(i) $\mathbb{F}$ is *right continuous*, i.e. for all $t \geq 0$ holds

$$\mathcal{F}_t = \mathcal{F}_{t^+} = \bigcap_{r>t} \mathcal{F}_r.$$

(ii) $\mathbb{F}$ is *P-full*, i.e. $\mathcal{N}^P \subset \mathcal{F}_0$, where $\mathcal{N}^P$ is the system of all subsets of P-null sets in $\mathcal{A}$.

3.2 Definition: Let $(\Omega, \mathcal{A}, \mathbb{F}, P)$ be a stochastic basis, where $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ is a right continuous filtration in $\mathcal{A}$. A d-dimensional, $\mathbb{F}$-adapted and P-almost continuous process $X = (X_t)_{t\geq 0}$ is called a $(P,\mathbb{F})$-*Brownian motion*, if for all $s < t$

$$E_P(\mathrm{e}^{i\zeta^T(X_t - X_s)} \,|\, \mathcal{F}_s) = \mathrm{e}^{-\frac{1}{2}(t-s)\,|\zeta|^2}, \qquad \zeta \in \mathbb{R}^d.$$

3.3 Problem: Given a r-dimensional $(P,\mathbb{F})$-Brownian motion

$$W = (W_t)_{t\geq 0} = \begin{pmatrix} (W_t^{(1)})_{t\geq 0} \\ \vdots \\ (W_t^{(r)})_{t\geq 0} \end{pmatrix}$$

and measurable functions b_i, $\sigma_{ij} \colon [0,\infty) \times \mathbb{R}^d \to \mathbb{R}$, $1 \leq i \leq d$, $1 \leq j \leq r$. Denote

$$b(t,x) = \begin{pmatrix} b_1(t,x) \\ \vdots \\ b_d(t,x) \end{pmatrix}, \quad \sigma(t,x) = \begin{pmatrix} \sigma_{11}(t,x) & \dots & \sigma_{1r}(t,x) \\ \vdots & & \vdots \\ \sigma_{d1}(t,x) & \dots & \sigma_{dr}(t,x) \end{pmatrix},$$

where $a(t,x) = \sigma(t,x)\sigma^T(t,x) \in \mathbb{R}^{d\times d}$. $b(t,x)$ is called the *drift coefficients*, $\sigma(t,x)$ is called the *volatility* and $a(t,x)$ is called the *diffusion matrix*.
Find a $\mathbb{R}^d$-valued, P-almost continuous process $X = (X_t)_{t\geq 0}$, which is a solution of the stochastic differential equation

$$\mathrm{d}X_t = b(t,X_t)\,\mathrm{d}t + \sigma(t,X_t)\mathrm{d}W_t, \quad X_0 = \zeta \tag{3.1}$$

with

$$\mathrm{d}X_t^{(i)} = b_i(t,X_t)\,\mathrm{d}t + \sum_{j=1}^{r}\sigma_{ij}(t,X_t)\mathrm{d}W_t^{(j)}, \quad 1\leq i\leq d.$$

3.4 Remark: Let $W = (W_t)_{t\geq 0}$ be a r-dimensional Brownian motion with $W_0 = 0$ and ζ a $\mathbb{R}^d$-valued random variable in $(\Omega, \mathcal{A}', P)$. Further let W and ζ be P-independent with $\mu := \mathcal{L}(\zeta|P)$. A filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ can be developed by

$$\mathcal{G}_t = \sigma(\zeta)\vee\sigma(W_r : 0\leq r\leq t),\ t\geq 0,$$
$$\mathcal{G}_\infty = \bigvee_t \mathcal{G}_t,$$
$$\mathcal{N} = \{N\subset\Omega : \text{there is a } G\in\mathcal{G} \text{ with } P(G) = 0 \text{ and } N\subset G\},$$
$$\mathcal{F}_t := \sigma(\mathcal{G}_t,\mathcal{N}),\ t\geq 0,$$
$$\mathcal{A} := \mathcal{F}_\infty = \bigvee_t \mathcal{F}_t.$$

Then $\mathbb{F}$ satisfies the usual hypotheses and W is a $\mathbb{F}$-Brownian motion. Denote $\mathbb{F}$ as $\mathbb{F}^{W,\zeta}$.

3.5 Definition: Let $(\Omega,\mathcal{A},\mathbb{F},P)$ be as in remark 3.4 with $\mathbb{F} = \mathbb{F}^{W,\zeta}$. A *strong solution with initial value* ζ for the stochastic differential equation (3.1) with respect to the Brownian motion W and the initial value ζ is every $\mathbb{F} = \mathbb{F}^{W,\zeta}$-adapted process $X = (X_t)_{t\geq 0}$ with continuous path, so that (i) and (ii) are satisfied:

(i) for all $1\leq i\leq d$, $1\leq j\leq r$

$$\int_0^{\cdot}|b_i(s,X_s)|\mathrm{d}s,\ \int_0^{\cdot}\sigma_{ij}^2(s,X_s)\mathrm{d}s$$

are local integrable.

(ii) except for P-indistinguishableness it holds

$$X = \zeta + \int_0^{\cdot} b(s,X_s)\mathrm{d}s + \int_0^{\cdot}\sigma(s,X_s)\mathrm{d}W_s.$$

3.6 Definition: The stochastic differential equation (3.1) for b and σ as in 3.3 has the property of *strong uniqueness*, if for all choices of $(\Omega,\mathcal{A},\mathbb{F},P)$, W and ζ as in remark 3.4 and for all pairs of solutions X, X' of the stochastic differential equation (3.1) the following holds:

$$X_0 = \zeta = X_0' \Longrightarrow X = X' \quad \text{except for } P\text{-indistinguishableness.}$$

3.7 Definition: For b and σ as in 3.3 write

$$|b|^2 = \sum_{i=1}^{d} |b_i|^2, \quad ||\sigma||^2 = \sum_{i=1}^{d} \sum_{j=1}^{r} |\sigma_{ij}|^2$$

and consider the following assumptions:

(i) *Local Lipschitz (l)*: for all $n \in \mathbb{N}$ there is a $K = K_n < \infty$, so that for all $t \geq 0$ and x, $x' \in \bar{B_n}(0)$ the following holds:

$$|b(t,x) - b(t,x')| + ||\sigma(t,x) - \sigma(t,x')|| \leq K_n |x - x'|.$$

(ii) *Global Lipschitz (L)*: there is a $K < \infty$, so that for all $t \geq 0$ and x, $x' \in \mathbb{R}^d$ the following holds:

$$|b(t,x) - b(t,x')| + ||\sigma(t,x) - \sigma(t,x')|| \leq K |x - x'|.$$

(iii) *Linear growth (G)*: for all $t \geq 0$ and $x \in \mathbb{R}^d$ the following holds:

$$|b(t,x)|^2 + ||\sigma(t,x)||^2 \leq K^2 (1 + |x|^2).$$

The following two theorems go back to Itô.

3.8 Theorem: If b and σ satisfy the local Lipschitz assumption (l), then the stochastic differential equation (3.1) has the property of strong uniqueness.

3.9 Theorem: Let $(\Omega, \mathcal{A}, \mathbb{F}, P)$, W, ζ be as in remark 3.4 with $E(|\zeta|^2) < \infty$. If the global Lipschitz assumption (L) and the linear growth assumption (G) holds, then the stochastic differential equation (3.1) has a strong solution for the initial value ζ and the following holds: for all $T < \infty$ there is a $C = C(T, K)$, so that $E(|X_t|^2) \leq C(1 + E(|\zeta|^2))\mathrm{e}^{Ct}$, $0 \leq t \leq T$.

3.2 Ornstein-Uhlenbeck Process Embedding

Before the Ornstein-Uhlenbeck process is defined, Itô's formula (see [13], section 3.3) has to be mentioned.

3.10 Theorem: Let $f\colon \mathbb{R} \to \mathbb{R}$ be a function of class C^2 and let $X = \{X_t, \mathcal{F}_t\colon 0 \leq t < \infty\}$ be a continuous semimartingale (see [13], 3.1 Definition). Then P-a.s.,

$$f(X_t) = f(X_0) + \int_0^t f'(X_s)\mathrm{d}M_s + \int_0^t f'(X_s)\mathrm{d}B_s + \frac{1}{2}\int_0^t f''(X_s)\mathrm{d}\langle M\rangle_s, \tag{3.2}$$

for $0 \leq t < \infty$.

Generally, the Ornstein-Uhlenbeck process is defined as follows (see [11], page 44):

3.11 Example: For $b(t,x) = \theta(\mu - x)$ and $\sigma(t,x) \equiv \sigma$, $\mu \in \mathbb{R}$, $\theta > 0$, $\sigma > 0$ the unique solution of

$$\mathrm{d}X_t = \theta(\mu - X_t)\,\mathrm{d}t + \sigma \mathrm{d}W_t, \quad X_0 = \zeta \tag{3.3}$$

is called the *Ornstein-Uhlenbeck process*. This parametrization of the Ornstein-Uhlenbeck process is common in finance modeling, where σ is interpreted as the volatility, μ is the long-run equilibrium value of the process, and θ is the speed of reversion. With Itô's formula (3.2) and choosing $f(t,x) = x\mathrm{e}^{\theta t}$ the explicit solution of (3.3) can be obtained:
In fact,

$$f_t(t,x) = \theta f(t,x), \quad f_x(t,x) = \mathrm{e}^{\theta t}, \quad f_{xx}(t,x) = 0.$$

Therefore,

$$\begin{aligned} X_t\mathrm{e}^{\theta t} &= f(t,X_t) \overset{(3.2)}{=} f(0,X_0) + \int_0^t \theta X_s \mathrm{e}^{\theta s}\mathrm{d}s + \int_0^t \mathrm{e}^{\theta s}\mathrm{d}X_s \\ &\overset{(3.3)}{=} \zeta + \int_0^t \theta X_s \mathrm{e}^{\theta s}\mathrm{d}s + \int_0^t \mathrm{e}^{\theta s}(\theta(\mu - X_s)\mathrm{d}s + \sigma \mathrm{d}W_s) \\ &= \zeta + \theta\mu \int_0^t \mathrm{e}^{\theta s}\mathrm{d}s + \sigma \int_0^t \mathrm{e}^{\theta s}\mathrm{d}W_s \\ &= \zeta + \mu(\mathrm{e}^{\theta t} - 1) + \sigma \int_0^t \mathrm{e}^{\theta s}\mathrm{d}W_s, \end{aligned}$$

from which follows via division with $\mathrm{e}^{\theta t}$

$$X_t = \mu + (\zeta - \mu)\mathrm{e}^{-\theta t} + \sigma \int_0^t \mathrm{e}^{-\theta(t-s)}\mathrm{d}W_s.$$

Now consider an embedding of the input in an Ornstein-Uhlenbeck process:

3.12 Example: Let $S(\,\cdot\,)$ be a T-periodic and piecewise continuous function, $\sigma > 0$ and $\gamma > 0$. Consider the Ornstein-Uhlenbeck process

$$\mathrm{d}X_t = \gamma(S(t) - X_t)\,\mathrm{d}t + \sigma \mathrm{d}W_t, \quad t \geq 0 \tag{3.4}$$

with initial value x_0. The Ornstein-Uhlenbeck process can be simulated using the transition semigroup $(P_{s,t})_{0\leq s<t\leq\infty}$ of X:

$$P_{s,t}(x,\,\cdot\,) = \mathcal{N}\left(x\mathrm{e}^{-\gamma(t-s)} + \int_0^{t-s} \mathrm{e}^{-\gamma\nu}\gamma S(t-\nu)\mathrm{d}\nu, \frac{\sigma^2}{2\gamma}\left(1 + \mathrm{e}^{-2\gamma(t-s)}\right)\right) \tag{3.5}$$

(see [10], example 2.3).

3.13 Remark: Due to example 3.12, equation (1.3) of the Hodgkin-Huxley model has changed as follows:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = X_t - \bar{g}_K n^4 \cdot (V - E_K) - \bar{g}_{Na} m^3 h \cdot (V - E_{Na}) - \bar{g}_L \cdot (V - E_L), \tag{3.6}$$

where $(X_t)_{t\geq 0}$ is an Ornstein-Uhlenbeck process

$$\mathrm{d}X_t = \gamma(I(t) - X_t)\,\mathrm{d}t + \sigma \mathrm{d}W_t, \quad t \geq 0, \tag{3.7}$$

and $I(t)$ is one of the functions (2.1), (2.2) or (2.3).

The simulations of the Ornstein-Uhlenbeck inputs (see B) were done with R, a free software environment for statistical computing and graphics, and were saved in a .txt-file. After that these files were imported during the simulations in C (see A.5, lines 50-61, 69, 87 and 103). The simulation of the membrane potential is pretty much the same as before, except for adding the parameters $\gamma \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ (see A.5, lines 26-32), $\sigma \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ (see A.5, lines 33-39) for the Ornstein-Uhlenbeck process and using the Euler method (see A.5, lines 86-101) instead of the classical Runge-Kutta method. Depending on which type of input is used, some changes have to be made:

(i) For the constant input simulation lines 19-25, 48, 49, 104 have to be commented out and instead of lines 58-60 and 78, 79, lines 53, 54 and 73, 74 have to be used, respectively.

(ii) For the periodic input simulation lines 55-57 and 75-77 have to be used instead of lines 58-60 and 78, 79, respectively.

(iii) For the pulse input simulation lines 58-60 and 78, 79 have to be used.

Due to stochastic input the following graphical devices were used:

(i) *Gating variables*: This is a simple plot of the simulated gating variables $n(t)$, $m(t)$ and $h(t)$.

(ii) *V-n plot*: See section 2.2.

(iii) *V-h plot*: This plot is the same as (ii) using the gating variable $h(t)$ instead of $n(t)$. The gating variable $m(t)$ was not used because the behavior of $m(t)$ is similar to the membrane potential V.

3.2.1 Ornstein-Uhlenbeck Process for Constant Input

For the constant input of section 2.1

$$I(t) \equiv a, \quad a \geq 0,\, t \geq 0,$$

the integral in equation (3.5) can be calculated:

$$\int_0^{t-s} \mathrm{e}^{-\gamma\nu}\gamma I(t-\nu)\mathrm{d}\nu = a(1-\mathrm{e}^{-\gamma(t-s)}).$$

The simulation is fast because of the simple structure of the integral (for implementation see B.1). The two amplitude levels $a_{fs} = 2.02775076$ and $a_{lrs} = 5.2653$ calculated for the equilibrium potentials (1.1) in chapter 2, section 2.1 were examined.
a_{fs} is the amplitude level to generate a first spike. Therefore 500 Ornstein-Uhlenbeck inputs for each $\gamma \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$, $\sigma \in \{0.05, 0.25, 0.5, 0.75, 0.95\}$ were simulated, and it was counted how many times the membrane potential had at least one spike (see table 3.1). If $\sigma = 0.95$ and $\gamma \in \{0.1, 0.25\}$, or $\sigma = 0.75$ and $\gamma \in 0.1$ the behavior of generating at least one action potential is maintained for over 90% of 500 simulations. The lower σ and the higher γ, the more it seems like it is a coin flip decision whether at least one action potential is generated or not.

σ \ γ	0.1	0.25	0.5	0.75	0.9
0.05	0.552	0.536	0.518	0.56	0.562
0.25	0.532	0.49	0.516	0.528	0.538
0.5	0.712	0.622	0.514	0.534	0.524
0.75	0.952	0.832	0.676	0.504	0.526
0.95	0.982	0.968	0.794	0.626	0.506

Table 3.1: In this table the relative frequencies are given, if at least one spike is generated for 500 constant inputs embedded in an Ornstein-Uhlenbeck process.

Surprisingly even if the input is above the level $a_{fs} = 2.02775076$ no action potentials are generated. For example, this behavior is observable in figure 3.1, where the Ornstein-Uhlenbeck process $(X_t)_{t\geq 0}$ with parameters $\sigma = 0.95$ and $\gamma = 0.1$ is above the level a_{fs} for approximately $t \in (10, 35)$.
a_{lrs} was defined as the lowest amplitude level to generate regular spiking. It seems like, that if $\sigma = 0.95$ and $\gamma \in \{0.1, 0.25\}$ the behavior of generating action potentials is maintained. The only difference is that the membrane potential does not have a fixed frequency, and as long as the Ornstein-Uhlenbeck input is below the level $a_{lrs} = 5.2653$ no action potentials are generated (see figure 3.2).

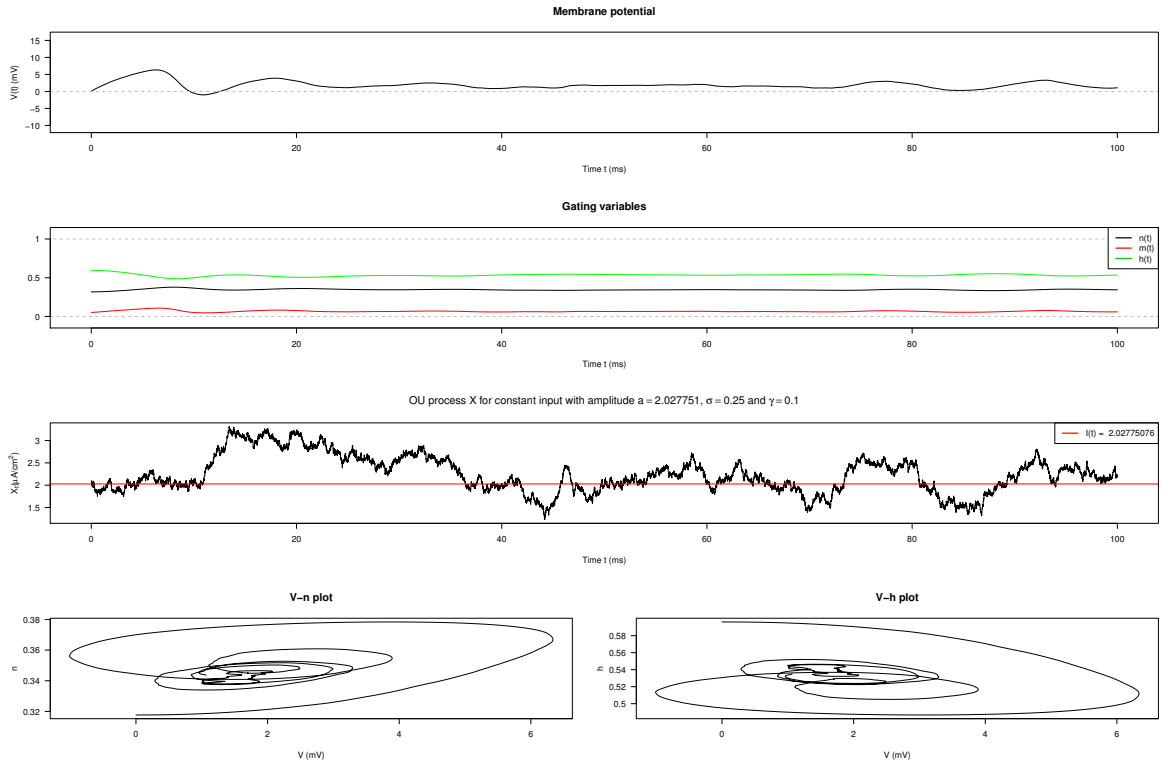


Figure 3.1: Simulation of constant input (2.1) embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.25$ and $\gamma = 0.1$. It is observable that even though the input is above $a_{fs} = 2.02775076$, no action potentials are generated

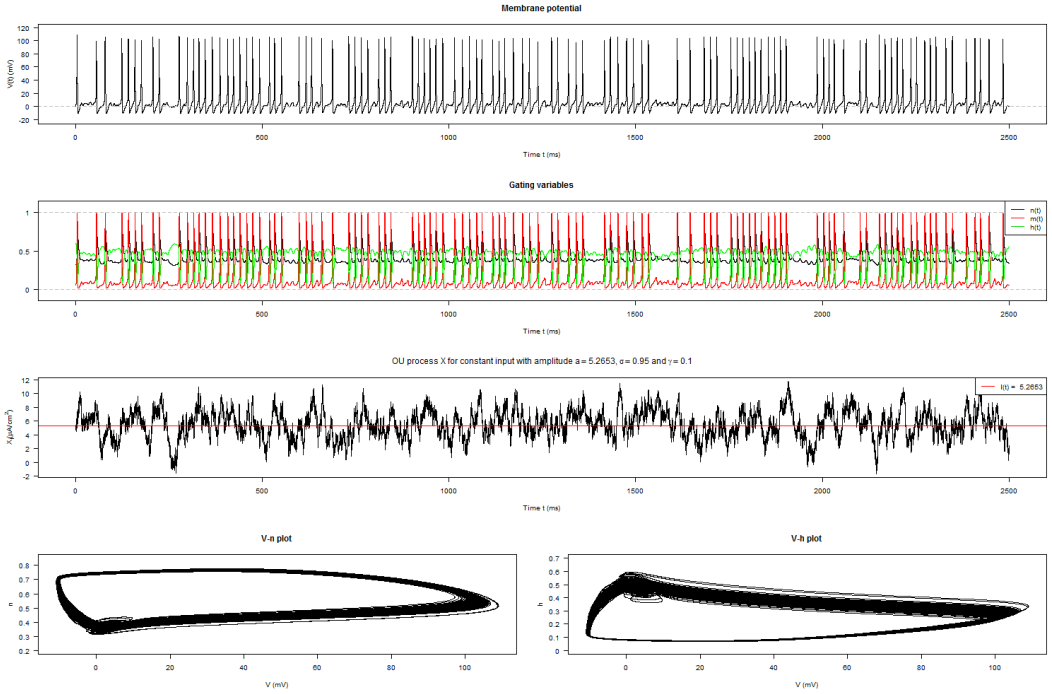


Figure 3.2: Simulation of constant input (2.1) embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.95$ and $\gamma = 0.1$.

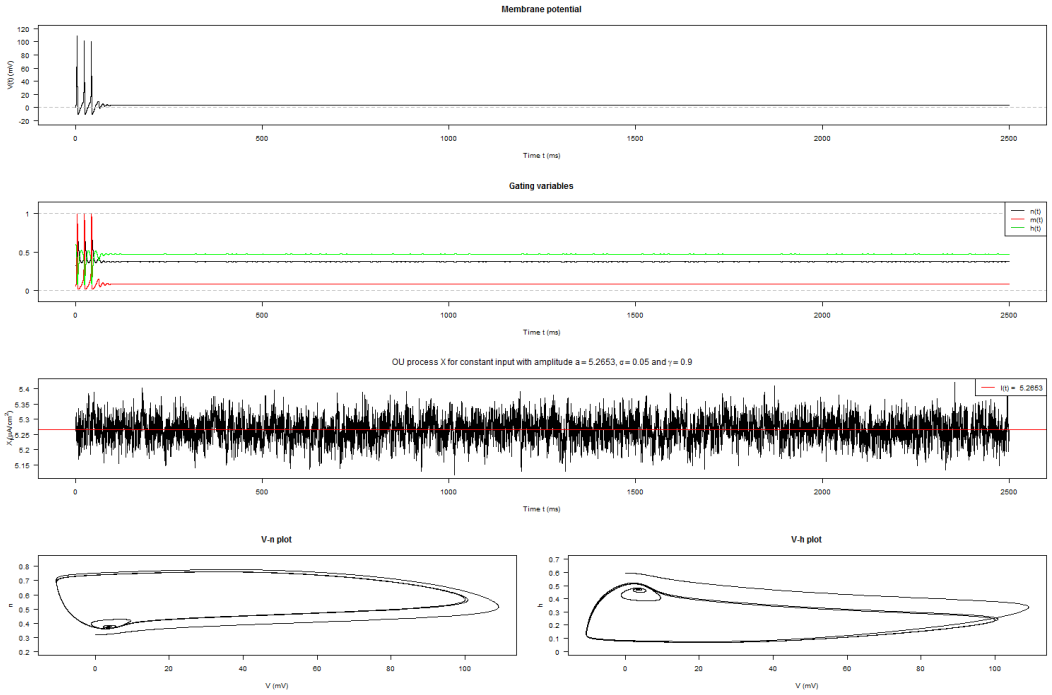


Figure 3.3: Simulation of constant input (2.1) embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.05$ and $\gamma = 0.9$.

The opposite of this behavior can be observed for an Ornstein-Uhlenbeck input with parameters $\sigma = 0.05$ and $\gamma \in \{0.75, 0.9\}$. Perhaps due to the initial conditions (1.12), spikes are generated just at the beginning of the simulation (see figure 3.3).

3.2.2 Ornstein-Uhlenbeck Process for Periodic Input

For the periodic input of section 2.2

$$I(t) = I_0(1 + \sin(2\,\pi\, f_s\, t)), \quad t \geq 0,$$

the integral in equation (3.5) (see $B.2$, line 14) is calculated during the simulation (see $B.2$, lines 26, 27). This makes it 4 times slower than the simulation of the Ornstein-Uhlenbeck process for a constant input.
Different values for I_0 and f_s (see table 3.2) were examined, for which the deterministic simulation shows a different behavior with regard to the ratios R_{bio} and R_{math}.

Ratios \ Parameter	$I_0 = 2.5$ $f_s = 15$	$I_0 = 2.5$ $f_s = 50$	$I_0 = 2.5$ $f_s = 75$	$I_0 = 2.5$ $f_s = 100$	$I_0 = 2.5$ $f_s = 140$	$I_0 = 2.5$ $f_s = 150$
R_{bio}	0	1	2/3	1/2	0.2464	1/3
R_{math}	1	1	1/3	1/2	0	1/3
Ratios \ Parameter	$I_0 = 5$ $f_s = 5$	$I_0 = 5$ $f_s = 20$	$I_0 = 5$ $f_s = 50$	$I_0 = 5$ $f_s = 95$	$I_0 = 5$ $f_s = 125$	
R_{bio}	0	2	1	2/3	1/2	
R_{math}	1	1	1	1/3	1/2	

Table 3.2: Ratios of the examined sinusoidal inputs (see figure 2.11 and figure 2.12).

Just some interesting graphics are viewed in this section.
In figure 3.4 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.5$, $\gamma = 0.5$ that is pretty similar to the embedded periodic input (2.2) with parameters $I_0 = 2.5$ and $f_s = 50$. The resulting membrane potential maintains the spiking behavior compared to the membrane potential simulated for the periodic input (2.2) with parameters $I_0 = 2.5$, $f_s = 50$. The only thing varying is the amplitude of the spikes due to the stochastic input. This can be observed in the V-n plot and the V-h plot. As a result, the behavior of a deterministic simulated membrane potential can be maintained by choosing the right parameters for the Ornstein-Uhlenbeck input.
In figure 3.5 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.75$, $\gamma = 0.25$. Due to $\gamma = 0.25$ the input is slower than the embedded periodic input (2.2) with parameters $I_0 = 2.5$ and $f_s = 15$. Even though $R_{bio} = 0$ for the membrane potential simulated for the deterministic periodic input (2.2) with parameters $I_0 = 2.5$, $f_s = 15$, the resulting membrane potential generates action potentials in an irregular manner. As a result, the stochastic input choosing the right parameters for the Ornstein-Uhlenbeck process can lead to spiking, whereas the appropriate deterministic, periodic input is not able to do so.

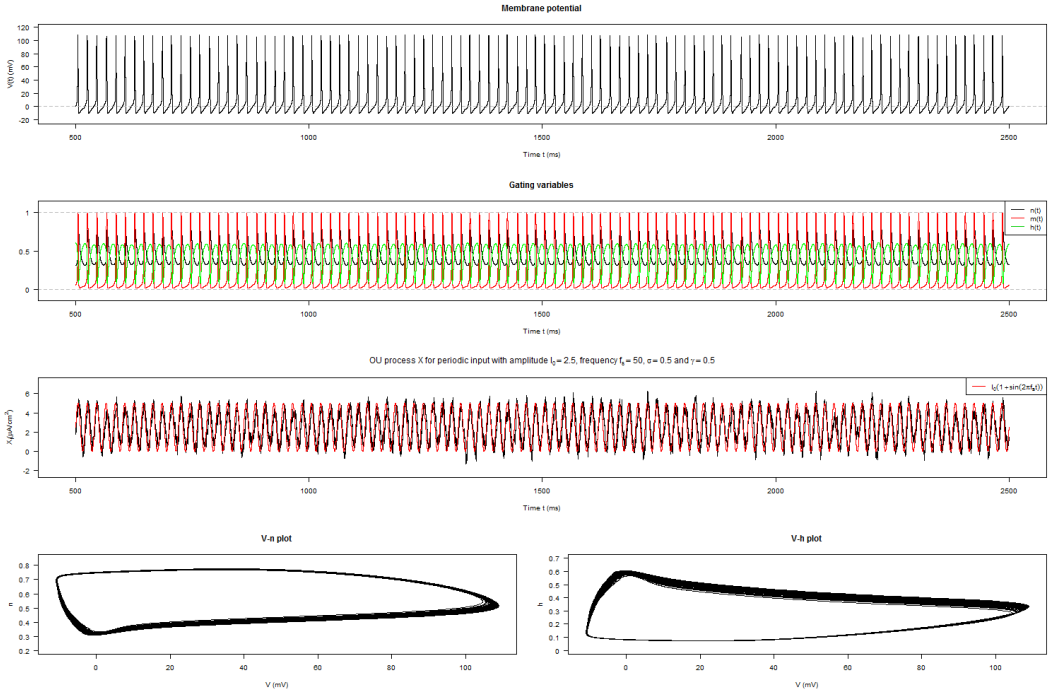


Figure 3.4: Simulation of sinusoidal input (2.2) with parameters $I_0 = 2.5$ and $f_s = 50$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.5$ and $\gamma = 0.5$.

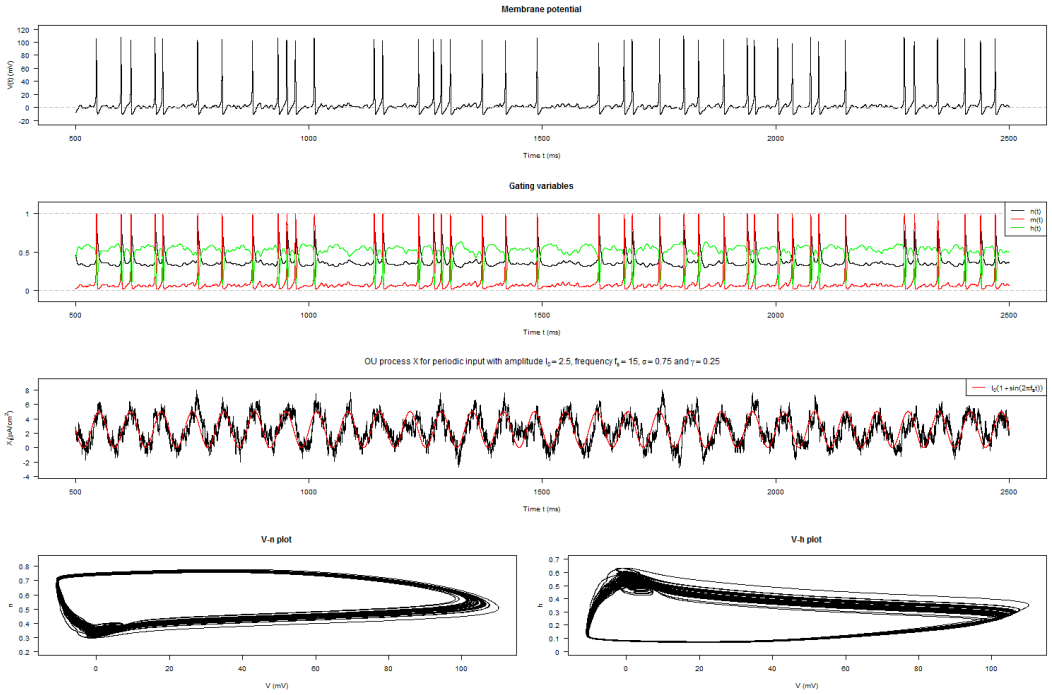


Figure 3.5: Simulation of sinusoidal input (2.2) with parameters $I_0 = 2.5$ and $f_s = 15$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.75$ and $\gamma = 0.25$.

In figure 3.6 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.05$, $\gamma = 0.25$ for an embedded periodic input (2.2) with parameters $I_0 = 2.5$ and $f_s = 75$. $R_{bio} = 2/3$ for the membrane potential simulated for the deterministic periodic input (2.2) with parameters $I_0 = 2.5$, $f_s = 75$. Due to the high frequency $f_s = 75$ and $\gamma = 0.25$ the input is not able to reach the amplitude of the embedded periodic input (2.2) with parameters $I_0 = 2.5$ and $f_s = 75$. Therefore the resulting membrane potential for the Ornstein-Uhlenbeck input behaves like the appropriate membrane potential for the deterministic periodic input (2.2) with parameters approximately $I_0 = 1.5$, $f_s = 75$ ($R_{bio} = 1/2$, see 2.11).

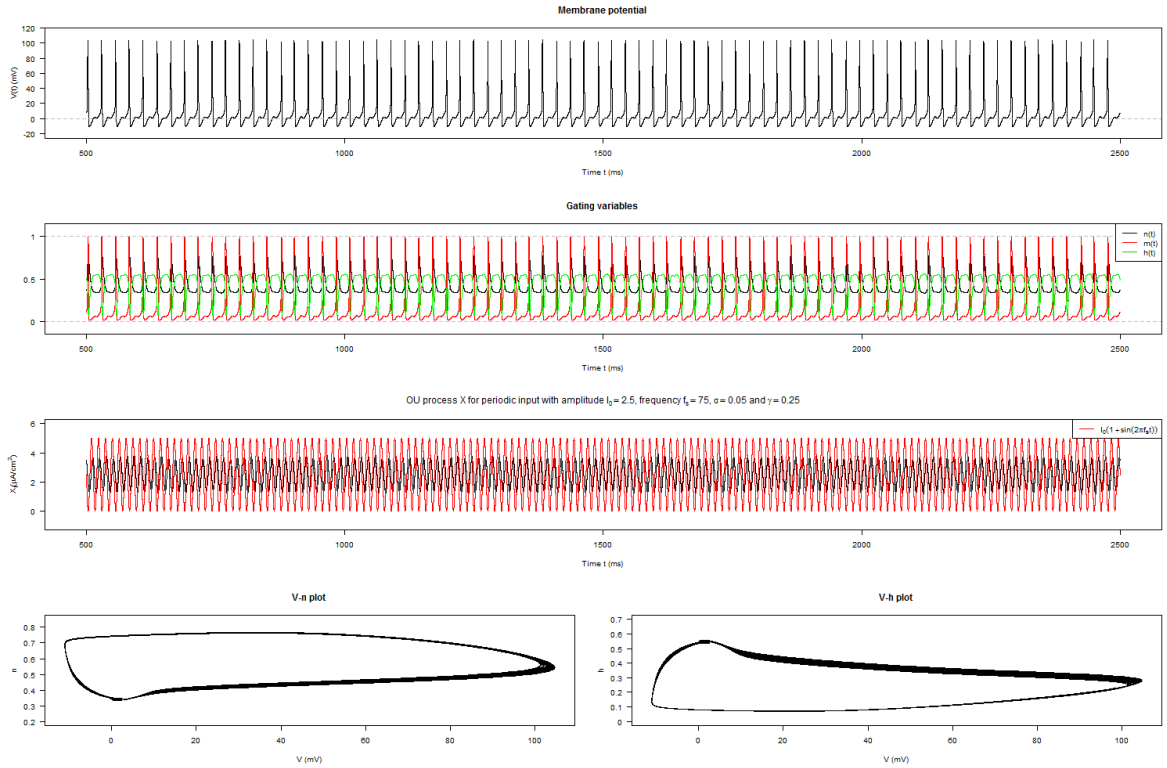


Figure 3.6: Simulation of sinusoidal input (2.2) with parameters $I_0 = 2.5$ and $f_s = 75$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.05$ and $\gamma = 0.25$.

In figure 3.7 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.5$, $\gamma = 0.75$ for an embedded periodic input (2.2) with parameters $I_0 = 5$ and $f_s = 5$. Even though $R_{bio} = 0$ for the membrane potential simulated for the deterministic periodic input (2.2) with parameters $I_0 = 5$, $f_s = 5$, the resulting membrane potential always generates a small spike train at the local maxima of the deterministic input in an irregular manner. In section 2.2 no ratios $R_{bio} > 2$ could be observed, but figure 3.7 shows that it is possible to create membrane potentials with $R_{bio} > 2$. As a result, the stochastic input choosing the right parameters for the Ornstein-Uhlenbeck process can lead to high frequently spiking, whereas the appropriate deterministic, periodic input is not able to do so.

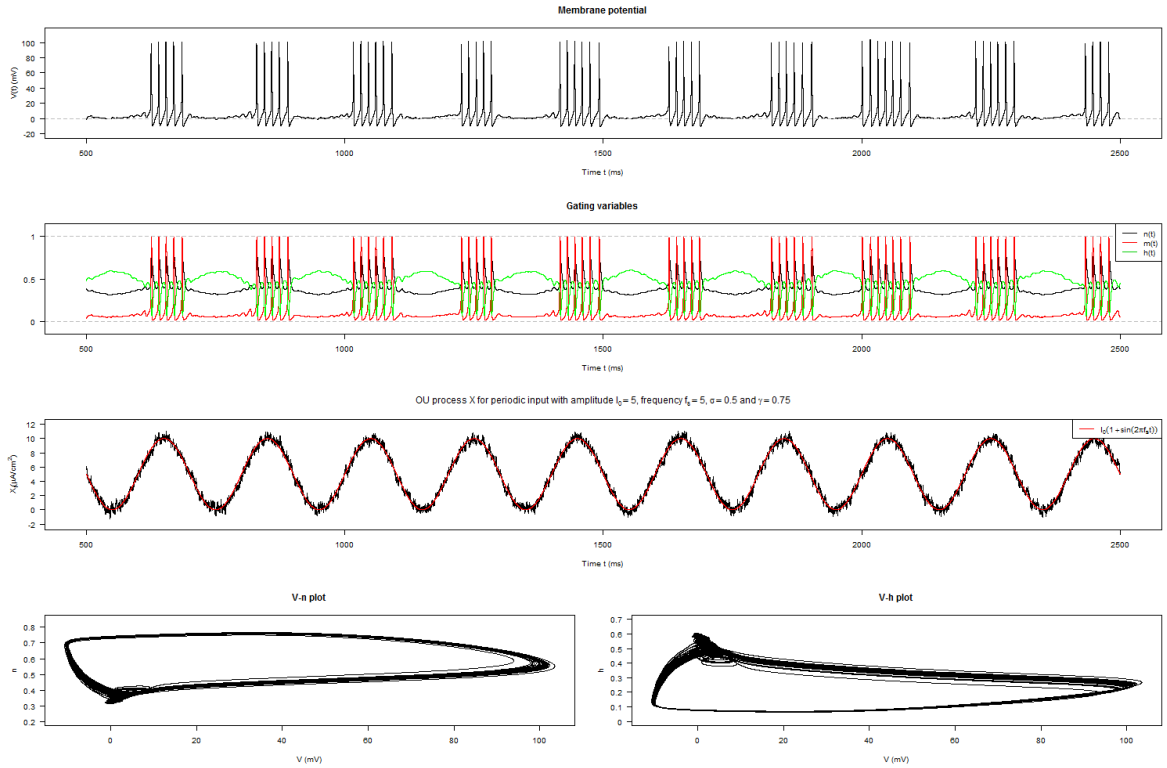


Figure 3.7: Simulation of sinusoidal input (2.2) with parameters $I_0 = 5$ and $f_s = 5$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.5$ and $\gamma = 0.75$.

3.2.3 Ornstein-Uhlenbeck Process for 1 ms Pulse Input

For the 1 ms pulse input of section 2.3

$$I(t) = \mathbb{1}_{(1,\infty)}(t) \sum_{m=0}^{f-1} a \cdot \mathbb{1}_{\left[m\frac{1000}{f}, m\frac{1000}{f}+1\right]}(t \bmod 1000), \quad t \geq 0,$$

the integral in equation (3.5) can be calculated (see $B.3$, lines 21, 22 and 26, 27), but if f is big a lot of cases have to queried (see $B.3$, lines 23-25):

$$\int_0^{t-s} \mathrm{e}^{-\gamma\nu}\gamma I(t-\nu)\mathrm{d}\nu = \begin{cases} a(1-\mathrm{e}^{-\gamma(t-s)}), & \text{if } I(t) = a, \\ 0, & \text{else.} \end{cases}$$

This makes it 1.5 times slower than the simulation of the Ornstein-Uhlenbeck process for a periodic input. Different values for a and f (see table 3.3) were examined, for which the deterministic simulation shows a different behavior with regard to the ratios P_{bio} and P_{math}. Just some interesting graphics are viewed in this section.

Ratios \ Parameter	$a = 8$ $f = 25$	$a = 8$ $f = 62$	$a = 8$ $f = 100$	$a = 8$ $f = 125$	$a = 8$ $f = 150$
P_{bio}	1	2/3	1/2	0.124	1/3
P_{math}	1	1/3	1/2	0	1/3

Ratios \ Parameter	$a = -18.5$ $f = 25$	$a = -18.5$ $f = 47$	$a = -18.5$ $f = 75$	$a = -18.5$ $f = 125$
P_{bio}	1	0.6383	1/2	0
P_{math}	1	0	1/2	1

Table 3.3: Ratios of the examined 1 ms pulse inputs (see figure 2.21 and figure 2.23)

In figure 3.8 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.05$, $\gamma = 0.75$ for an embedded pulse input (2.3) with parameters $a = 8$ and $f = 62$. $P_{bio} = 2/3$ for the membrane potential simulated for the deterministic pulse input (2.3) with parameters $a = 8$, $f = 62$. The membrane potential maintains the behavior of the deterministic pulse input ($P_{bio} = 2/3$), even though the Ornstein-Uhlenbeck input is not able to reach the amplitude $a = 8$ of the embedded pulse input (2.3). As a result, the behavior of a deterministic simulated membrane potential can be maintained choosing the right parameters for the Ornstein-Uhlenbeck input.

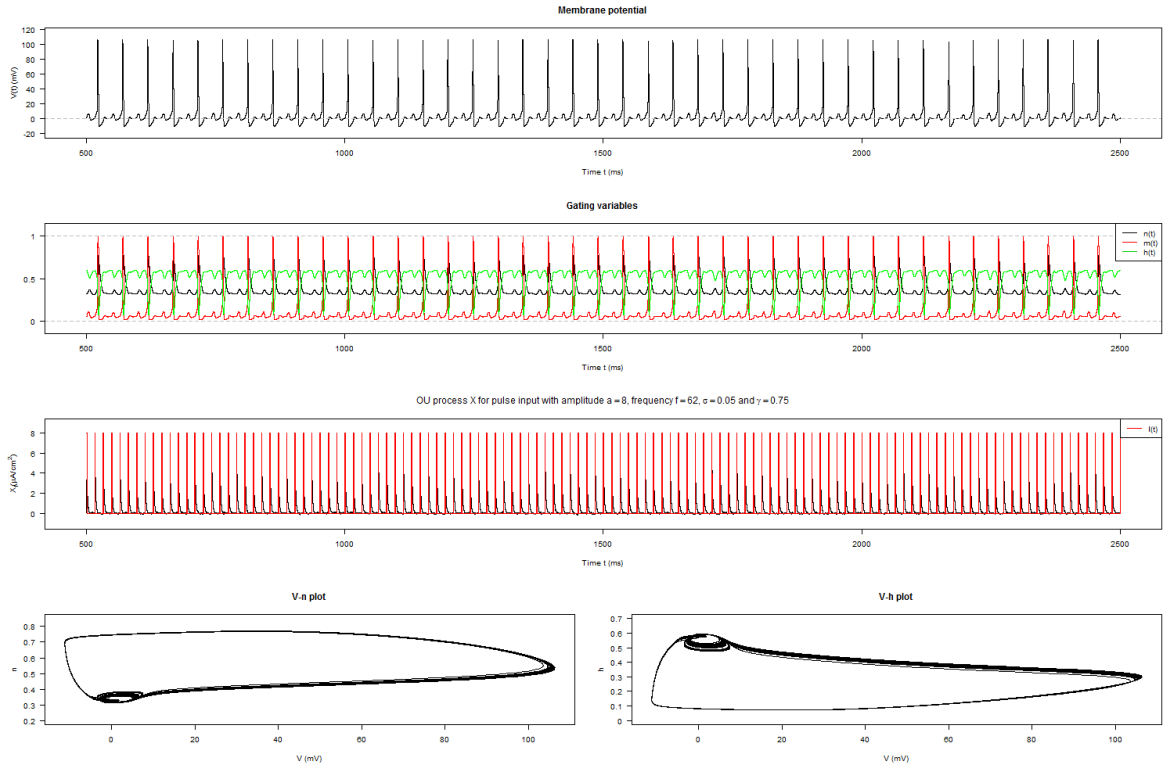


Figure 3.8: Simulation of a 1 ms pulse input (2.3) with parameters $a = 8$ and $f_s = 62$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.05$ and $\gamma = 0.75$.

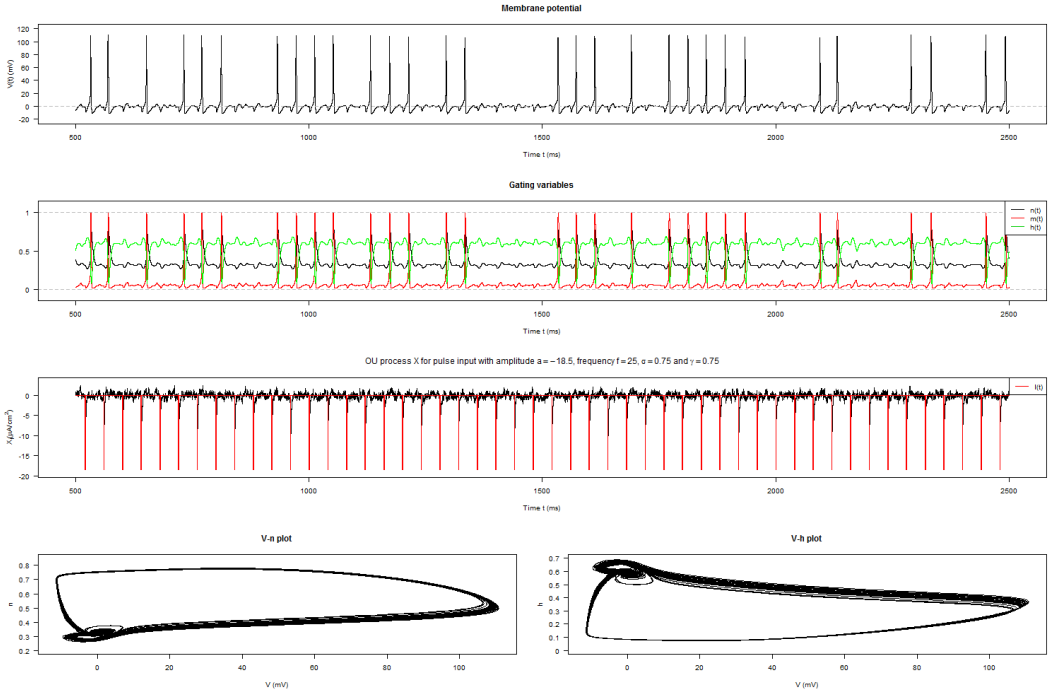


Figure 3.9: Simulation of a 1 ms pulse input (2.3) with parameters $a = -18.5$ and $f_s = 25$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.75$ and $\gamma = 0.75$.

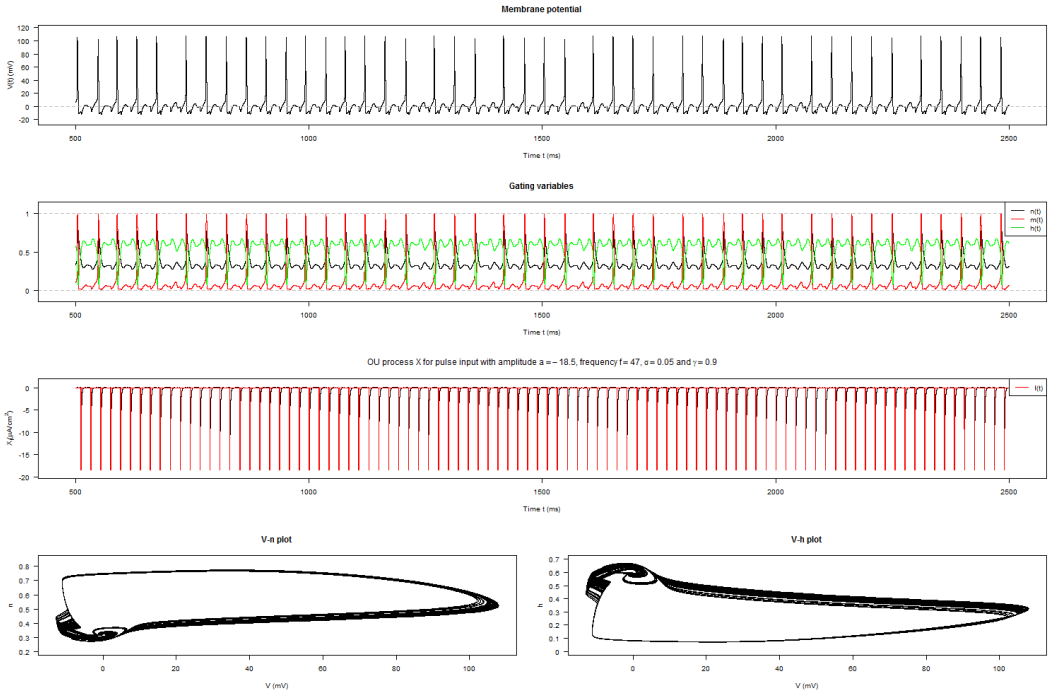


Figure 3.10: Simulation of a 1 ms pulse input (2.3) with parameters $a = -18.5$ and $f_s = 47$ embedded in an Ornstein-Uhlenbeck process with parameters $\sigma = 0.05$ and $\gamma = 0.9$.

In figure 3.9 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.75$, $\gamma = 0.75$ for an embedded pulse input (2.3) with parameters $a = -18.5$ and $f = 25$. Due to $\sigma = 0.75$ the membrane potential behaves irregular. Again, the amplitudes of the Ornstein-Uhlenbeck input are lower than the amplitudes of the embedded pulse input (2.3), but leads to spiking. In figure 3.10 there is an Ornstein-Uhlenbeck input with parameters $\sigma = 0.05$, $\gamma = 0.9$ for an embedded pulse input (2.3) with parameters $a = -18.5$ and $f = 47$. The membrane potential is irregular for the embedded pulse input (2.3) with parameters $a = -18.5$ and $f = 47$ ($P_{math} = 0$). Still the membrane potential for the Ornstein-Uhlenbeck input is irregular, but leaving out the single gabs, where no action potentials are generated, it looks like that for every two pulses of the Ornstein-Uhlenbeck input a spike is generated. This is again an example for the profitableness embedding signals into an stochastic process.

Appendix A

C Codes

A.1 Rating Constants and Hodgkin-Huxley Equations

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  // Implementation of the steady state (in)activation function in the HH model.
6  // Parameter:
7  //      V  :  Applied current (mV)
8  // Return:
9  //      Steady state (in)activation function
10
11 double ninf(double V){
12    double an;
13    double bn;
14    if(V == 10){
15       an = 1/10;
16    }else{
17       an = 0.01*(10-V)/(exp((10-V)/10)-1);
18    }
19    bn = 0.125*exp(-V/80);
20    return(an/(an+bn));
21 }
22
23 double minf(double V){
24    double am;
25    double bm;
26    if(V == 25){
27       am = 1;
28    }else{
29       am = 0.1*(25-V)/(exp((25-V)/10)-1);
30    }
31    bm = 4*exp(-V/18);
32    return(am/(am+bm));
33 }
34
35 double hinf(double V){
36    double ah;
37    double bh;
38    ah = 0.07*exp(-V/20);
39    bh = 1/(1+exp((30-V)/10));
```

```
40    return(ah/(ah+bh));
41 }
42
43 // Implementation of the voltage-dependent time constant in the HH model.
44 // Parameter:
45 //       V  :  Applied current (mV)
46 // Return:
47 //       Voltage-dependent time constant
48
49 double taun(double V){
50    double an;
51    double bn;
52    if(V == 10){
53       an = 1/10;
54    }else{
55       an = 0.01*(10-V)/(exp((10-V)/10)-1);
56    }
57    bn = 0.125*exp(-V/80);
58    return(1/(an+bn));
59 }
60
61 double taum(double V){
62    double am;
63    double bm;
64    if(V == 25){
65       am = 1;
66    }else{
67       am = 0.1*(25-V)/(exp((25-V)/10)-1);
68    }
69    bm = 4*exp(-V/18);
70    return(1/(am+bm));
71 }
72
73 double tauh(double V){
74    double ah;
75    double bh;
76    ah = 0.07*exp(-V/20);
77    bh = 1/(1+exp((30-V)/10));
78    return(1/(ah+bh));
79 }
80
81 // Implementation of the function "f(t,y)" for the classical Runge-Kutta method.
82 // Parameter:
83 //       y0,y1,y2,y3   :  Values of the dependent variables
84 //       I             :  External injected current (mA / cm^2)
85 //       *v,*n,*m,*h   :  Return values
86 //                        1. entry --> Evaluation of the membrane potential
87 //                        2. entry --> Evaluation of the function n()
88 //                        3. entry --> Evaluation of the function m()
89 //                        4. entry --> Evaluation of the function h()
90
91 void f(double y0, double y1, double y2, double y3, double I,
92        double *v, double *n, double *m, double *h){
93 // Using the values for the equilibrium potentials of Hodgkin and Huxley
94 // *v = I - 36*pow(y1,4)*(y0+12) - 120*pow(y2,3)*y3*(y0-115) - 0.3*(y0-10.613);
95 // Using the values for the equilibrium potentials of Izhikevich
96    *v = I - 36*pow(y1,4)*(y0+12) - 120*pow(y2,3)*y3*(y0-120) - 0.3*(y0-10.6);
97    *n = (ninf(y0)-y1)/taun(y0);
```

```
98    *m = (minf(y0)-y2)/taum(y0);
99    *h = (hinf(y0)-y3)/tauh(y0);
100 }
```

A.2 Runge-Kutta Method for Constant Input

```
1  // Implementation of the HH equations using the classical Runge-Kutta method.
2  // Return in .txt:
3  //         Matrix :  1. column --> Membrane potential
4  //                   2. column --> Function n()
5  //                   3. column --> Function m()
6  //                   4. column --> Function h()
7
8  main(){
9     // Setting parameters:
10    double v0 = 0;              // Initialization of the membrane potential (mV)
11    double x_max = 100;         // Final value of the independent variable (time)
12    double s = 0.005;           // Step size between 0 and x_max
13    int N = x_max/s;            // Number of steps
14    struct HH{
15       double v, n, m, h;
16    } V[5];                     // Initialization of the dependent variable
17    double amp[11];             // Initialization of the amplitude vector
18    int j;
19    for (j=0;j<11;j++){
20       amp[j] = 2.0277507 + j*0.00000001;
21    }
22    char text[25];              // A string to write in the file name
23
24    // Start of the simulations
25    int l;
26    for (l=0; l<11; l++){
27       // Setting initial values:
28       double x = 0;             // Simulation starts at time 0
29       V[0].v = v0;
30       V[0].n = ninf(v0);
31       V[0].m = minf(v0);
32       V[0].h = hinf(v0);
33
34       // Writing the files
35       FILE *fp;
36       sprintf(text,"C:/Amp=%.8fRinzel.txt", amp[l]);
37       fp = fopen(text,"w");
38       fprintf(fp,"%.10f %.10f %.10f %.10f\n", V[0].v,V[0].n,V[0].m,V[0].h);
39
40       // Evaluation loop for the classical Runge-Kutta method
41       int i=1;
42       while(i < N+2){
43          f(V[0].v, V[0].n, V[0].m, V[0].h, amp[l],
44            &V[1].v, &V[1].n, &V[1].m, &V[1].h);
45          f(V[0].v+s/2*V[1].v, V[0].n+s/2*V[1].n, V[0].m+s/2*V[1].m,
46            V[0].h+s/2*V[1].h, amp[l], &V[2].v, &V[2].n, &V[2].m, &V[2].h);
47          f(V[0].v+s/2*V[2].v, V[0].n+s/2*V[2].n, V[0].m+s/2*V[2].m,
48            V[0].h+s/2*V[2].h, amp[l], &V[3].v, &V[3].n, &V[3].m, &V[3].h);
49          f(V[0].v+s*V[3].v, V[0].n+s*V[3].n, V[0].m+s*V[3].m, V[0].h+s*V[3].h,
50            amp[l], &V[4].v, &V[4].n, &V[4].m, &V[4].h);
51
52          V[0].v = V[0].v+s*(V[1].v+2*(V[2].v+V[3].v)+V[4].v)/6;
```

```
53          V[0].n = V[0].n+s*(V[1].n+2*(V[2].n+V[3].n)+V[4].n)/6;
54          V[0].m = V[0].m+s*(V[1].m+2*(V[2].m+V[3].m)+V[4].m)/6;
55          V[0].h = V[0].h+s*(V[1].h+2*(V[2].h+V[3].h)+V[4].h)/6;
56
57          x = round(x*1000)/1000+s;
58          fprintf(fp,"%.10f %.10f %.10f %.10f\n", V[0].v,V[0].n,V[0].m,V[0].h);
59          i++;
60       }
61       fclose(fp);
62    }
63 }
```

A.3 Runge-Kutta Method for Periodic Input

```
1  // Implementation of a sinusoidal input
2  // Parameter:
3  //      t        : Time (ms)
4  //      a        : Amplitude of the sinusoidal input (mV)
5  //      fs       : Frequency of the sinusoidal input (Hz)
6  //
7  // Return:
8  //      Sinusoidal input at time t
9
10 double I(double t, double a, int fs){
11    return(a+a*sin(M_PI*fs*0.002*t));
12 }
13
14 // Implementation of the HH equations using the classical Runge-Kutta method.
15 // Return in .txt:
16 //         Matrix :  1. column --> Membrane potential
17 //                   2. column --> Function n()
18 //                   3. column --> Function m()
19 //                   4. column --> Function h()
20
21 main(){
22    // Setting parameters:
23    double v0 = 0;              // Initialization of the membrane potential (mV)
24    double x_max = 2500;        // Final value of the independent variable (time)
25    double s = 0.005;           // Increment between 0 and x_max
26    int N = x_max/s;            // Number of steps
27    struct HH{
28        double v, n, m, h;
29    } V[5];                     // Initialization of the dependent variable
30    double amp[38];             // Initialization of the amplitude vector
31    int j;
32    for (j=0;j<39;j++){
33       amp[j] = 1.2 + j*0.1;
34    }
35    int fs[150];                // Initialization of the frequency vector
36    int k;
37    for (k=0;k<150;k++){
38       fs[k] = 1+k;
39    }
40    char text[25];              // A string to write in the file name
41
42    // Start of the simulations
43    int l,p;
44    for (l=0; l<39; l++){
```

```
45        for (p=0;p<150;p++){
46           // Setting initial values:
47           double x = 0;        // Simulation starts at time 0
48           V[0].v = v0;
49           V[0].n = ninf(v0);
50           V[0].m = minf(v0);
51           V[0].h = hinf(v0);
52
53           // Writing the files
54           FILE *fp;
55           sprintf(text,"C:/FAP_Amp=%.1f,Freq=%i.txt", amp[l],fs[p]);
56           fp = fopen(text,"w");
57           fprintf(fp,"%.10f %.10f %.10f %.10f\n", V[0].v,V[0].n,V[0].m,V[0].h);
58
59           // Evaluation loop for the classical Runge-Kutta method
60           int i=1;
61           while(i < N+2){
62              f(V[0].v, V[0].n, V[0].m, V[0].h,
63                I(x,amp[l],fs[p]), &V[1].v, &V[1].n, &V[1].m, &V[1].h);
64              f(V[0].v+s/2*V[1].v, V[0].n+s/2*V[1].n, V[0].m+s/2*V[1].m,
65                V[0].h+s/2*V[1].h, I(x+s/2,amp[l],fs[p]), &V[2].v, &V[2].n,
66                &V[2].m, &V[2].h);
67              f(V[0].v+s/2*V[2].v, V[0].n+s/2*V[2].n, V[0].m+s/2*V[2].m,
68                V[0].h+s/2*V[2].h, I(x+s/2,amp[l],fs[p]), &V[3].v, &V[3].n,
69                &V[3].m, &V[3].h);
70              f(V[0].v+s*V[3].v, V[0].n+s*V[3].n, V[0].m+s*V[3].m,
71                V[0].h+s*V[3].h, I(x+s,amp[l],fs[p]), &V[4].v, &V[4].n,
72                &V[4].m, &V[4].h);
73
74              V[0].v = V[0].v+s*(V[1].v+2*(V[2].v+V[3].v)+V[4].v)/6;
75              V[0].n = V[0].n+s*(V[1].n+2*(V[2].n+V[3].n)+V[4].n)/6;
76              V[0].m = V[0].m+s*(V[1].m+2*(V[2].m+V[3].m)+V[4].m)/6;
77              V[0].h = V[0].h+s*(V[1].h+2*(V[2].h+V[3].h)+V[4].h)/6;
78
79              x = round(x*1000)/1000+s;
80              fprintf(fp,"%.10f %.10f %.10f %.10f\n",V[0].v,V[0].n,V[0].m,V[0].h);
81              i++;
82           }
83           fclose(fp);
84        }
85     }
86  }
```

A.4 Runge-Kutta Method for 1 ms Pulse Input

```
1  // Implementation of a 1 ms pulse input
2  // Parameter:
3  //      t        : Time (ms)
4  //      a        : Amplitude of the pulse input (mV)
5  //      f        : Frequency of the pulse input (Hz)
6  //
7  // Return:
8  //      1 ms pulse input at time t <= 3000
9
10 double I(double t, double amp, double f){
11    int l;
12    for (l=1;l<f+1;l++){
13       if(t>=(1000/f)*l && t<=(1000/f)*l+1){
```

```
14          return(a);
15       }
16       if(t>=1000+(1000/f)*l && t<=1000+(1000/f)*l+1){
17          return(a);
18       }
19       if(t>=2000+(1000/f)*l && t<=2000+(1000/f)*l+1){
20          return(a);
21       }
22    }
23    return(0);
24 }
25
26 // Implementation of the HH equations using the classical Runge-Kutta method.
27 // Return in .txt:
28 //         Matrix :  1. column --> Membrane potential
29 //                   2. column --> Function n()
30 //                   3. column --> Function m()
31 //                   4. column --> Function h()
32
33 main(){
34    // Setting parameters:
35    double v0 = 0;              // Initialization of the membrane potential (mV)
36    double x_max = 2500;        // Final value of the independent variable (time)
37    double s = 0.005;           // Increment between 0 and x_max
38    int N = x_max/s;            // Number of steps
39    struct HH{
40        double v, n, m, h;
41    } V[5];                     // Initialization of the dependent variable
42    double amp[56];             // Initialization of the amplitude vector
43    int j;
44    for (j=0;j<21;j++){
45        amp[j] = 5.5 + j*0.1;
46    }
47    for (j=0;j<35;j++){
48        amp[j+21] = -20 + j*0.1;
49    }
50    int fs[150];                // Initialization of the frequency vector
51    int k;
52    for (k=0;k<150;k++){
53        fs[k] = 1+k;
54    }
55    char text[25];              // A string to write in the file name
56
57    // Start of the simulations
58    int l,p;
59    for (l=0; l<56; l++){
60       for (p=0;p<150;p++){
61          //Setting initial values:
62          double x = 0;         // Simulation starts at time 0
63          V[0].v = v0;
64          V[0].n = ninf(v0);
65          V[0].m = minf(v0);
66          V[0].h = hinf(v0);
67
68          // Writing the files
69          FILE *fp;
70          sprintf(text,"C:/Amp=%.1f,Freq=%iIz.txt", amp[l],fs[p]);
71          fp = fopen(text,"w");
```

```
72          fprintf(fp,"%.10f %.10f %.10f %.10f\n", V[0].v,V[0].n,V[0].m,V[0].h);
73
74          // Evaluation loop for the classical Runge-Kutta method
75          int i=1;
76          while(i < N+2){
77             f(V[0].v, V[0].n, V[0].m, V[0].h, I(x,amp[l],fs[p]),
78               &V[1].v, &V[1].n, &V[1].m, &V[1].h);
79             f(V[0].v+s/2*V[1].v, V[0].n+s/2*V[1].n, V[0].m+s/2*V[1].m,
80               V[0].h+s/2*V[1].h, I(x+s/2,amp[l],fs[p]), &V[2].v, &V[2].n,
81               &V[2].m, &V[2].h);
82             f(V[0].v+s/2*V[2].v, V[0].n+s/2*V[2].n, V[0].m+s/2*V[2].m,
83               V[0].h+s/2*V[2].h, I(x+s/2,amp[l],fs[p]), &V[3].v, &V[3].n,
84               &V[3].m, &V[3].h);
85             f(V[0].v+s*V[3].v, V[0].n+s*V[3].n, V[0].m+s*V[3].m,
86               V[0].h+s*V[3].h, I(x+s,amp[l],fs[p]), &V[4].v, &V[4].n,
87               &V[4].m, &V[4].h);
88
89             V[0].v = V[0].v+s*(V[1].v+2*(V[2].v+V[3].v)+V[4].v)/6;
90             V[0].n = V[0].n+s*(V[1].n+2*(V[2].n+V[3].n)+V[4].n)/6;
91             V[0].m = V[0].m+s*(V[1].m+2*(V[2].m+V[3].m)+V[4].m)/6;
92             V[0].h = V[0].h+s*(V[1].h+2*(V[2].h+V[3].h)+V[4].h)/6;
93
94             x = round(x*1000)/1000+s;
95             fprintf(fp,"%.10f %.10f %.10f %.10f\n",V[0].v,V[0].n,V[0].m,V[0].h);
96             i++;
97          }
98          fclose(fp);
99       }
100   }
101 }
```

A.5 Euler Method for Ornstein-Uhlenbeck Inputs

```
1  // Implementation of the Hodgkin-Huxley equation using the Euler method
2  // Return in .txt:
3  //         Matrix : 1. column --> OU input
4  //                  2. column --> membrane potential
5  //                  3. column --> function n()
6  //                  4. column --> function m()
7  //                  5. column --> function h()
8
9  main(){
10    // Setting parameters:
11    double v0 = 0;              // Initialization of the membrane potential (mV)
12    double x_max = 2500;        // Final value of the independent variable (time)
13    double s = 0.005;           // Increment between 0 and x_max
14    int N = x_max/s;            // Number of steps
15    struct HH{
16        double v, n, m, h;
17    } V[2];                     // Initialization of the dependent variable
18    double amp =  8;            // Initialization of the amplitude
19    int f_s[6];                 // Initialization of the frequency vector
20    f_s[0] = 0;
21    f_s[1] = 25;
22    f_s[2] = 62;
23    f_s[3] = 100;
24    f_s[4] = 125;
25    f_s[5] = 150;
```

```
26    double gamma[6];          // Initialization of the gamma vector
27    gamma[0] = 0;
28    gamma[1] = 0.1;
29    gamma[2] = 0.25;
30    gamma[3] = 0.5;
31    gamma[4] = 0.75;
32    gamma[5] = 0.9;
33    double sigma[6];          // Initialization of the sigma value
34    sigma[0] = 0;
35    sigma[1] = 0.05;
36    sigma[2] = 0.25;
37    sigma[3] = 0.5;
38    sigma[4] = 0.75;
39    sigma[5] = 0.95;
40    char text[100];           // A string to write in the file name
41    char text1[100];          // A string to write in the file name
42
43    // Start of the simulations
44    int l;
45    for (l=1; l<6; l++){                  // Sigma loop
46       int p;
47       for (p=1;p<6;p++){                 // Gamma loop
48       int q;
49          for (q=1;q<6;q++){              // Frequency loop
50             // read data of OU Input (Constant, Period or Pulse Input)
51             FILE *datei;
52             float test;                 // Parameter to write in the input
53             //sprintf(text1,"C:/OU_const_amp=%.4f,sigma=%.0f,gamma=%.0f.txt",
54             //        amp,sigma[l]*100,gamma[p]*100);
55             //sprintf(text1,"C:/
56             //        OU_period_amp=%.1f,f_s=%i,sigma=%.0f,gamma=%.0f.txt",
57             //        amp,f_s[q],sigma[l]*100,gamma[p]*100);
58             sprintf(text1,"C:/
59                     OU_pulse_amp=%.1f,f_s=%i,sigma=%.0f,gamma=%.0f.txt",
60                     amp,f_s[q],sigma[l]*100,gamma[p]*100);
61             datei = fopen(text1,"r");
62
63             // Setting initial values:
64             double x = 0;               // Simulation starts at time 0
65             V[0].v = v0;
66             V[0].n = ninf(v0);
67             V[0].m = minf(v0);
68             V[0].h = hinf(v0);
69             fscanf(datei, "%f", &test);
70
71             // Writing the files (Constant, Period or Pulse Input)
72             FILE *fp;
73             //sprintf(text,"C:/Const/Amp=%.4f,sigma=%.0f,gamma=%.0f.txt",
74             //        amp,sigma[l]*100,gamma[p]*100);
75             //sprintf(text,"C:/Period/
76             //        Amp=%.0f,f_s=%i,sigma=%.0f,gamma=%.0f.txt",
77             //        amp,f_s[q],sigma[l]*100,gamma[p]*100);
78             sprintf(text,"C:/Pulse/Amp=%.0f,f_s=%i,sigma=%.0f,gamma=%.0f.txt",
79                     amp,f_s[q],sigma[l]*100,gamma[p]*100);
80             fp = fopen(text,"w");
81             fprintf(fp,"%f %.10f %.10f %.10f %.10f\n",test,
82                     V[0].v,V[0].n,V[0].m,V[0].h);
83
```

```
84              // Evaluation loop for the Euler method
85              int i=1;
86              while(i < N+2){
87                 fscanf(datei, "%f", &test);
88                 f(V[0].v, V[0].n, V[0].m, V[0].h, test,
89                   &V[1].v, &V[1].n, &V[1].m, &V[1].h);
90
91                 V[0].v = V[0].v+s*V[1].v;
92                 V[0].n = V[0].n+s*V[1].n;
93                 V[0].m = V[0].m+s*V[1].m;
94                 V[0].h = V[0].h+s*V[1].h;
95
96                 x = round(x*1000)/1000+s;
97                 fprintf(fp,"%f %.10f %.10f %.10f %.10f\n",
98                         test,V[0].v,V[0].n,V[0].m,V[0].h);
99
100                i++;
101             }
102             fclose(fp);
103             fclose(datei);
104          }
105       }
106    }
107 }
```

Appendix B

R Codes

B.1 Ornstein-Uhlenbeck Process for Constant Input

```
### Setting parameters
set.seed(12345);                          # Seed
s <- 0.005;                               # Step size
x.max <- 100;                             # End of time vector
time <- seq(0,x.max,s);                   # Time vector

### Parameters for OU process
amp <- 2.02775076;                        # Amplitude of the constant input
gamma <- c(0.1,0.25,0.5,0.75,0.9);        # Speed of reversion
sigma <- c(0.05,0.25,0.5,0.75,0.95);      # Volatility

### Exact simulation
X <- numeric(length(time));               # Initialization of the OU process
X[1] <- amp;
for (k in 1:length(sigma)){
  for (j in 1:length(gamma)){
    for (i in 2:length(time)){
       X[i] <- rnorm(1, X[i-1]*exp(-gamma[j]*s)+amp*(1-exp(-gamma[j]*s)),
                        sqrt(sigma[k]^2/(2*gamma[j])*(1-exp(-2*gamma[j]*s))) )
    }
    ### Save the data
    write.table(round(X,6), file = paste("C:/OU_const_amp=",amp,
                ",sigma=",sigma[k]*100,",gamma=",gamma[j]*100,".txt",sep=""),
                col.names=FALSE, row.names=FALSE);
 }
}
```

B.2 Ornstein-Uhlenbeck Process for Periodic Input

```
### Setting parameters
set.seed(12345);                          # Seed
s <- 0.005;                               # Step size
x.max <- 2500;                            # End of time vector
time <- seq(0,x.max,s);                   # Time vector

### Parameters for OU process
i0 <- 2.5;                                # Amplitude of the periodic input
f_s <- c(15,50,75,100,140,150);           # Frequency of the periodic input
gamma <- c(0.1,0.25,0.5,0.75,0.9);        # Speed of reversion
```

```
sigma <- c(0.05,0.25,0.5,0.75,0.95);   # Volatility

#### Function for the integration
f2 <- function(x) return( exp(-g*x)*i0*(1+sin(pi*(fs*0.002)*(t-x)))*g );

### Exact simulation
X <- numeric(length(time));
X[1] <- i0;
for (m in 1:length(f_s)){
  fs <- f_s[m];
  for (k in 1:length(sigma)){
    for (j in 1:length(gamma)){
      g <- gamma[j];
      for (i in 2:length(time)){
        t <- time[i];
        X[i] <- rnorm(1, X[i-1]*exp(-g*s)+integrate(f2,lower=0,upper=s)$value,
                         sqrt(sigma[k]^2/(2*g)*(1-exp(-2*g*s))) )
      }

      ### Save the data
      write.table(round(X,6), file = paste("C:/OU_period_amp=",i0,
                  ",f_s=",f_s[m],",sigma=",sigma[k]*100,
                  ",gamma=",gamma[j]*100,".txt",sep=""),
                  col.names=FALSE, row.names=FALSE);
    }
  }
}
```

B.3 Ornstein-Uhlenbeck Process for 1 ms Pulse Input

```
### Setting parameters
set.seed(12345);                          # Seed
s <- 0.005;                               # Step size
x.max <- 2500;                            # End of time vector
time <- seq(0,x.max,s);                   # Time vector

### Parameters for OU process
amp <- 8;                                 # Amplitude of the pulse input
f_s <- c(25,62,100,125,150);              # Frequency of the pulse input
gamma <- c(0.1,0.25,0.5,0.75,0.9);        # Speed of reversion
sigma <- c(0.05,0.25,0.5,0.75,0.95);      # Volatility

### Exact simulation
X <- numeric(length(time));
for (m in 1:length(f_s)){
  for (k in 1:length(sigma)){
    for (j in 1:length(gamma)){
      g <- gamma[j];
      for (i in 2:length(time)){
        t <- time[i];
        X[i] <- rnorm(1, X[i-1]*exp(-g*s),
                         sqrt(sigma[k]^2/(2*g)*(1-exp(-2*g*s))));
        if(t>1){
          for (l in 0:(f_s[m]-1)){
            if((t%%1000)>=((1000/f_s[m])*l) && (t%%1000)<=(1000/f_s[m])*l+1){
              X[i] <- rnorm(1, X[i-1]*exp(-g*s)+amp*(1-exp(-g*s)),
                               sqrt(sigma[k]^2/(2*g)*(1-exp(-2*g*s))) )
            }
```

```
29            }
30          }
31        }
32
33        ### Save the data
34        write.table(round(X,6), file = paste("C:/OU_pulse_amp=",amp,
35                    ",f_s=",f_s[m],",sigma=",sigma[k]*100,
36                    ",gamma=",gamma[j]*100,".txt",sep=""),
37                    col.names=FALSE, row.names=FALSE);
38      }
39    }
40 }
```

Bibliography

[1] Aihara, Kazuyuki, and Matsumoto, Gen, and Ikegaya, Yuhji. "Periodic and Non-periodic Responses of a Periodically Forced Hodgkin-Huxley Oscillator."
Journal of Theoretical Biology 109 (1984): 249-269.

[2] Cole, K.S., and Antosiewicz, H.A., and Rabinowitz, P. "Automatic Computation of Nerve Excitation."
Journal of the Society for Industrial and Applied Mathematics 3 (1955): 153-172.

[3] Conway, John B. *Functions of One Complex Variable.*
New York Inc., Springer-Verlag, 1973, 2nd edition (1978).

[4] Fitzhugh, R., and Antosiewicz, H.A. "Automatic Computation of Nerve Excitation – Detailed Corrections and Additions."
Journal of the Society for Industrial and Applied Mathematics 7 (1959): 447-458.

[5] Fohlmeister, Jürgen F. and Adelman, William J., and Poppele, Richard E. "Excitation Properties of the Squid Axon Membrane and Model Systems with Current Stimulation."
Biophysical Journal 30 (1980): 79-98.

[6] Hanke-Bourgeois, Martin. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens.*
Wiesbaden, Vieweg+Teubner, 2002, 3rd edition (2009).

[7] Hassard, Brian. "Bifurcation of Periodic Solutions of the Hodgkin-Huxley Model for the Squid Giant Axon."
Journal of Theoretical Biology 71 (1978): 401-420.

[8] Hodgkin, A.L., and Huxley, A.F. "A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve."
Journal of Physiology 117 (1952): 500-544.

[9] Holden, A.V. "The Response of Excitable Membrane Models to a Cyclic Input."
Biological Cybernetics 21 (1976): 1-7.

[10] Höpfner, R. and Kutoyants, Y. "Estimating Discontinuous Periodic Signals in a Time Inhomogeneous Diffusion."
arXiv:0903.5061v2 [math.ST] (2010).

[11] Iacus, Stefano M. *Simulation and Inference for Stochastic Differential Equations: With R Examples.*
New York, Springer Science+Business Media, 2008.

[12] Izhikevich, Eugene M. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting.*
Cambridge, The MIT Press, 2007.

[13] Karatzas, Ioannis and Shreve, Steven E. *Brownian Motion and Stochastic Calculus.*
New York, Springer-Verlag, 1988.

[14] Königsberger, Konrad. *Analsis 1.*
Berlin Heidelberg, Springer-Verlag, 1990, 6th edition (2004).

[15] Rinzel, John, and Miller, Robert N. "Numerical Calculation of Stable and Unstable Periodic Solutions to the Hodgkin-Huxley Equations."
Mathematical Biosciences 49 (1980): 27-59.