Université de Liège - Faculté des Sciences Appliquées

Development of a bifurcation identification interface applied to the analysis of neuronal excitability

Master thesis

Travail de fin d'études réalisé en vue de l'obtention du grade de

Master Ingénieur Civil en électricité (option électronique)

par Gillis Thibault

Année académique 2016-2017

Table of contents

Abstract

This master thesis concerns the implementation of a novel, computationally efficient bifurcation numerical analysis interface in the Julia compiled programming language. The interface involves the use of the well-known bisection or Newton-Raphson methods in order to locate bifurcations in the neuron models, as well as the use of numerical approximation methods of Jacobian matrices through forward numerical differentiation of the system's equations.

The interface that is built aims at the identification of the bifurcations in neuron models in order to determine their excitability type. A recent paper-motivated canonical model is chosen as an example to which the interface can be applied as a proof of concept. This numerical analysis of the example model outputs results that highlight the importance of dynamical analysis of neuron models, i.e. analysis over a range of time-scale parameters, versus the more common static analysis of models through the visual inspection of their phase plane representation.

Normal form identification based on visual inspection only is at considerable risk that the original system is identified to may not be the correct one. The results obtained through the use of this interface on a two-dimensional therefore motivate the need for extensive numerical analysis of original high-dimensional neuron models for various values of time-scale separation in order to reliably identify the bifurcation normal form that they can be reduced to.

Introduction

The study of neuronal excitability consists in the observation and identification of neuron behaviour in various conditions. Neurons have indeed been experimentally identified to display different types of activity, which include quiescence, spiking, or bursting activity. Quiescence consists in a lack of spontaneous activity, but does not restrict the ability to react to external stimuli. Spiking consists in spontaneous oscillatory activity, i.e. constantly alternating modifications of the neuron membrane potential. Bursting is a combination of the two previous phenomena corresponding to the display of short bursts of spiking activity separated by periods of quiescence. Neuron spiking behaviour is classified into types, or classes of excitability. The excitability type of a neuron defines the current-frequency relationship of its spiking activity, and therefore its possible function in a biological neuronal network. For example, a neuron that switches from quiescence to spiking and displays a constant frequency in its oscillatory behaviour cannot be used as a frequency encoder in a network.

In the context of neuron modelling as non-linear systems, the switch between rest and spiking activity corresponds to a bifurcation in the system. Bifurcations have been studied in the fields of mathematics since the works of Henri Poincaré in 1885 and have been subsequently applied to neuron modelling since the works of Ermentrout in 1986. They consist in transitions in non-linear systems between configurations of varying stability, e.g. from global stability to global instability, due to a shift in the nature of the different attractors. In neuronal systems, the attractors involved in bifurcations display properties local to the bifurcation that allow direct identification of the excitability type of the neuron undergoing the bifurcation, thus allowing to bypass the otherwise expensive process of experimentally measuring their current-frequency relationship.

The problem of the determination of a neuron excitability type can therefore be reduced to the localisation and identification of the bifurcation displayed by its equivalent mathematical model. While current solutions to the identification of bifurcations in neuron models exist in the form of a number of individual software implementations such as XPPAUT or NEURON, these are implemented in many different programming languages such as Fortran or Matlab and make a coherent numerical analysis of a neuron model a cumbersome task.

The main goal of this master thesis is to implement a novel, computationally efficient bifurcation numerical analysis interface for the study of neuron excitability. The proposed interface is entirely written in the Julia programming language. The method proposed as a solution to the bifurcation identification problem involves the use of well-known numerical non-linear system solving algorithms such as the bisection or Newton-Raphson methods. These algorithms are used to locate the bifurcations involved in the initiation of neuron spiking activity as a function of the excitatory current. Numerical differentiation methods are then used in order to determine local properties of the bifurcations of the systems through the computation of the system Jacobian matrix at the location of the stable equilibrium involved in the bifurcation.

The development of the interface covered in this document aims at motivating the use of a single-language and self-contained interface as a means of solving this family of problems. These constraints on the development of the interface allow its logic cores to be easily identified and maintained in terms of both stability and accuracy.

The main programming challenge is the development of the bifurcation search and identification algorithm using a previously unknown programming language. Three different implementations are built in order to explore different ways of solving the problem. The three implementations are evaluated on their computational performance as well as their accuracy.

A recent paper-motivated canonical neuron model is chosen as an example to which the interface can be applied as a proof of concept. This numerical analysis of the example model through rigorous identification of its bifurcations outputs results that highlight the importance of dynamical analysis of neuron models, i.e. analysis over a range of time-scale parameters, versus the more common static analysis of models through the visual inspection of their phase plane representation.

In particular, type II excitability is numerically identified for very strong time-scale separation at a location that is commonly identified to type I excitability through visual inspection. This is due to the fact that the shapes of the nullclines in this region of the system parameters look very much like those of a canonical type I bifurcation-based neuron model, although they actually are fundamentally different at a scale much smaller than that of visual inspection. The growth of a type I excitability zone is subsequently observed in this region for increasing values of the time-scale coupling factor, therefore hinting at the fact that dynamical analysis of neuron models is required in order to determine their exact behaviour, as this behaviour may freely switch between type I and type II excitability according to the models time-related parameters.

This raises controversy in regards to the widely-accepted method of model reduction through visual identification of the neuron behaviour to normal forms of bifurcations. As long as this identification process is only motivated by simple visual inspection, it is bound to output erroneous results in a number of visually confusing cases.

For these reasons, the results obtained through the use of this interface on a two-dimensional motivate the need for extensive numerical analysis of original high-dimensional neuron models for various values of time-scale separation in order to reliably identify the bifurcation normal form that they can be reduced to.

This document is structured in three chapters : context, methodology, and scientific output. The chapters are organized in such a way that reading them in sequence should equip the reader with sufficient theoretical background and knowledge of the software implementations that they may understand the source and validity of the scientific output of the methods in use. Extensive coverage of textbook knowledge, however, has been avoided in order to focus on the work performed and the theoretical notions that were required to be assimilated during the elaboration of this master thesis.

Chapter 1 : Elements of experimental and computational neuroscience

This first chapter sets the context of this master thesis. It covers the minimal amount of theoretical background required in order to understand the problem and the software implementation of its solution covered in the second chapter, as well as the results obtained through the application of the interface to the canonical model covered in the third chapter.

In the first part of this chapter, the physiological structure and main properties of neurons, as well as the original experimental definition of excitability types is recalled. While a greater part of the extensive biological description of neurons and all their components has been avoided, special focus has been put on the description of action potential generation and ion channels. These are the most defining features of neurons, and the most important factors in their external behaviour as seen from a computational point of view. Furthermore, the identification of these mechanisms are necessary in order to justify conductance-based modelling, which underlies most neuron mathematical modelling.

In the second part of this chapter, the modelling of neurons as non-linear systems is presented. Using this mathematical representation, the switch between rest and sustained activity in neurons corresponds to a bifurcation, i.e. a shift in the nature of the different attractors characterising in the system. The basics of non-linear systems analysis and the characteristics of the bifurcations of interest for this work are briefly covered. They are applied to the visual analysis of the classical Hodgkin-Huxley model and the canonical transcritical bifurcation model that the interface is applied to as a proof of concept in the third chapter.

Part 1 : Biological neuroscience

Section 1.1.1 : Brief description of a neuron

A neuron is a specialized cell of the brain. It is made out of three distinct parts : dendrites, an axon, and a cell body, or soma. The neuron is the communicating cell of the brain ; it receives a number of signals through its dendrites, at intersection with other neuron axons called synapses, in the form of local modification of the electrical potential. The signals from several dendrites propagate and combine along their common path to the cell body. The resulting aggregated signal depends on the nature of the synapses, e.g. additive or subtractive, or any of the many possibilities that the electrochemical structures making up the synapses allow. The neuron reacts to this aggregated signal in a specific way, depending on its excitability, i.e. its behaviour w.r.t. an excitatory current, and outputs the result along its axon to be transmitted further into the central nervous system. [1]

A schematic representation of a neuron corresponding to the above description is presented in Fig. 1.

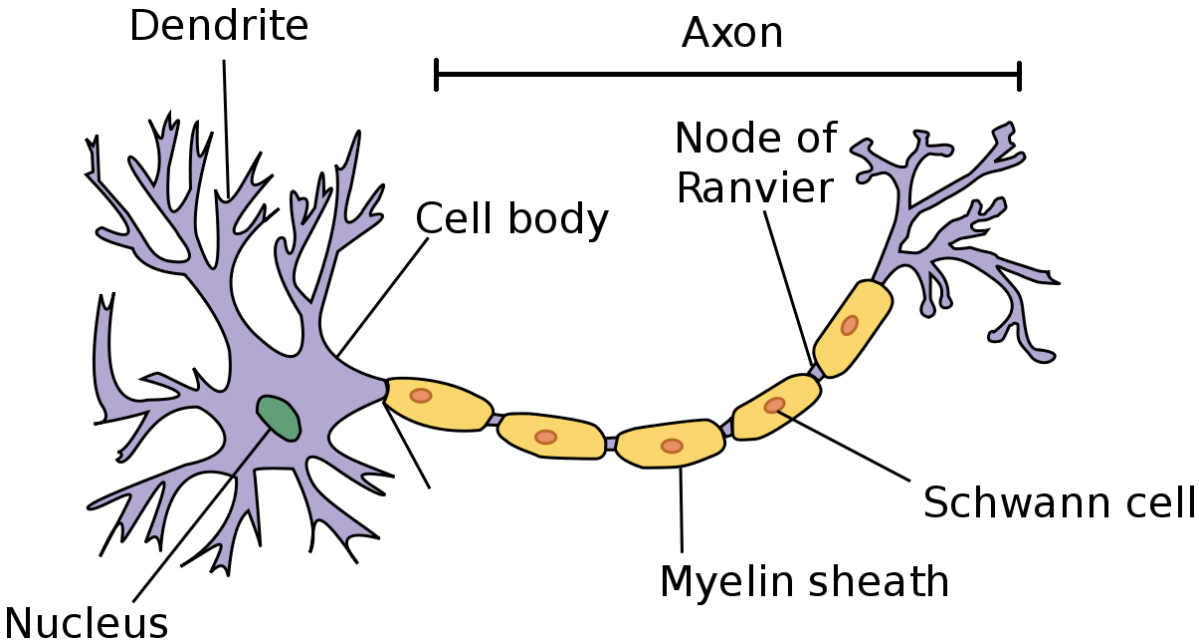

Fig 1: Schematic representation of a neuron. The dendrites, cell body and axon can be identified as the three main parts of the neuron. The cell nucleus is explicitly represented in the cell body. Myelin sheaths, Schwann cells and nodes of Ranvier are external structures built on the neuron through the action of surrounding cells that facilitate the transmission of signals along the neuron axon. (license-free illustration)

The signals transmitted by neurons come in many shapes. Examples of transmission patterns found in a rat's brain are depicted in Fig. 2. In this figure, patterns of both spiking and bursting activity can be observed, as first described in the introduction to this document. The nature of the electrical potential spikes that can be observed as making up these transmission patterns is covered in further details in the next sections of this chapter.

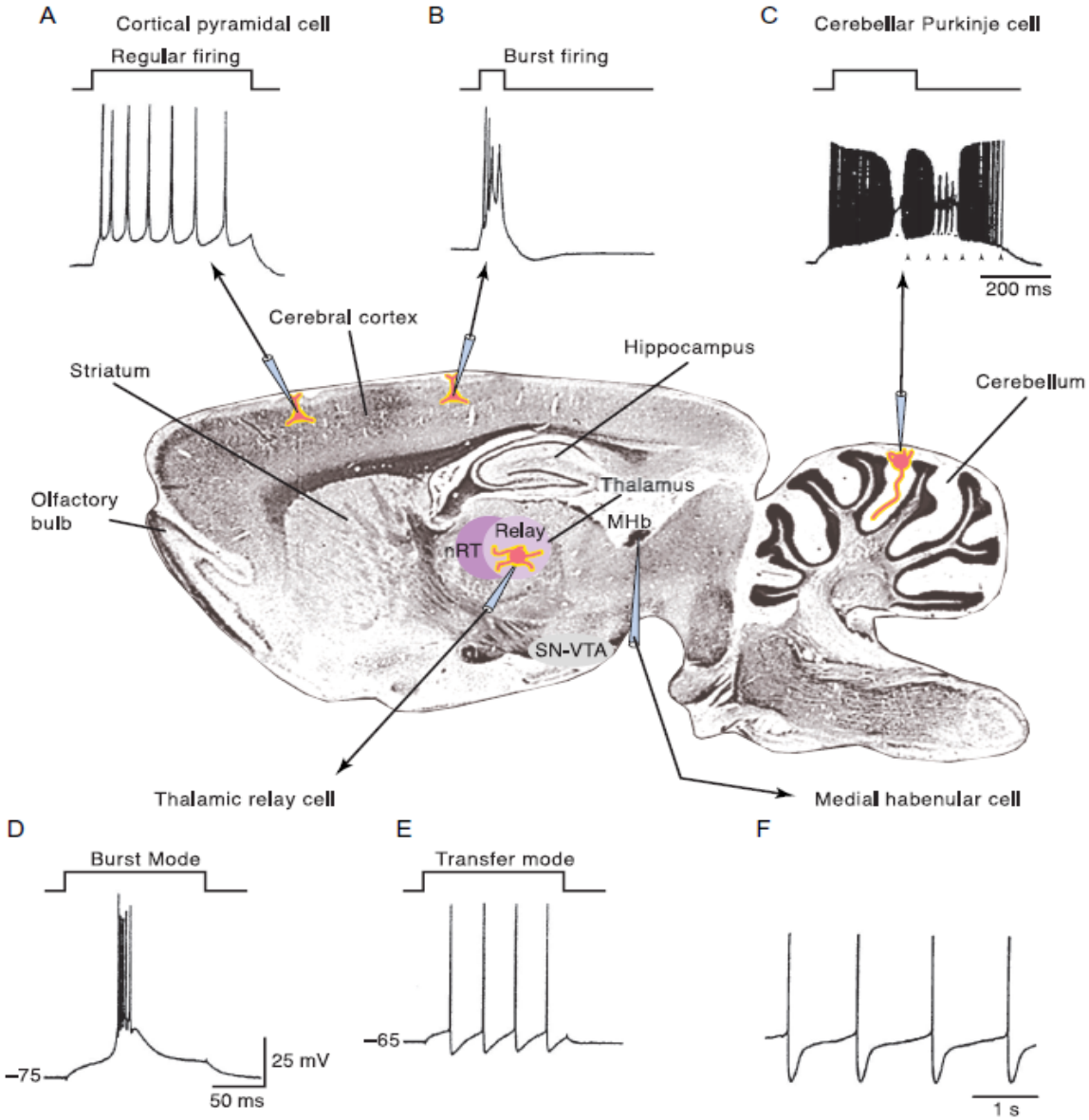

Fig 2: Examples of transmission patterns in a rat's brain. Spiking and bursting activity can be identified, e.g. in (A) and (B) respectively. The patterns display varying amplitudes and frequencies, depending on the zone of the brain that is studied. (extracted from [1] - pg 367)

Section 1.1.2 : Neuronal membrane and ion channels

The ability of a neuron to transmit signals is entirely dependent on the structure and current state of its membrane ; more precisely, it depends on the concentration of various ions in the intra- and extra-cellular media, and the degree to which these ions can travel through the membrane from one of these medium to the other. As with most cellular membranes, the neuronal membrane is made of a bilayer, i.e. two layers, of phospholipids arrayed in such a way that is it highly impermeable to polar molecules such as ions. [1]

The only way for ions to go through the membrane is to travel through specific trans-membrane proteins called ion channels. Ion channels are specialized structures that span the width of the membrane and allow the flow of ions at their specific location in the membrane. The ion channels involved in neuronal signal transmission are voltage-gated, meaning that they are equipped with gates that open or close according to the potential that is applied to them, i.e. across the membrane. In the context of neuron mathematical modelling, all elementary ion channels acting individually are aggregated in a global statistical ion permeability parameter. This global parameter expresses the total permeability of the membrane to the ion species as being proportional to the average number of channels that are currently open multiplied by their individual permeabilities, or rather to the average proportion of open channels multiplied by the total maximum permeability. This allows the global permeability parameter to evolve smoothly as a continuous function of the neuron membrane potential.

Channels are selectively permeable to one species of ion ; there are therefore a number of types of different channels corresponding to each species of ion that are required to go through the membrane during the normal function of the cell. There also are several types of gates, i.e. of gating particles that can transition between the two states that either block or allow the passage of ions in the elementary channel that they are associated with. Among those, activation gates and inactivation gates respectively open and close the channels during an increase of the neuron membrane potential.

A representation of the two most important channels in neurons, i.e. the potassium and sodium channels, is presented in Fig. 3.

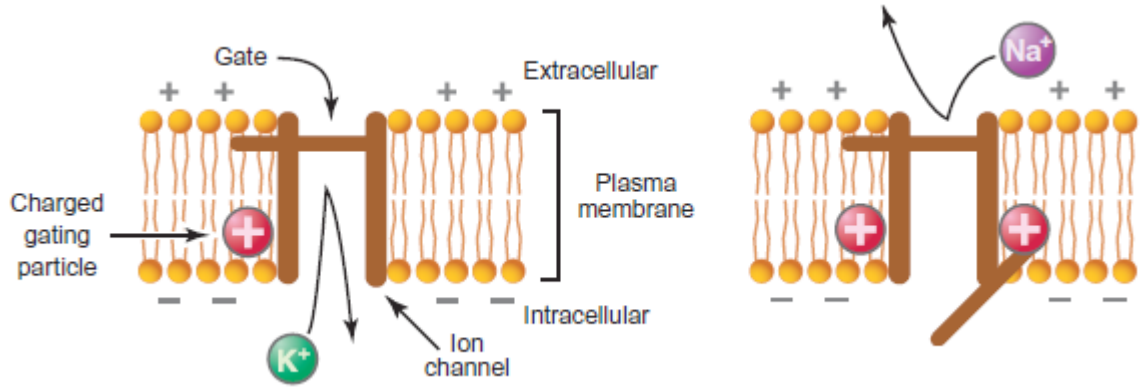

Fig 3 : Schematic representations of the potassium (left) and sodium (right) ion channels. The potassium channel can be seen to feature a single activation gate, while the sodium channel features both an activation and an inactivation gate. The natural tendency of potassium ions is to leave the cell, and of sodium ions to enter it, as is discussed in the next section (extracted and adapted from [1] - pg 415)

Section 1.1.3 : Ion concentration gradients and pumps

While ion channels regulate the ability of ions to flow through the membrane, they are not the source of these movements. The forces that can create movement in ions arise from two phenomena : diffusion down concentration gradients, and drift due to electrical fields, as ions are charged particles. Therefore, ion channels are only useful if there exists either a concentration gradient, or a built-in electrical field. As it happens, both are present across the membrane. ([1])

Ion pumps are the complementary mechanism to the ion channels : consuming energy in the form of ATP, they actively work to create specific concentration gradients in a number of ions that are used to generate action potentials. The key players are Na^+ (Sodium), K^+ (Potassium), Cl^- (Chlorine) and Ca^{2+}(Calcium).

In the context of a concentration gradient in a given species of ions, the Nernst equation can be applied in order to determine the equilibrium between diffusion and drift currents. As ions tend to travel down their concentration gradient through their associated channels, their movement create charge unbalances that oppose the diffusion current through the creation of an electrical field inducing a drift current.

The Nernst equation is presented in expression 1, where R is the ideal gas constant, T the temperature, z the charge of the ion, F the Faraday constant, and the letter "o" or "i" signals whether the ion concentration is measured out or in of the cell.

$$E_{equ} = \frac{RT}{zF} \ln \left[\sum \frac{[\text{ion concentration}]o}{[\text{ion concentration}]i} \right] \qquad \text{(expr. 1)}$$

When a number of ions are present in the intra- and extra-cellular media, the Goldman-Hodgkin-Katz equation can be used as an approximative generalization of the Nernst single-species equations to several species. In this equation, the concentration of the different species of ions are weighted by the respective permeability of these ions through the membrane. In other words, this equation expresses an equilibrium between the respective tendencies of the different ions to create drift-diffusion equilibria ; would a given ion display a much higher permeability than the others, then the resulting electrical field and resulting potential would be close to the equilibrium potential corresponding to a situation where only this ion is present.

An important fact is that contrary to the case corresponding to the Nernst equation, the potential defined by the Goldman-Hodgkin-Katz equation corresponds to a middle-ground between all the species involved ; therefore, none of the ions is in equilibrium, and non-zero net ion currents always exist. Countering these current (either drift or diffusion, depending on whether the global equilibrium potential is above or under the individual species' equilibrium potential) is one of the roles of the ion pumps.

The Goldman-Hodgkin-Katz equation is presented in expression 2, where R is the ideal gas constant, T the temperature, F the Faraday constant, p_k the membrane permeability associated to the ion "k", noted as "k+" or "k-" depending on its polarity, and the letter "o" or "i" signals whether the ion concentration is measured out or in of the cell.

$$E_{equ} = \frac{RT}{F} \ln \left[\sum \frac{p_k[k+]o + p_k[k-]i}{p_k[k+]i + p_k[k-]o} \right] \qquad \text{(expr. 2)}$$

This equation, however, fails to hold true in the presence of e.g. Ca^{2+} ions, as their charge is twice that of e.g. K^+ ions, and is not accounted for in expression 2.

Section 1.1.4 : The generation of action potentials

An action potential is an electrical pulse generated at a neuron membrane. It is one of the most important features that define a neuron, if not the most important ; it is what makes a neuron an "excitable" cell, rather than a passive structural entity. Action potentials are the basic building blocks of neuronal activity. Armed with the information covered in the previous sections, the events defining an action potential can be described and understood - the chosen example being the squid giant axon studied by Hodgkin and Huxley in 1939, as described in **[1]**.

An illustration of a recording of the membrane potential of the squid axon during the generation of an action potential is presented in Fig. 4.

Fig 4: Recording of the squid axon's membrane potential during the generation of an action potential. The successive phases of rest, fast depolarization, peak, hyperpolarization and slow relaxation to rest can be observed from left to right. (extracted from [1], pg 352]

At rest, the squid axon is characterised by constant concentration gradients of ions across its membrane, thanks to the operation of its ion pumps. The ion permeabilities values involved in the Goldman-Hodgkin-Katz equation are such that the K^+ permeability is considerably higher than the others. The global equilibrium is therefore fairly close to the potassium equilibrium potential : their values are of approximatively -60mV and -76mV, respectively. The membrane is said to be polarized. **([1])**

When a stimulus in the form of an excitatory current is applied to a neuron, its membrane potential increases. If it increases up to a given threshold potential whose exact value depends on the specific nature of the neuron (between -55mV and -45mV for the squid axon), voltage-gated Na^+-permeable channels activate, and the permeability associated with these ions rises sharply, bringing the membrane potential upwards in a positive feedback mechanism. The Na^+ ions become the main players in the Goldman-Hodgkin-Katz equation, and the global equilibrium membrane potential is brought closer to the sodium equilibrium potential, their values being of about 40mV and 55mV, respectively. The membrane is said to be depolarized.

The increase in the membrane potential then triggers two other mechanisms associated with the ion channels : the inactivation of the Na^+ channels, and the activation of the K^+ channels. The permeabilities of Na^+ and K^+ respectively fall and rise sharply, and the global equilibrium potential is brought back towards the potassium equilibrium potential in a negative feedback mechanism, and to a lower potential than at rest : down to about -65mV. The membrane is said to be hyperpolarized.

Finally, the hyperpolarized phase triggers the deactivation de-inactivation of the Na^+ channels, and the deactivation of K^+ channels. The potential returns to the resting value in a slow relaxation phase, and another potential can then be generated. The duration of the hyperpolarized phase defines a length of time during which the neuron first cannot be forced to generate another action potential, then can only be forced to do so with an increased level of stimulation. These are the absolute and relative refractory periods, respectively. These periods limit the maximum frequency at which action potentials can be generated in a repeated pattern ; they also keep the signals from reverberating back to their origin, from the axon back to the cell body and the dendrites.

For every phase of the action potential, there are ions that travel through the membrane and modify the concentration gradients, and thus the equilibrium potentials by the Nernst and Goldman-Hodgkin-Katz equations. However, the quantities of ions required to modify the membrane potential in order to generate a single action potential are negligible w.r.t. to the original concentrations. The orders of magnitude are the following : a single action potential involves a shift of about 6 μM (micro-mollar, i.e. micro mole per cubic meter) w.r.t. to an original concentration of about 18mM (milli-molar). Over large periods of times and great numbers of action potentials, it is the ion pumps that work against the shift in concentrations by bringing the ions back to their intended sides of the membrane.

There are cases when the pumps aren't strong enough to maintain the concentration gradients : certain forms of seizure cause fast, synchronized and repeated generation of action potentials. Alternative mechanisms originating in the action of glial cells are then required to maintain the desired unbalance. ([1])

Section 1.1.5 : Neuronal excitability types - experimental viewpoint

Hodgkin, one of the pioneer and most famous names in neuroscience, originally defined three classes, or types, of neuronal excitability. These definitions are based on observations made during experimental procedures that consisted in applying a current through a *Carcinus maenas* axon, i.e. the squid axon discussed in **[1]** and used for the description of an action potential in the last section. While invertebrate neurons differ somewhat from mammalian neurons, it has been proven that the generation of action potential in both groups is performed through similar processes. [2]

Hodgkin's proposed classification is based on whether the axon displays cyclic behaviour, i.e. responds to a constant excitatory current with trains of action potentials.

- **I** : cyclic behaviour over a wide range of frequencies ; the frequency evolves as a smooth function of the applied current
- **II** : cyclic behaviour at an approximately constant frequency that is insensitive to the applied current ; lower threshold post spike
- **III** : no cyclic behaviour, or only for very high values of the current

A representation of the current-frequency curves and temporal evolution of the membrane potential of neurons of type I and II is presented in Fig. 5.

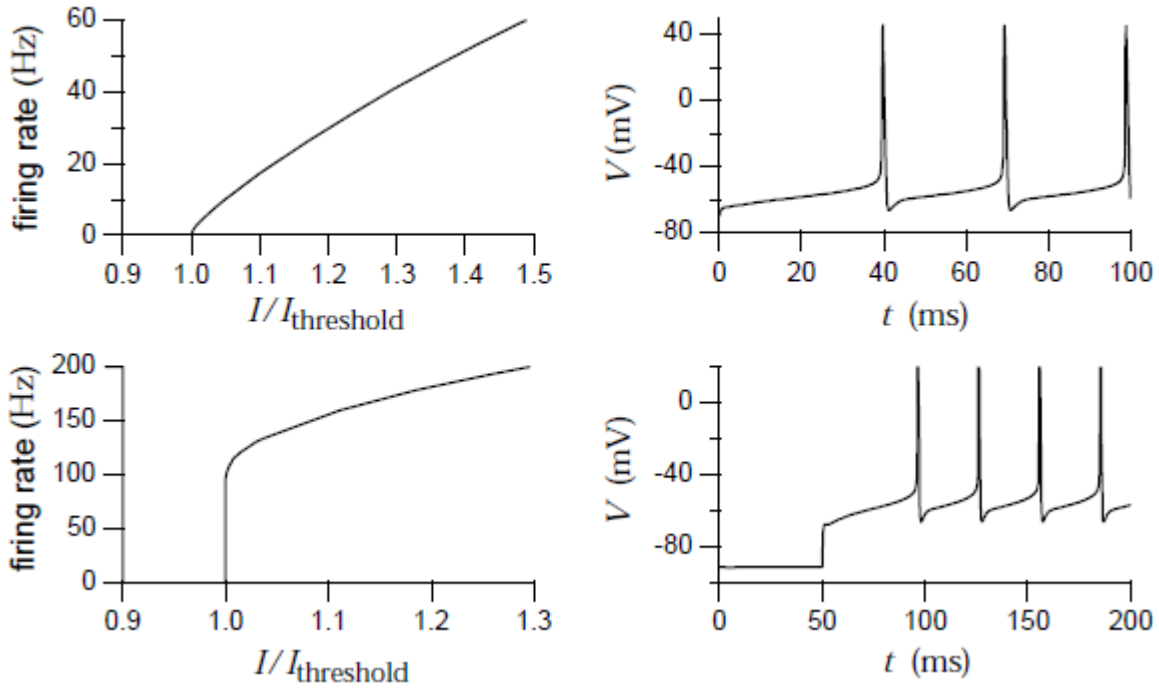

Fig 5: Current-frequency curves of type I and type II neurons (left) and temporal evolution of the neuron membrane potential (right) from the Connor-Stevens model, with type I corresponding to the upper traces and type II to the lower traces. Type I excitability can be seen to correspond to a nearly-linear current-frequency curve allowing low frequencies, while type II is considered to be nearly constant at a fixed high frequency (extracted from [3] - pg 202)

Following the experimental observations of Hodgkin on the squid axon, class I neurons are assessed to correspond to motor neurons and sensory receptors, as proven through experimental procedure in associated contemporary works. The frequency evolves smoothly in ranges of about 5-150 Hz. Class II neurons feature constant frequencies in the range 75-150 Hz. Class III neurons are assessed to correspond to any kind of neuron that has been left in sea water for a long period ; left for an even longer period of times, they become completely inexcitable.

The curves presented in fig. 4, however, correspond to a mathematical model of a neuron proposed by Connor-Stevens [4] ; the exact ranges of frequencies corresponding to the excitability types therefore differ from those observed by Hodgkin, although the qualitative observation pertaining to the shape of the frequency-curve does not.

In Hodgkin's own words, this classification is of an arbitrary nature, since a single axon may show different types of behaviour during the course of one experiment, depending on external factors such as storage conditions.

The second part of this chapter, however, introduces mathematical modelling and covers the notions of bifurcations in non-linear systems which give strong mathematical evidence to the accuracy of the classification in excitability types proposed by Hodgkin. Furthermore, the discussion on the capacity of a neuron to display several types of excitability during the course of its life is supported by the abilities of the chosen example model for this work to display different excitability types depending on the value of its internal parameters which can be identified to physiological parameters of the neuron.

Part 2 : Computational neuroscience

Section 1.2.1 : Neurons in the context of signal processing

From a signal processing viewpoint, neurons are active transmitters. As described in the first part of this chapter, their internal structure comports ion pumps and channels that allow for local modifications in the neurons membrane potential to be generated when an excitatory current is applied to them. The direct consequence of this is that signals are actively regenerated along the length of the neuron, instead of progressively fading out following the classical decreasing exponential describing the propagation of signals in any standard resistive conductor, such as typical balanced lines to which the Telegrapher's equation are applied. [1], [3]

The Telegrapher's equations and their solution under steady-state and small losses assumptions are recalled in expression 3, where V is the membrane potential, r_m the membrane resistance, r_i the internal resistance, V_o the value of the membrane potential at the position x=0. A representation of the corresponding equivalent circuit is illustrated in Fig. 6.

$$V = \frac{r_m}{r_i}\frac{d^2V}{dx^2} \quad \text{is solved by} \quad V = V_0 e^{\frac{-x}{\lambda}} \quad \text{where} \quad \lambda = \sqrt{\frac{r_m}{r_i}} \qquad \text{(expr. 3)}$$

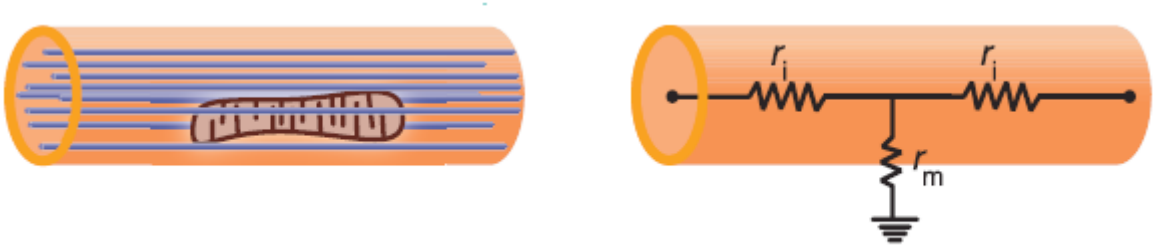

Fig 6: Representation of a section of a neuron (left) and its corresponding equivalent circuit (right) neglecting reactive components and considering the neuron is a purely resistive component (extracted and adapted from [1] - pg 514)

The signals travelling through the central nervous system consist in information from a great number of different sources ; among those, the skin nerves, the visual cortex, the olfactory system, ... all provide information of a different nature. However, they share neurons as the physical means by which this information is encoded. Therefore, the excitability types of the neurons define a great part of the basic structure of the physical layer of inter-neuronal communication, as identified in the OSI model. The exact use of these in order to build an encoding scheme and complete the definition of the physical layer vary, depending on the region of the brain that is being considered. [1]

Several schemes of information encoding have been experimentally observed in the animal brain. Rate code, or frequency encoding, has been found e.g. in the motor cortex neurons where the amount and direction of force required to perform a movement are encoded in the frequency of these neurons firing activity. Temporal code has been found in the hippocampus, where the phase of bursts activity is used to encode a measure of spatial location. Many other schemes exist, such as "coarse" coding, i.e. where a neuron excitability is linked to a single concept. [1]

Section 1.2.2 : Neuron mathematical modelling

Neuron mathematical modelling is made difficult by the fact that the cells complex behaviour can only be captured by non-linear systems. The exact number of dimensions of these systems is subject to discussion : on the lower end of the spectrum, there are models that are built through identification to normal forms of bifurcations, i.e. to the simplest system that displays the observed behaviour that is identified as a bifurcation. On the higher end of the spectrum, there are models that display up to a hundred different variables that aim to fit perfectly to experimental data.

Historically, the first models to be proposed were conductance-based models ; these aimed at explaining the mechanisms underlying neuronal excitability. The first of these is the classical four-dimensions Hodgkin-Huxley model, and is presented further in this second part of the chapter. [5]

The growth in the fields of mathematics in non-linear system analysis and of computer-enabled numerical analysis later spurred the research on the link between neuronal excitability and bifurcation theory. Reduction of the conductance-based models was performed and allowed the use of visual methods from the fields of non-linear systems analysis, i.e. mainly phase plane analysis through the work of FitzHugh, Rinzel and Ermentrout.

In particular, FitzHugh proposed in 1961 a model that was able to display neuronal dynamics through a model lacking explicit expressions of conductances or currents, but rather based on bifurcation theory : it is a generalized case of the well-known Van der Pol oscillator, to which it corresponds if two of its parameters are set to zero. As a matter of fact, the original name of the model, as extracted from the seminal paper, is the "Bonhoeffer - van der Pol oscillator". [6] The example model to which the methods developed for this work are applied is strongly inspired from the FitzHugh model, as presented further in this second part of the chapter.

A considerable risk with normal form identification is that the bifurcation that the original system is identified to may not be the correct one. As a practical example of the problems induced by this notion, the author of **[5]** contrived to fit normal-form models to experimental findings by progressively altering them with complicated discontinuous reset mechanisms. The validity of this approach is subject to discussion, and an argument can be made towards extensive numerical exploration of original models in order to detect the exact bifurcation underlying their behaviour. Following this approach systematically, heavy alterations to the models in order to fit experimental data should never be necessary, as errors could be traced back to an erroneous identification of the underlying bifurcation.

Section 1.2.3 : Conductance-based neuron modelling

The first part of this chapter presents the modern way of understanding the mechanisms driving the generation of action potentials in neurons. Conductance-based models are based on this understanding and model a neuron internals as a set of ion channels whose state of being partially open or closed allow for various quantities of ion currents to flow through and affect the neuron membrane electrical potential ; this state is identified to a non-linear conductance g_n (i.e. the permeability of ion channels) in an expression of Ohm's law for each of the ion channels, with the outside and inside of the neuron membrane forming two plates of a capacitor C_m consisting in the phospholipid membrane bilayer that can be charged or discharged through the different channels. [11]

A generic representation of this family of models is given in Fig. 7.

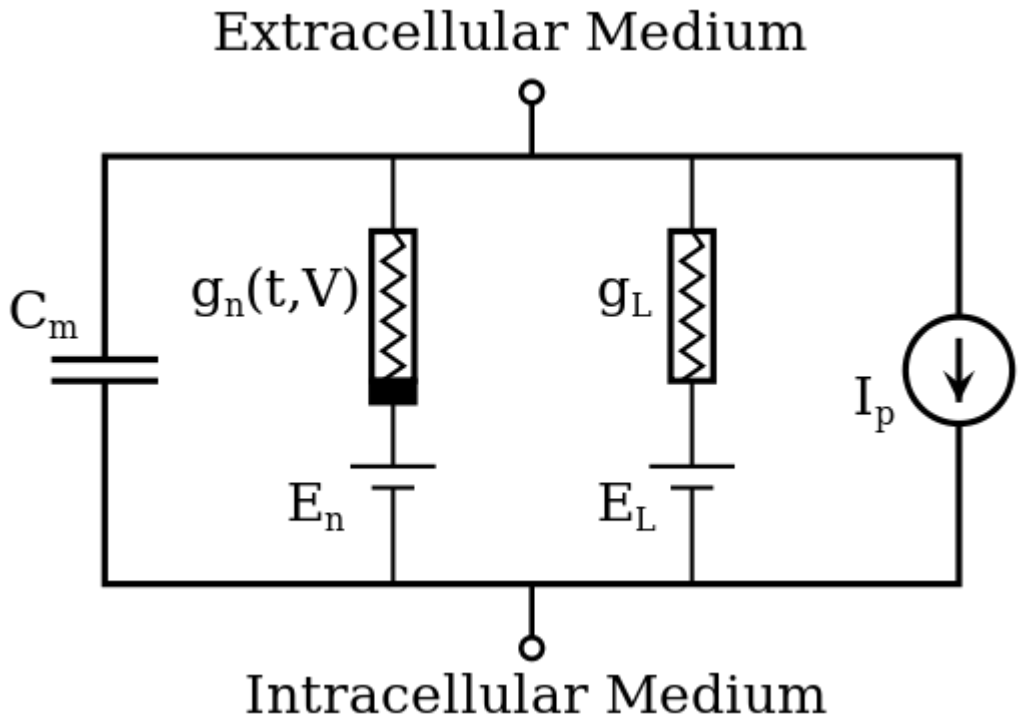

Fig 7: Schematic representation of conductance-based models. The channels are identified to non-linear conductances g_n, with additional leaking channels g_L. The ions are driven into the channels by their Nernst potential, modelled as electrical generators E_n, with additional leakage potential E_L. The ions pumps work constantly and are modelled as a current source Ip. The membrane forms a capacitor C_m between the extra- and intra-cellular media. (license-free illustration)

Each ion channel is driven by a force assimilated to an electrical generator E_n . This force corresponds the single-species equilibrium potential quantified by the Nernst equation ; the net electrical force driving the ions to which the channel is subjected is the difference between the membrane potential and the Nernst potential, both expressed with respect to the same reference, i.e. the surrounding bath on the outside of the membrane.

The ion pumps are modelled by a constant current source I_p whose flow is opposite to the ion channels' branches. A given amount of leakage current can also be modelled by a dedicated (constantly open) leakage channel g_L and a generator E_L .

The main complexity of these models reside in the modelling of the ion channel permeabilities as the non-linear conductances g_n . These must account for the mechanisms of the many elementary voltage-gated ion channels, as described in the previous sections.

The gates are identified to a number of blocking particles that are defined by a transition probability between open and closed states. Therefore, the Boltzmann equation from the domain of statistical mechanics can be used in order to define the distribution of particles between the open (respectively, "up" state) and closed (respectively, "down" state) states. The same approach is used in other fields of physics, such as the modelling of electron-hole pairs creation in semiconductor devices.

The ion channels are seen as the aggregation of a set of elementary channels that are each gated by a given number of particles. These particles block the ions when they are in a down state, and allow the passage of ions when they are in a up state. The total ion channel conductance results in the proportion of elementary channels that are currently letting ions through. The exact developments can be found in [1], or any equivalent material.

The mathematical expression of the classical Hodkgin-Huxley conductance-based model is presented in expression 4 :

$$\begin{cases} C\dot{v} = -\overline{g_K}\,n^4(V-E_K) - \overline{g_{Na}}\,m^3 h(V-E_{Na}) - \overline{g_l}(V-E_l) + I_{app} + I_{pump} & \textit{(Membrane potential)} \\ \dot{n} = \alpha_n(V)(1-n) - \beta_n(V)n & \textit{(Potassium activation)} \\ \dot{m} = \alpha_m(V)(1-m) - \beta_m(V)m & \textit{(Sodium activation)} \\ \dot{h} = \alpha_h(V)(1-h) - \beta_h(V)h & \textit{(Sodium inactivation)} \end{cases}$$

(expr. 4)

The expressions $\alpha_x(V)$ and $\beta_x(V)$ are physiological coefficients that are experimentally identified as functions of V ; they correspond to the rate of transition of the gating particles to the up and down state, respectively ; their value is determined through experimental fit.

The expressions $\overline{g_X}$ correspond to the maximum conductances of the ion channels and allow for the total conductance to be expressed as the product of this quantity and the probability of a single elementary channel being open, rather than compute the sum of every channel and their elementary conductance (which would probably be extremely difficult to identify). This probability is the variable characterising the state of the channel in the system, and is called the activation variable of the ion.

The probability of a channel being open depends on the number of particles that are assumed to be involved in this channel and their nature. For example, the Na channel is assumed to have 3 activating particles and 1 inactivating particle ; the total probability of the elementary channel to be open is the product of the events of the 3 activating particles being up, and the 1 inactivating particle being down.

The stimulation is modelled by the applied current I_{app} ; this parameter allows the neuron to switch between rest and its activity pattern. I_{pump} is considered to be constant, as developed in the previous section presenting the ion pumps.

Section 1.2.4 : Reduction of the classical model

The reduction to two dimensions of the Hodgkin-Huxley model was originally built using physiological arguments and experimental findings**.** Later work by FitzHugh, Ermentrout and contemporaries proved that these arguments could be identified to properties of non-linear systems through the theory of bifurcations, e.g. that the reduced forms based on physiological arguments can be linked to normal forms of bifurcations whose pattern was identified in the behaviour of the experimental specimens.

Firstly, the transition rates of the activating particles of the sodium current ($\alpha_m(V)$ and $\beta_m(V)$ affecting m) are found to be much greater than those of n and h , the transition rates of the activating particles of the potassium current and of the inactivating particles of the sodium current, respectively. In other words, the activation of the sodium current operates on a much faster time scale than the other two mechanisms**.**

Secondly, these latter two transition rates operate on roughly the same time scale w.r.t. each other. This holds true up to the point that it allows to experimentally prove that the expression $n+h$ is approximately constant, thus forming a relationship between those two variables that allows the simplification of one of them.

These arguments allow to form a new, approached mathematical expression of the reduced model which is presented on expr. 5 :

$$\begin{cases} C\dot{v} = -\overline{g_K}\left(\frac{w}{s}\right)^{wp}(V-E_K) - \overline{g_{Na}}\, m_\infty(V)^{mp}(1-w)(V-E_{Na}) \\ \qquad\qquad -\overline{g_l}(V-E_l) \qquad\qquad \textit{(Membrane potential)} \\ \qquad\qquad +I_{app}+I_{pump} \\ \dot{w} = \phi(T)\dfrac{w_\infty(V)-w}{\tau_w(V)} \qquad\qquad \textit{(Activation variable)} \end{cases}$$

(expr. 5)

The terms corresponding to the sodium and potassium currents can still be identified in the equation of the membrane potential, however their associated activation variables have all been simplified to a single w activation variable and coefficients s , wp , $m_\infty(V)$, mp , $w_\infty(V)$, $\tau_w(V)$; the functions of V are sigmoids.

Section 1.2.5 : The basics of non-linear system analysis

Non-linear systems are impossible to solve analytically such that explicit equations defining the evolution of the variables over time may be extracted from the system mathematical definition. Many examples of these kinds of problems exist in the scientific literature ; one of the first was the "three-body problem", i.e. the problem of computing the trajectory of the sun, the earth and the moon as objects that each attract each other through the laws of gravity. The famous mathematician Poincaré found a way to extract qualitative, rather than quantitative information from this problem by developing a revolutionary geometric approach - the aim was to determine not the exact trajectories of the bodies, but notions of long-term stability, i.e. whether the bodies would be able to stay on stable orbits rather than fly off far to infinity. [7]

As non-linear system analysis doesn't aim at finding exact trajectories, but rather to determine whether there are stable equilibria, i.e. positions that will attract any number of trajectories after a given period of time, one of the most important topic of non-linear system analysis is the study of how these equilibria evolve w.r.t. to modifications in the system parameters. When the nature of an equilibrium changes, i.e. whether it is stable or not is modified, or when equilibria appear or disappear, there is a bifurcation.

There are as many bifurcations as there are possible transitions between configurations, and the number of possible configurations explodes with the increase in the number of equilibria and variables in the system. The definition and labelling of these is a topic of its own, and still subject to controversy. Since the methods developed for this work are applied to a two-dimensional system with only a few equilibria, only those bifurcations that are met in the course of this work are covered in this document.

Section 1.2.6 : Neuronal excitability types - computational viewpoint

Neuronal excitability types as seen through the lens of computational neuroscience are mainly characterised by two things : the bifurcations to which they correspond, which are discussed further in this second part of the chapter, and the role they play in networks.

Neurons displaying type I excitability are able to transmit a large range of frequencies, and therefore are thought to encode the amplitude of the excitatory current, i.e. the amplitude of the input signal, into the frequency of the outgoing action potentials, i.e. the frequency of the output signal. [3]

Neurons displaying type II excitability are only able to transmit at an approximately constant frequency, and therefore are thought to be on/off encoders ; they either generate action potentials at their resonant frequency, if the input signal has a high enough amplitude, or they don't.

While only type I excitability allows for frequency encoding, both type I and type II can be used for temporal encoding, i.e. where the information is encoded in the time period separating two bursts of any frequency. In practice, recent discoveries point to the fact that the actual encoding scheme is part frequency and part temporal, as well as multiplexed combinations of these two, depending on the region of the central nervous system that is considered.

Section 1.2.7 : Bifurcations and their local properties

A bifurcation defines the threshold between a (locally or globally) stable configuration and an unstable one. If this bifurcation corresponds to the system transition from a configuration with a stable equilibrium to a configuration with none, then this also is the transition to either one of the types I or II of excitability - this can be simply proven with the Poincaré-Bendixon theorem presented in the appendices. This theorem states that if a trajectory is confined to a closed, bounded region and there are no fixed points in this region, then the trajectory must eventually approach a closed cycle. What interests us in particular is that the excitability type of the neuronal model can be assessed from the nature of this bifurcation.[7]

While a bifurcation is in part defined by the stabilisation or destabilisation of equilibria, or creation or destruction of equilibria that it concerns, it is also defined by the behaviour of the equilibria undergoing said bifurcation.

The stability of an equilibrium can be assessed through the extraction of the eigenvalues of the system Jacobian matrix evaluated at the equilibrium position. This corresponds to a linearisation of the system around the equilibrium point, which is a valid approach to assess the behaviour of the system only under strong hypothesises of small perturbations, i.e. in a range of perturbations fairly close to the equilibrium position. This type of computation is typical of stability analysis in adjacent fields such as rational mechanics. There is more information to be gained through this procedure than the simple notion of whether the equilibrium is stable or not, however ; this information comes in the form of the exact eigenvalues of the Jacobian matrix. [7]

In the context of this work, it is sufficient to say that the condition for an equilibrium to be stable is that its eigenvalues have strictly negative real parts. They can, however, and do display non-zero complex values in certain cases. Therefore, each bifurcation has in addition to its associated equilibrium configurations a pattern of eigenvalues of a specific nature ; these are described in the following section.

The following terms must be defined (in the context of a two-dimensional system) in order to allow the following discussions on bifurcations :

- A stable node is an equilibrium that is stable in all directions
- An unstable node is an equilibrium that is unstable in all directions
- A saddle point is an equilibrium that is stable along at least one direction, and unstable along at least one direction
- An attractor basin of attraction corresponds to the zone of the phase plane where trajectories are guaranteed to converge towards this attractor

An illustration of these concepts is presented in Fig. 8, in the section presenting saddle-node bifurcations. In this figure, the upper illustrations presents a node and a saddle, and the direction of the vector field around these equilibria. The node can be seen to attract trajectories from every directions, while the saddle can be seen to only attract trajectories along a single axis.

The saddle-node, as a compromise between a saddle and a node, attracts all trajectories from one half of the phase plane, with the axis separating the two halves being included in its basin of attraction ; the basins of attraction are pictured in dark grey while the rest of the phase plane is left as a white background.

Section 1.2.8 : Saddle-node & SNIC bifurcation

A saddle-node bifurcation, or SN bifurcation, corresponds to the following sequence of events : a saddle and a node collide and annihilate each other ; the equilibria simply disappear, although a ghost of varying importance persists at the point of their annihilation. The node has two pure real and different negative Jacobian matrix eigenvalues up until it disappears due to the bifurcation, and the saddle has one pure real positive value. The eigenvalues of the stable equilibrium increase in a monotonous fashion until they cross the imaginary axis. [7]

This sequence of events in illustrated in Fig. 8, from top to bottom.

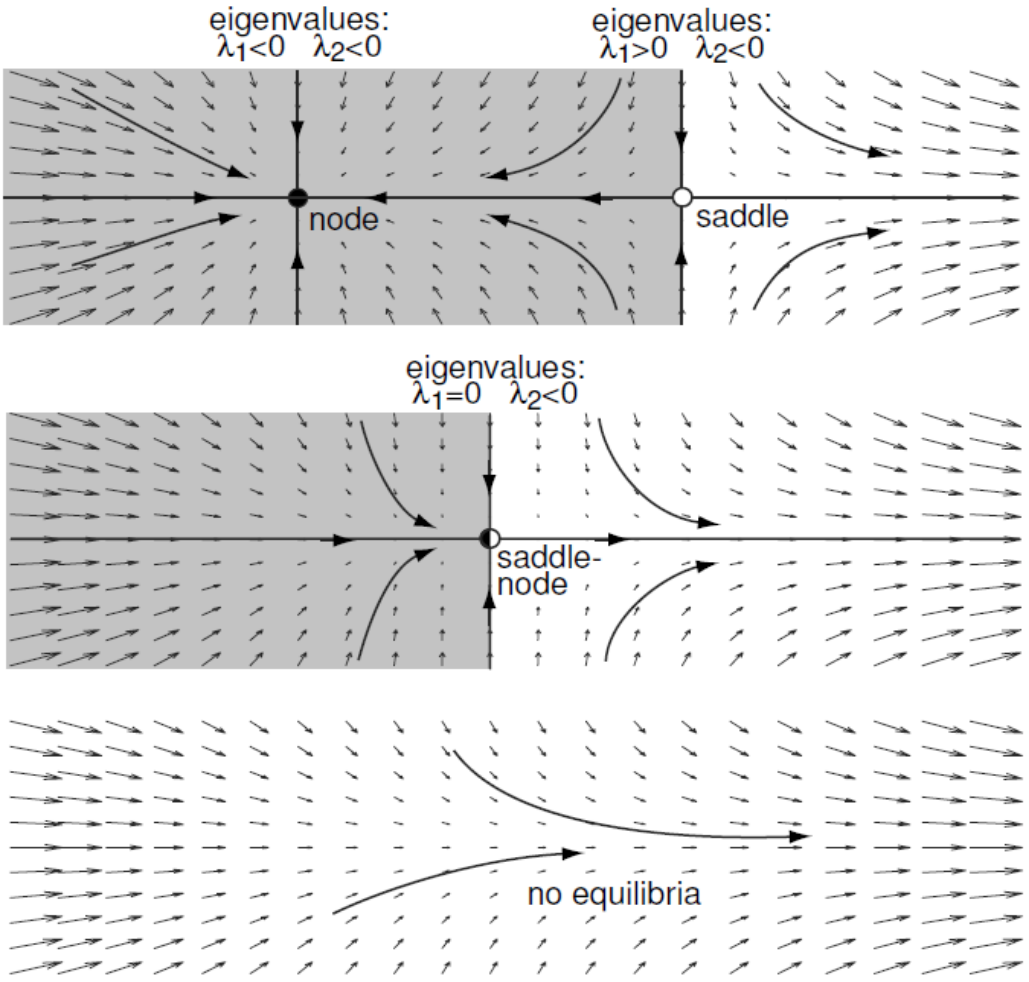

Fig 8: Depiction of a saddle-node bifurcation in a two-dimensional system, with explicit representation of the vector field. The system features a node and a saddle (top part), which collide and merge together(middle part) and annihilate each other (bottom part). Of particular interest is the fact that the vector field does not show any circular trajectories in the immediate vicinity of the saddle-node bifurcation,to the contrary to HOPF bifurcations (extracted from [5] - page 132)

Due to the evolution of the eigenvalues of the node, the system features strictly non-oscillating convergent behaviour around the node before the bifurcation ; complex eigenvalues induce circular trajectories, and the saddle-node bifurcation never features non-zero complex parts. This can be observed in Fig. 8, as the vector field around the saddle-node is strictly directed towards or away from it, but never drawing circular fields such as those inducing oscillations in the system.

If the system features only a single stable equilibrium before the bifurcation, then it causes the system to follow an invariant circle of large amplitude and approximately linear frequency with respect to the excitatory current, i.e. type I excitability. In this case, the bifurcation is labelled as a saddle-node on invariant circle, or SNIC bifurcation. An example SNIC bifurcation is illustrated in Fig. 9.

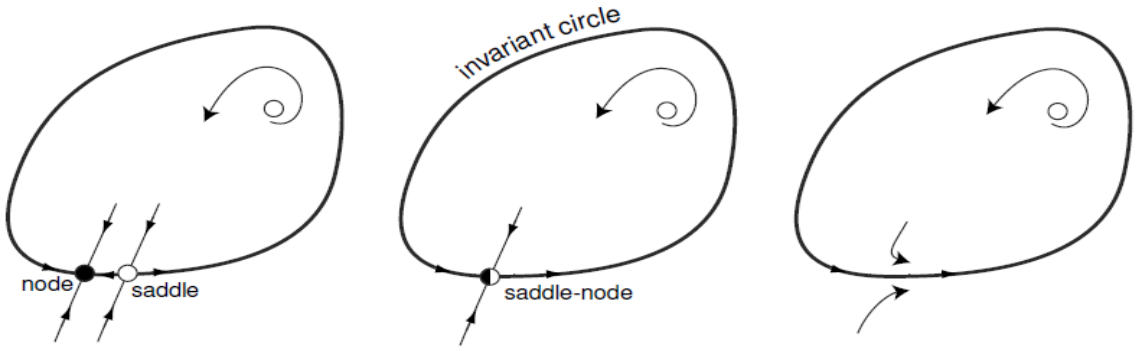

Fig 9: Depiction of a saddle-node on invariant circle, or SNIC bifurcation in a two-dimensional system. A saddle-node bifurcation takes place, and as there are no additional equilibria in the system while the space is bounded, a stable limit cycle appears where the trajectories previously transited from the unstable saddle to the stable node. (extracted from [5] - pg 223)

The saddle-node bifurcation normal form can be expressed as a one-dimensional system. It can, however, easily be extended to two-dimensional system by setting the derivative of the second variable to be a constant of any value.

Section 1.2.9 : Saddle-homoclinic bifurcation

If the system features the creation of a stable cycle before the bifurcation, then the trajectory will converge towards this existing cycle after the saddle-node bifurcation has the node disappear, and display the same type of current-frequency relationship. The main difference in the two is that the cycle is not located at the position of the SN bifurcation, therefore displaying a slightly different temporal membrane potential curve. In this case, the bifurcation is labelled as a saddle-node - saddle-homoclinic, or SN-SH bifurcation.

As the saddle-homoclinic is not directly observed in the context of this work, it is not detailed further ; it is sufficient to describe it as a bifurcation that sees the creation of a cycle at the position of a saddle. This is the kind of bifurcations that causes the system to feature a cycle before a SN bifurcation, and therefore to prevent SNIC bifurcations.

Section 1.2.10 : Andronov-Hopf bifurcation

An Andronov-Hopf bifurcation, or HOPF bifurcation, corresponds to the following sequence of events : a node is in a configuration that sees it go from a stable configuration to an unstable one, i.e. it goes through a specific position of a nullcline that separates its attractive and unstable branches in the phase plane representation. [7]

The node has two pure real negative Jacobian matrix eigenvalues up until shortly before the bifurcation, when the two real negative become equal and see a complex part appear, so that the two eigenvalues become complex conjugates. The values stay as complex conjugates, and their real part increases until it becomes zero at the bifurcation, and positive after that. This means that an unstable, oscillating equilibrium persists after the bifurcation and continues to move around the phase plane with the increase in the excitatory current. Due to the evolution of the eigenvalues of the node, the system features oscillating convergent behaviour around the node. This can be observed in Fig. 10, as the vector field around the node forms circular patterns :

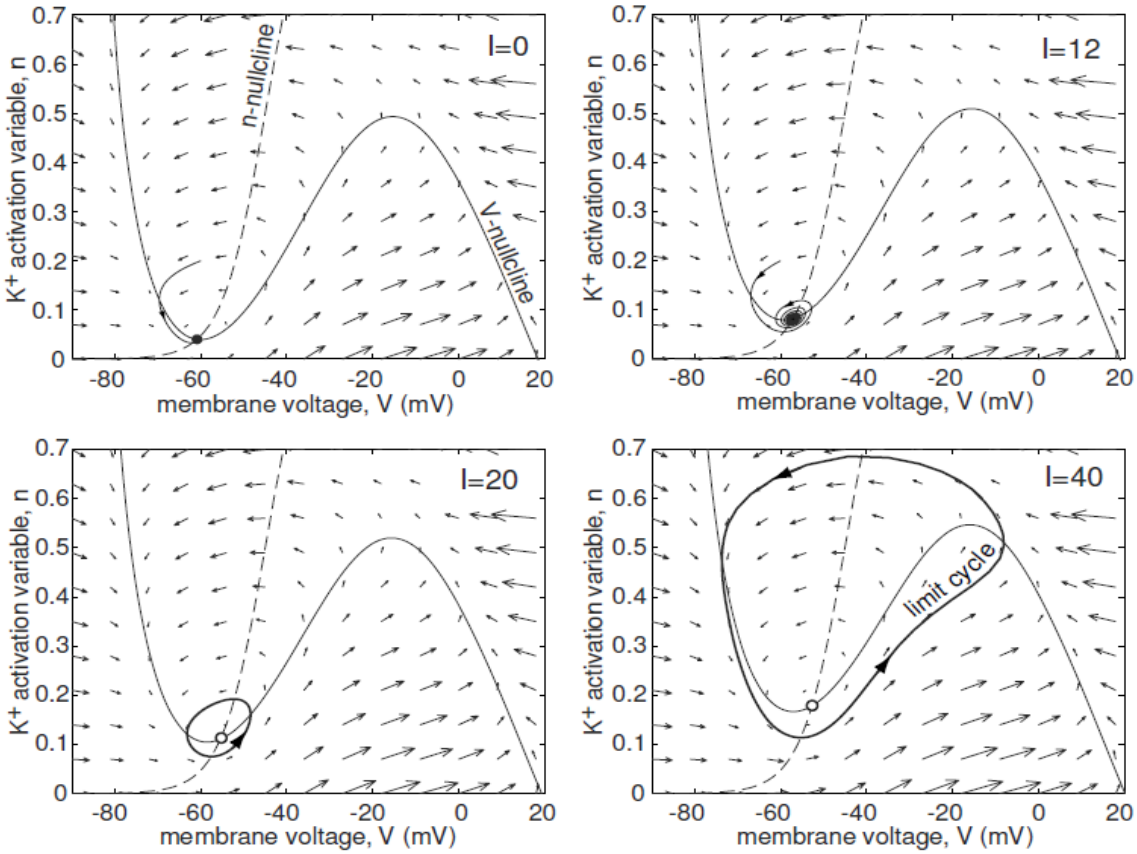

Fig 10: Depiction of a Andronov-Hopf, or HOPF bifurcation in a two-dimensional system, with explicit representation of the vector field. The system features a node (upper left) which moves on the nullcline towards the right, seeing the apparition of circular trajectories around it (top right), then getting destabilised (bottom-left) as the newly appeared cycle grows in amplitude (bottom-left) but not frequency.(not shown) (extracted from [5] - page 134)

Section 1.2.11 : Visual analysis of the reduced HH model

While a complete course on the subject of non-linear systems analysis is not the subject of this work, a quick example analysis is proposed as a way to introduce the reader to the methods commonly used in the field. In particular, that of visual inspection, i.e. deduction of the behaviour of a system based solely upon the observation of the configuration of its nullclines drawn in the space of its variables ; that is, for a two-dimensional model, on the phase plane defined by its two variables. [1]

The nullclines of the system, i.e. the n-V curves corresponding to when the derivatives are equal to zero, are drawn on the phase plane and a trajectory is computed for the visualization purposes w.r.t. the system behaviour, as illustrated in Fig. 11.

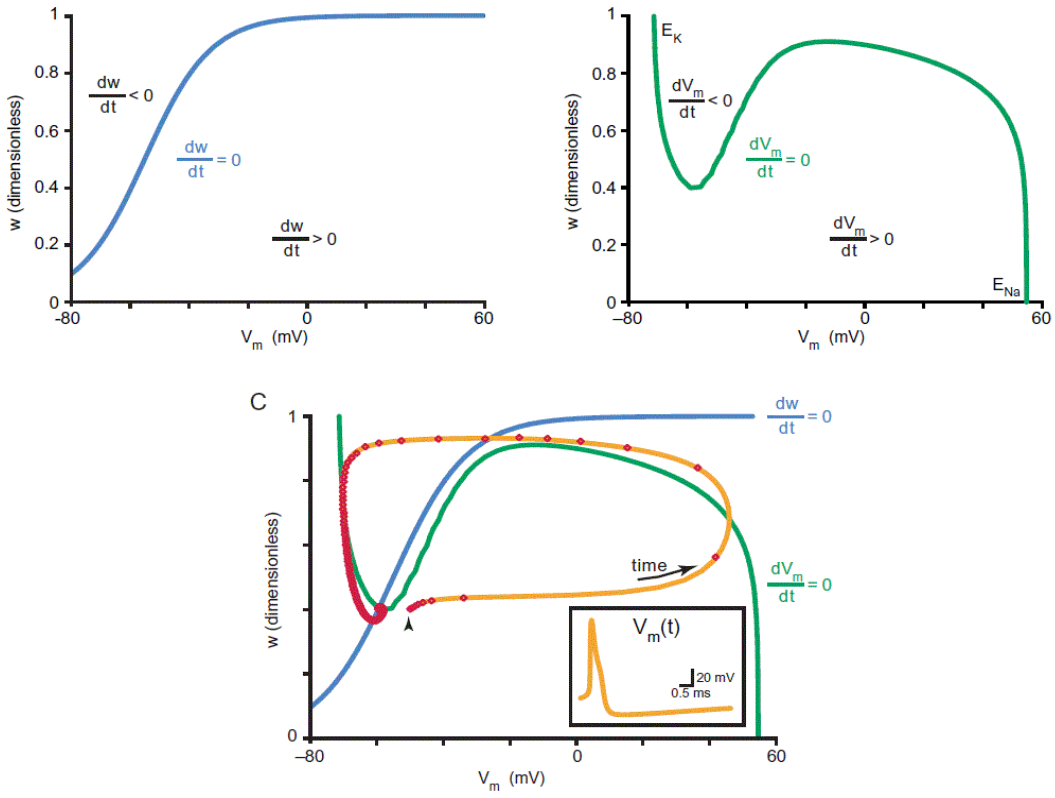

Fig 11: Depiction of the nullcline corresponding to the membrane potential (top-left) and the activation variable (top-right), then superposition of the two with an added trajectory in the system obtained through simulation. (bottom) The trajectory starts at a value of the membrane potential above the excitation threshold, and an action potential is generated. The trajectory then converges to the equilibrium situated at the intersection of the nullclines. A temporal recording of the membrane potential is added for visualization purposes. (extracted from [1] - pg 428)

Looking at the trajectory on the lower part of fig. 10, the first observation that can be made is that the system settles after a given time at the intersection of the nullclines. This is due to the fact that this corresponds to a rest situation, i.e. a situation where all derivatives defining the system are equal to zero.

Increasing the excitatory current applied to the neuron alters the shape and position of the nullclines, and therefore of the trajectories, as illustrated in Fig. 12.

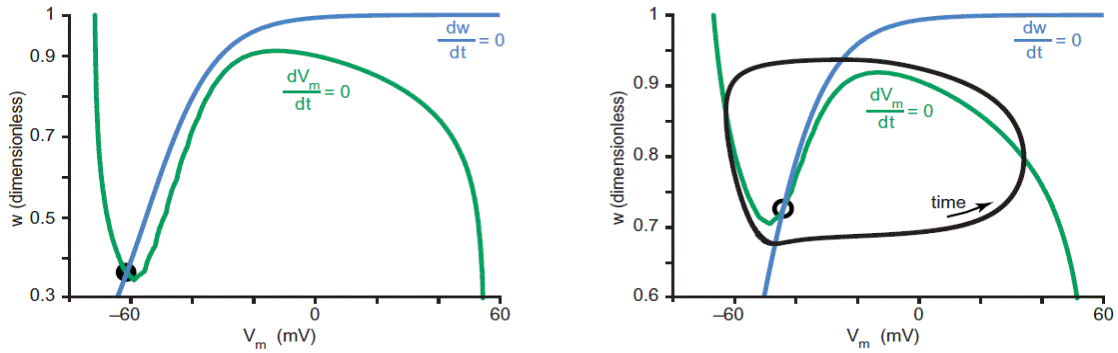

Fig 12: Recording of the evolution of the nullclines of the reduced Hogdkin-Huxley model with increasing values of the excitatory current. The current is first set to a value allowing the neuron to rest (left) then is increased until the neuron displays spiking activity (right). The equilibrium on the left part is denoted by a full dot as it is stable, while it is denoted by an empty dot as it becomes unstable during spiking activity. (extracted from [1]- pg 430)

Care has to been taken in comparing Figs. 11 and 12, as the vertical scale has been modified in order to zoom on the equilibrium position. The left panel of fig. 11 corresponds to the same situation as the one depicted in Fig. 10. The right panel of fig. 12, however, displays a configuration that has changed w.r.t. the left panel in that the nullclines shape morphed in such a way that while they still intersect, the corresponding equilibrium does not attract trajectories in the system anymore.

The difference between the two situations is that either one, or both of the eigenvalues of the system Jacobian matrix evaluated at the position of the equilibrium crossed the imaginary axis ; their real part grew from a negative to a positive value.

The common practice, however, is to contend with a visual inspection. In this case, the usual explanation extracted from the observation of these phase plane diagrams is that the V-nullcline features a stable branch to the left of its local minimum, and a stable branch to its right. While as long as these branches are only very loosely defined, this is technically true, the habit of automatically considering them to be separated at the local minimum is an error - this habit may have its origins in the study of one-dimensional systems, where vector fields on the phase plane are strictly horizontal. [17]

The bifurcation that the system undergoes between its quiescent and oscillating configurations is a HOPF bifurcation, since a single node looses its stability as the excitatory current is increased. According to the Poincaré-Bendixson theorem, a stable cycle must then be present and be followed by all trajectories in the system.

Extensive numerical study of the model shows that the eigenvalues of the equilibrium follow the HOPF bifurcation pattern, i.e. that they are complex conjugates with a growing real part as the bifurcation is approached, and that they cross the imaginary axis right at the bifurcation.

Section 1.2.12 : Transcritical (TC) canonical model

The model whose study is the subject of this work is extracted from recent research by Drion et al., as covered in [8] and [9]. It is a two-dimensional reduction of the classical Hodgkin-Huxley model previously augmented with an additional calcium channel in order to display physiological behaviour that was experimentally identified but not yet modelled in such a simple way, i.e. the ability to switch between spiking and bursting behaviour. Its two variables are the membrane potential and an aggregated activation variable of all slow ion channels, respectively.

The first step in building the model is to extend the classical Hodgkin-Huxley presented in the first section through the addition of calcium current. The mathematical expression of the model is presented on expression 6 :

$$\begin{cases} C\dot{v} &= -\overline{g_K}\,n^4(V-V_K)-\overline{g_{Na}}\,m^3h(V-V_{Na})-\overline{g_l}(V-V_l) \\ & \quad +I_{app}+I_{pump} \\ & \quad -\overline{g_{Ca}}\,d(V-V_{Ca}) \qquad \textit{(Membrane potential)} \\ \dot{n} &= \alpha_n(V)(1-n)-\beta_n(V)n \qquad \textit{(Potassium activation)} \\ \dot{m} &= \alpha_m(V)(1-m)-\beta_m(V)m \qquad \textit{(Sodium activation)} \\ \dot{h} &= \alpha_h(V)(1-h)-\beta_h(V)h \qquad \textit{(Sodium inactivation)} \\ \dot{d} &= \alpha_d(V)(1-d)-\beta_d(V)d \qquad \textit{(Calcium activation)} \end{cases}$$

(expr. 6)

Compared to the expression of the classical model, the only difference is in the additional Ca^{2+} current and its associated particle transition probability. The activation variables are expressed in terms of their transition rates rather than the form using the steady-state value and the time constant preferred for the reduced Hodgkin-Huxley model - the two forms are equivalent.

While the algebraic expression cannot express this, it is very important to take into account that the transition rate of the calcium activation variable is of the same order as that of the potassium activation variable and the sodium inactivation variable, and therefore slower than the sodium activation variable. This current has slow kinetics compared to its peers, and this is what allows the system to display two patterns of action potential generation, i.e. spiking and bursting. The calcium current varies in a time scale of its own and modifies the pattern formed by the action potentials, but not the action potentials themselves.

The second step is to reduce the model back to two dimensions through physiological arguments much like those applied to the classical Hodkgin-Huxley model. With the approximation of the Ca^{2+} variable as a constant, due to its very slow kinetics, the model loses its spontaneous ability to display bursting activity. However, bursting can still be achieved by altering the value of the calcium conductance considered as a parameter of the system.

The mathematical expression of the reduced model is presented in expression 7.

$$\begin{cases} C\dot{v} = -\overline{g_K} n^4 (V - E_K) - \overline{g_{Na}} m_\infty (V)^3 (0.89 - 1.1\mathrm{n})(V - E_{Na}) \\ \qquad\qquad - \overline{g_l}(V - E_l) \\ \qquad\qquad + I_{app} + I_{pump} \qquad\qquad \textit{(Membrane potential)} \\ \qquad\qquad - \overline{g_{Ca}} n^3 (V - E_{Ca}) \\ \dot{n} = \alpha_n(V)(1-n) - \beta_n(V) n \qquad\qquad \textit{(Activation variable)} \end{cases}$$

(expr. 7)

The final expression that is implemented corresponds to a normal-form identification of expr. 7 to a transcritical bifurcation, following a novel interpretation generalizing the FitzHugh family of models, as motivated in [8]. The mathematical expression of the final model is presented on expression 8 :

$$\begin{cases} \dot{v} = V - \dfrac{V^3}{3} - n^2 + I_{app} \qquad \textit{(Membrane potential)} \\ \dot{n} = \epsilon\left(n_\infty (V - V_0) + n_0 - n\right) \qquad \textit{(Activation variable)} \end{cases}$$

(expr. 8)

In this final expression, most coefficients have been simplified. Those that are left correspond to the following quantities :

- I_{app} : The applied excitatory current - this parameter corresponds to the excitation that the neuron is subjected to, i.e. the total aggregated signal that it receives
- ε : The time-scale coupling factor - this parameter controls the time-scale separation between the mechanism inducing the spike in the action potential, and the mechanism inducing the relaxation of the system after a spike has been generated
- V_o : The half-activation membrane potential - this parameter controls the horizontal position of the N-nullcline
- n_o : The ion channel balance factor - this parameter controls the vertical position of the N-nullcline

In particular, the parameters V_o and n_o have important physiological interpretations. The parameter V_o corresponds to the half-activation potential of the recovery variable. This can be identified to an image of the excitability threshold of the neuron that is modelled. The parameter n_o is an image of the calcium channel density. It defines the balance between potassium and calcium channels near the threshold potential. If it is very positive, the potassium channels are dominant. If it is very negative, the calcium channels are dominant. This, in turn, determines whether the model displays mainly restorative or mainly regenerative behaviour, respectively, which corresponds to a balance between the negative feedback mechanism of the potassium channels and the positive feedback mechanism of the calcium channels.

The ε parameter corresponds to the time-scale coupling factor, and therefore rules over the relative speeds of the mechanisms present in the system. This factor underlies all the dynamical analysis performed during this work and covered in the second part of the last chapter of this document.

Section 1.2.13 : Visual analysis of the TC model

A visual inspection of the model is covered in [8]**.** The results are summarised in Fig. 13 as a bifurcation map on the plane of the two parameters (V_o,n_o) :

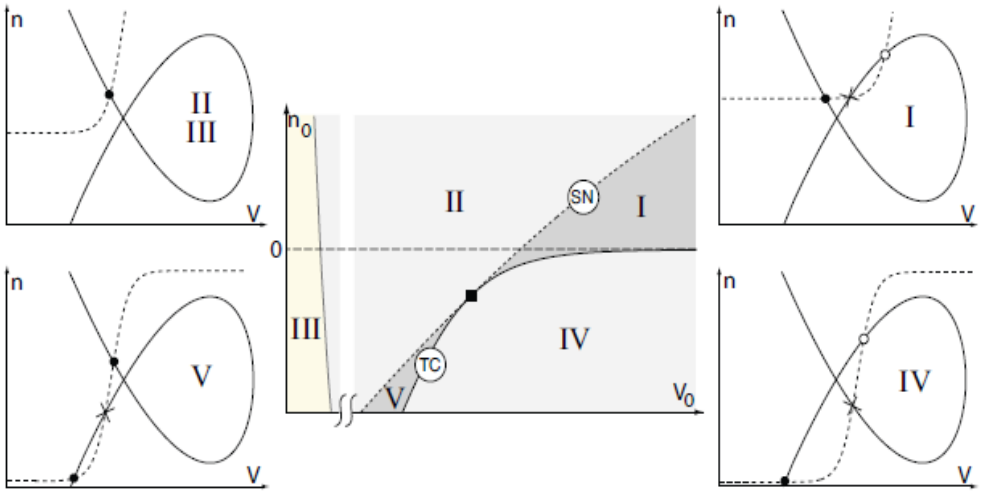

Fig 13: Illustrative summary of the visual analysis of the example model. Curves are nullclines, full dots are stable equilibria, crosses saddles, empty dots unstable equilibria. The space of the model's two main parameters (v_0,n_0) is seen to be separated in a number of regions corresponding to nullclines intersection patterns, subsequently conditioning the excitability types of the neuron. Types I, II and III correspond to the Hodgkin-Huxley excitability types, while types IV and V are newly proposed excitability types.(extracted from [8] - pg 10)

The validity of this visual inspection is restricted to the singular limits, i.e. considering the time-scale separation of the system different mechanisms approaches infinity. In other words, one variable is considered to evolve infinitely fast w.r.t. to the other ; in this case, the activation variable. These approximations are very much alike to the assumptions made in order to build the reduced Hodgkin-Huxley model.

The physiological interpretation proposed in [8] is that neuronal excitability is ruled by a transcritical bifurcation. The curve defining the position on the space of the system's parameters (V_o,n_o) where the transcritical bifurcation takes place can be computed through an analytical identification to a balance of the ionic currents affecting the system. The currents are assessed to be acting either restoratively or regeneratively, and the zones separated by the transcritical curve therefore correspond to behaviour that is either mainly restorative, or mainly regenerative. Furthermore, the mainly restorative region, i.e. the region II in Fig. 13, is identified with the behaviour of the classical Hodgkin-Huxley model which undergoes a HOPF bifurcation and displays type II excitability. The region III corresponds to no bifurcation in the physiological current range. The mainly regenerative region, i.e. the region IV in Fig. 13, is identified to a novel type of excitability that is the subject of the referenced paper, which is based on neither a HOPF nor SNIC bifurcation, but rather a SN-SH bifurcation. The region V corresponds to yet another novel type of excitability based on a modified version of a HOPF bifurcation.

Finally, the region I corresponds to type I excitability ruled by a SNIC bifurcation. It is not, however, the only region with SNIC bifurcations, as the transcritical bifurcation is a generalized case of the SNIC bifurcation. Therefore, there are in effect two type I regions.

Chapter 2 : Methodology

This second chapter concerns the methodology of the work performed during this thesis. It covers the software implementation of the bifurcation identification interface whose development is the subject of this thesis by presenting its underlying logic flow.

In the first part of this chapter, the Julia language is briefly introduced, along with the packages of its official extended library involved in the implementation of the interface. In-depth coverage of the language's syntactic specifics and inner workings is avoided, except for the defining features that set the Julia language as a strong competitor to be chosen as a a programmer's preferred language to developed scientific computation interface, along with e.g. the Matlab environment.

In the second part of this chapter, the software implementation of the bifurcation search and identification are covered at a high level of abstraction w.r.t. the specifics of the Julia language in order to skip any explicit integration of source code. The problem to be solved is formalised in a mathematical expression that root-finding algorithms such as the Newton-Raphson method can work on. Three different implementations of a stability detection function critical to the bifurcation identification method are presented. These implementations are based on three different packages of the official extended library of the Julia language, and perform with varying degrees of computational efficiency and accuracy.

In the third part of this chapter, the three separate implementations of the bifurcation search methods are evaluated and their efficiency and accuracy are assessed w.r.t. to the problem requirements.

The source code supporting the interface is freely available at the following address : http://www.montefiore.ulg.ac.be/~guilldrion/TFE/Thibault Gillis - Source Code.zip

Part 1 : The Julia language

Section 2.1.1 : Presentation of the language

The Julia language first appeared as the outcome of a 2009 MIT project, and subsequently went open-source in 2012. It is maintained by its original developers Stefan Karpinski, Jeff Bezanson, and Viral Shah, as well as by a number of external contributors, many of which are part of the scientific community. [10],[11]

It was originally conceived as an efficient alternative to Matlab, Python, or R, in the context of their use in scientific computing. The Julia basic library integrates a great number of open source C and Fortran libraries for linear algebra, random number generation or string processing. The Julia language has since its original launch enjoyed support by a growing number of scientists of various fields, such that many paper-motivated packages are now available as a part of the extended official library of the language. The packages used for this work are all extracted from this extended library, and therefore officially endorsed by the language authors through rigorous testing and conformity check procedures.

The Julia language is free, open-source, and compiled. Thanks to a compiler equipped with state-of-the-art mechanisms, however, the Julia language can be used in an environment very much like that of an interpreted language such as Matlab or Python. In particular, a process of "JIT" (i.e. "Just In Time") compilation performs efficient on-the-fly incremental compilation. Additionally, dedicated advanced compilation commands are available for power user to optimize the performance of their code and reach C-like computation speeds.

The choice of Julia as the programming language for this thesis arose from the fact that this seemed to consist in an interesting new programming challenge, as well as a way to allow the development of new implementations of a number of methods that are often extracted from a number of different sources in a coherent, self-sufficient interface. For example, the XPP environment is often used as a means to perform bifurcation analysis, along with MATLAB as a means to implement the equations of a conductance-based model and to simulate them. This consists in a fairly cumbersome way of performing computations that could be implemented in a single language. Therefore, special care has been taken so that the software supporting this thesis were entirely written and supported by the Julia language.

Section 2.1.2 : Julia packages - ODE

This package is one of the several available Julia implementation of the main ODE solver algorithms, i.e. the Runge-Kutta methods used to solve systems of ordinary differential equations through explicit time simulation. A complete differential equation solving ecosystem has since been developed by the author, in which the ODE package consists in one of the several implementations than can be used in the ecosystem. **([12], [13])**

The ODE package requires a number of libraries involved in the computations of the Runge-Kutta methods, i.e. differentiation and integration. In particular, the ForwardDiff package implements forward differentiation, and allows the numerical evaluation of the system Jacobian matrix at any point of the phase plane and hence of any equilibrium eigenvalues.

The package features implementation of non-stiff and stiff ODE solvers, i.e. typical ode23, ode45, ode78, ode23s solvers as can be encountered in other scientific computing environements such as Matlab.

Section 2.1.3 : Julia packages - NLsolve

This package contains implementations of the trust region and Newton-Raphson algorithms used to find zeroes of a multivariate function. These methods feature drawbacks common to algorithms such as that of the gradient descent, i.e. that they can only converge towards the nearest equilibrium, and that they are not guaranteed to converge at all depending on the exact properties of the system. **([14], [15])**

When the algorithms do converge however, they generally do so with excellent rates of convergence compared to more robust methods such as bisection, or indirect methods of finding equilibria such as simulation, i.e. the use of numerical ODE solvers using forward Euler integration, or Runge-Kutta methods. **([16])**

Both the trust-region and Newton-Raphson algorithms are methods that approximate the objective function (i.e. the system to solve) using a quadratic approximation. In the trust-region method, if the approximation is found to be accurate, the local region is expanded, else it is contracted. The size of the initial trust region can be explicitly specified. In the Newton-Raphson method, a line search strategy is used in order to converge towards a local minimum. The exact strategy can be explicitly specified. **[App]**

The solvers' API allows the specification of an explicit expression of the Jacobian matrix, thus avoiding an expensive and possibly inaccurate numerical approximation of the Jacobian matrix through finite differentiation. In the context of this work, this option is ignored in order to keep and assess the performances of the interface independently of a specific example system, for which an expression of the Jacobian matrix could not be easily determined.

Section 2.1.4 : Julia packages - Roots

This package contains implementations of classical univariate functions roots-finding algorithm, such as the Newton-Raphson, Halley's or secant methods. It is used on the equation defining the derivative of the membrane potential where the activation variable has been replaced with the corresponding function of the membrane potential, when the derivatives equal zero, i.e. at the equilibrium points. [17]

The solvers accepts a number of parameters, among which of particular interest is a number of sub-intervals into which the original interval must be divided. The methods are specified to be unable to discern between double zeroes present in a single bracket ; this has consequences that are discussed in the performances and results sections.

As this package requires that the user first reduce the system to a single-variable equation describing the equilibria, it is mainly intended to be used on the example model as an error detection means. It is not assumed that in the general context in which this interface aims to be used, such an analytical expression will always be available or derivable. In the context of conductance-based models, however, this assumption is reasonable.

Section 2.1.5 : Julia packages - ImplicitEquations & Plots

These packages contain implementations of functions used to output the results of the scientific computations in a graphical form. They are fairly similar to any such implementation in other languages such as MATLAB or Python, and in fact sometimes rely on calls to functions written in these languages. While this is very inefficient, it is accepted as this part of the software is not time-constrained. [18],[19]

While these packages are used as visualization tools and allow verification of the result of the methods used to identify the bifurcation through visual inspection, they obviously aren't part of the identification methods themselves. This would involve complicated graphical methods and be essentially as computationally efficient as a complete space sweep, i.e. highly inefficient.

The ImplicitEquations package contains an implementation of the Tupper method of drawing phase plane graphs of implicit equations, i.e. equations of several variables that are not solved in the form x = f(x). The algorithm handles the parameter space one cluster at a time ; a cluster is a sub-division of the complete parameter space. The size of a cluster is defined through parameters N,M, as they define the level of segmentation of the space.

These functions are used in order to draw the nullclines on the phase plane of the parameters (V_o,n_o) of the system.

The functions of the Plots package are used in order to draw trajectories atop the nullclines on the phase plane, the bifurcation maps resulting in the operation of the bifurcation search and identification algorithm, as well as the transcritical curve separating the regions of the bifurcation map. These elements are detailed further in the second part of this chapter.

Part 2 : Software architecture

Section 2.2.1 : Problem statement

The problem to be solved can be stated as follows : the interface to be built aims at performing the identification of the bifurcations undergone by the system as the excitatory current applied to it increases, in order to determine the excitability types of the neuron as a function of its three parameters (V_o,n_o, ε).

The problem is structured into the following steps:

- The example system must be implemented in order to be studied numerically

➔ Build a module containing the neuron mathematical model, with access to each of its parameters, i.e. the rest positions (v_o,n_o), the time-scale coupling factor ε and the excitatory current I_{app}

- The bifurcation must be localised for a set of parameters (v_o,n_o, ε)

➔ Build a module containing a numerical method that is able to find the bifurcation in the system by manipulating the value of the excitatory current I_{app}

➔ In the final structure, three different stability detection methods are built and are separated from a bisection algorithm wrapper compatible with any implementation

- The equilibria of the system must be located for a value of the excitatory current very close to the location of the bifurcation in order to identify the bifurcation through local properties of the equilibria undergoing said bifurcation

➔ The methods used to locate the bifurcations are based on the number and nature of the equilibria present at a given value of the excitatory current I_{app}, and therefore can be directly used in order to determine the nature of the bifurcation

- A complete bifurcation map must be drawn for the desired values of (v_o,n_o, ε)

➔ Build a common module that performs the previous operations in loops

- Various experimental and exploration methods are required in order to assess the accuracy of the developed methods ; all scientifically significant results must be displayed so that they are easily understood and can be linked with the theory

➔ Build a graphical module that allows the visual inspection of the system through the drawing of the phase plane, simulated trajectories, and the final bifurcation map

A special effort as been applied in order to dissociate and abstract as many parts as possible, so that a given piece of logic may be localized at a single precise position and reused as required. This improves both readability and ease of maintenance of the code.

Section 2.2.2 : Neuron model module

The TC model is implemented using the dedicated Julia syntax for functions. The exact equations that are implemented are the following, i.e. those of expression 8 recalled from section 1.2.12.

$$\begin{cases} \dot{v} &= V - \frac{V^3}{3} - n^2 + I_{app} \quad (\textit{Membrane potential}) \\ \dot{n} &= \epsilon\left(n_\infty\left(V - V_0\right) + n_0 - n\right) \quad (\textit{Activation variable}) \end{cases} \qquad \text{(expr. 8)}$$

The variables of the system are the membrane potential V and the aggregated activation variable n. The parameters of the system are :

- I_{app} : The applied excitatory current - this parameter corresponds to the excitation that the neuron is subjected to, i.e. the total aggregated signal that it receives
- ε : The time-scale coupling factor - this parameter controls the time-scale separation between the mechanism inducing the spike in the action potential, and the mechanism inducing the relaxation of the system after a spike has been generated
- V_o : The half-activation membrane potential - this parameter controls the horizontal position of the N-nullcline
- n_o : The ion channel balance factor - this parameter controls the vertical position of the N-nullcline

Additional methods are implemented in the module. These are used in order to modify the system parameters and expose dedicated functions performing the required computations on the eigenvalues of the system Jacobian matrix using numerical approximation of the Jacobian matrix extracted from the ForwardDiff package.

Section 2.2.3 : Formalisation of the bifurcation research objective function

Most research problems can be identified with some form of root-finding problem through appropriate formalisation of the original problem. A number of solutions have been proposed for this class of problems, with the corresponding algorithms making up a large part of the field of numerical analysis. While the exact hypothesises required to use a given algorithm vary, they all have in common that they require an objective function to work on. This objective function can take many shapes, but must have its root correspond to the researched location in order for the procedure to have any meaning, possibly through a number of changes of variables. **([16])**

In the context of this work, the objective function must be an image of the global stability of the system in terms of equilibrium points : considering a growing excitatory current, the root of this function must correspond to the transition between the last configuration with at least one stable equilibrium point and a configuration with no equilibrium at all. In this last configuration, any trajectories must follow a stable cycle, as stated by the Poincaré-Bendixson theorem.

Section 2.2.4 : Candidate objective - eigenvalues

From the theory of non-linear systems, it is known that the eigenvalues of any equilibrium that is going to transition to instability through a bifurcation display the following behaviour, as discussed in sections 1.2.8 and 1.2.10 :

- their real part increases monotonously from real negative values towards zero
- they reach zero right at the bifurcation
- for a HOPF bifurcation, they keep increasing after crossing the zero
- for a SNIC bifurcation, the equilibrium and its eigenvalues do not exist anymore

While the equilibrium involved in a HOPF bifurcation could in theory be tracked and its eigenvalues studied before and after the bifurcation, it simply cannot be done for SNICs, as the equilibria disappear completely. Using the eigenvalues to construct an objective function is therefore impossible.

Section 2.2.5 : Candidate objective - oscillation frequency

As discussed in sections 1.1.5 and 1.2.4 presenting the neuronal excitability types, a neuron subject to an excitatory current will start from quiescence and start spiking at frequencies that depend on its excitability type. Therefore, the current-frequency curve of an excited system consists in an image of a stability function : it displays a constant zero value under the bifurcation, then either a monotonous growing curve for type I neurons, or a jump and quasi-constant curve for type II neurons, as illustrated in Fig. 5, section 1.1.5.

While the frequency function is defined on the whole space of values of the applied current, the fact that it is equal to zero on the lower range of currents leading up to the bifurcation makes it inadequate for most root-finding algorithms, such as the secant or newton methods, as they require non-trivially-zero functions on both side of the zero.

What is more, computing the oscillation frequency of the system at every chosen value of the applied current requires a complete simulation of a trajectory of the system. This is extremely costly in terms of computing time, barring a greater mastery of the simulation parameters and precision requirements.

Section 2.2.6 : Candidate objective - global stability

The simplest option is to forgo numerical functions altogether and to settle for the simplest boolean function : the function defined by the boolean criterion of whether the system is globally stable, or not.

This, and the above-mentioned arguments, imply that the only root-finding algorithm that can be used is that of the bisection. What is more, the function value, i.e. the global stability of the system, has to be assessed for each chosen value of the applied current. The implementation of the corresponding stability detection function is covered in following dedicated sections, and justifies the separation of the search algorithm into a separate module (one per implementation), as described in the problem statement. **[App]**

Section 2.2.7 : Bisection applied to the final candidate

The bisection solution, as with most algorithms, has both advantages and drawbacks. The main advantage is that is it infinitely robust, i.e. that it is guaranteed to converge as long as its required hypothesises are met. These consist in the assurance that the algorithm is given a starting range that is guaranteed to contain the zero, and that the objective function sign is constant and opposite on both sides of the zero. [16]

The main disadvantage is its convergence rate. Compared with algorithms that require explicit non-constant, non-boolean functions such as the secant or Newton-Raphson method which display convergence rates of about 1.618 and 2, respectively, the bisection method's rate of convergence is of 1, i.e. it displays linear convergence. As a matter of fact, the bisection method computes an additional bit of precision on each iteration. [16]

Section 2.2.8 : Common module - Bifurcation search

A flowchart illustration of the logic path corresponding to the bifurcation search and identification algorithm is presented in Fig. 14 and detailed in the following paragraphs.

In order to guarantee that the bisection algorithm is applied to an interval that contains one, and only one bifurcation, a sweep is first performed. As a sweep is a highly inefficient means to navigate the variable's space, the step is kept as loose as possible w.r.t. to the spacing of bifurcations ; that is, it must be short enough to prevent that two bifurcations be present in a single interval, lest the bisection interval fail to perform as expected. The matter of quantifying the width in terms of excitatory current which separates two given bifurcations in the system is a complicated one, and no appropriate documentation on the subject has been found during the course of this work.

On the other hand, while choosing a shorter sweeping step makes the implementation less efficient, it is certain to improve its accuracy. Therefore, the step is chosen experimentally as the greatest value outputting the same results as a very strict step value of e.g. 0.01. The value meeting this criterion is found to be of approximately 0.5, thus avoiding a considerable factor of 50 times the number of iterations, had the stricter step been kept out of prudence.

The corresponding operations are performed through the "intervalFetcher" function of the common module.

A first hypothesis is required in order to guarantee those of the bisection algorithm : that the first equilibrium found on the lower border of the excitatory current range be considered stable. This is used in order to have the sweep detect the first bifurcation as the loss of stability of this first equilibrium. This corresponds to the hypothesis that for a low value of the excitatory current, the neuron is in a quiescent state. This is physiologically relevant for all neurons, and therefore admitted without further discussion. (pathological cases of neuronal models displaying their rest range at lower values of the applied current can simply be shifted by a change of variable consisting in a constant increment)

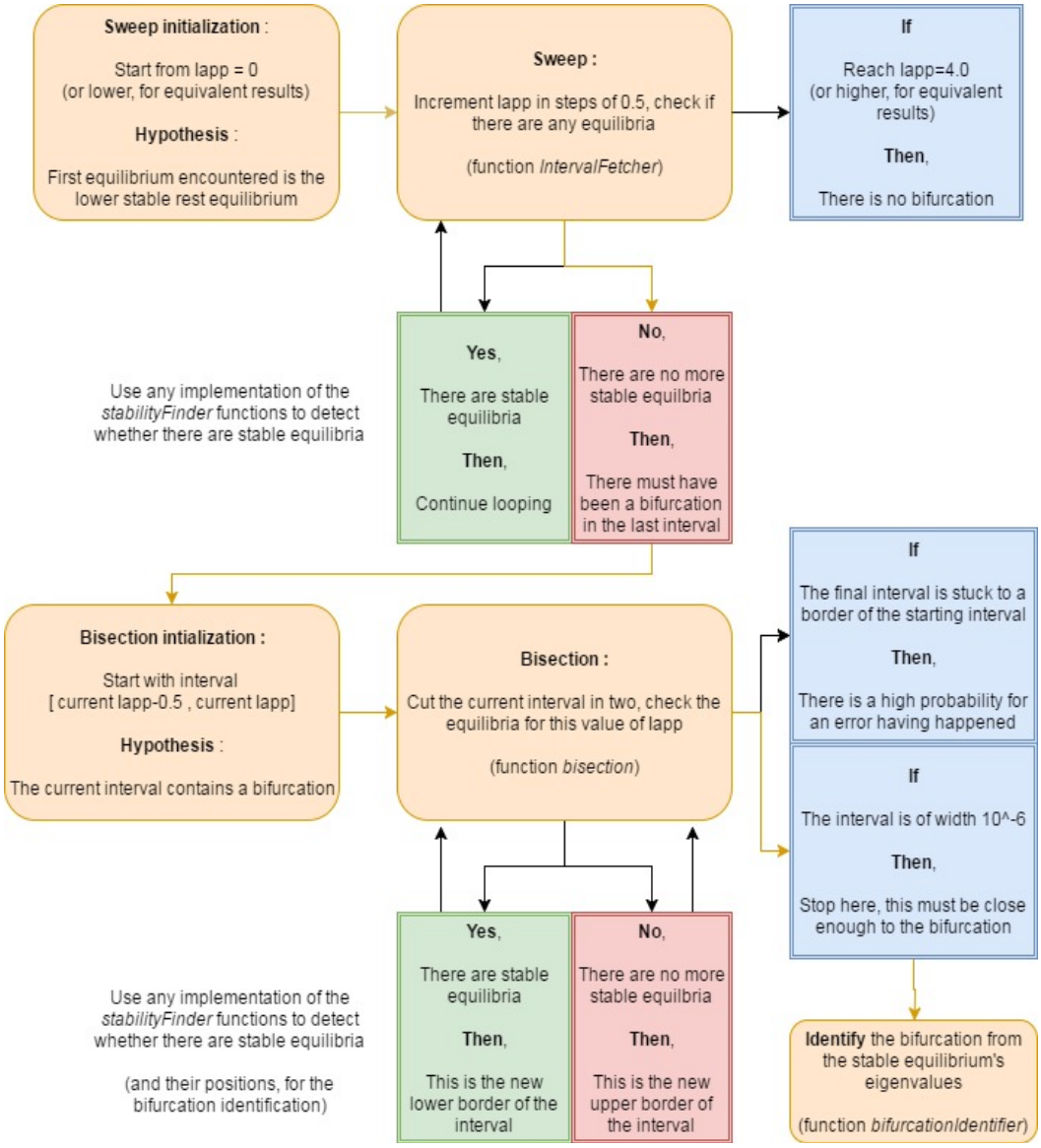

Fig 14: Flowchart depiction of the bifurcation search and identification algorithm logic path. The algorithm is started in the top-left position, and the ideal path is coloured in orange. The algorithm can be seen to be structured in two main steps, i.e. the sweep and bisection iterative methods, with conditional clauses leading either to progression, to looping, or to error and exit. The initialisation conditions of the two iterative methods as well as the associated hypothesises are depicted as parts of the orange main path.

Assuming that the objective function is defined, implementing the bisection algorithm is a simple matter of applying its definition to the situation : the variable is the excitatory current applied to the neuron, and the output is the truth value corresponding to whether the system oscillates or not. The algorithm consists in cutting the interval in two, and keeping the half that still contains the root, i.e. whose borders respectively correspond to a stable and an unstable configuration. **[App]**

The precision required of the bisection algorithm, i.e. the desired width of the resulting interval assumed to contain the bifurcation, doesn't require strict quantification, as it is the nature of the eigenvalues and not their exact value that is needed in order to identify the bifurcation. However, the interval must be short enough that the local properties of the bifurcation may be observed. As for the width between two given bifurcations, appropriate documentation on the subject could not be found during the course of this work. Therefore, a rather strict width of 10^{-6} has been picked ; this value was experimentally found to be the largest value inducing no additional noise in the final bifurcation map, starting from stricter values of the precision criterion. The consequences of such a high degree of precision are much lighter than if applied to the sweep, since the bisection displays linear convergence. The number of additional steps required for e.g. an increase in precision from 10^{-2} to 10^{-6} can be computed using the formula present in the appendices, section 2.1, and is of about 14 additional steps.

The corresponding operations are performed through the "bisection" function of the common module.

Using the interval computed through the bisection algorithm, the bifurcation must then be identified. This is performed through both evaluation of the eigenvalues of the node involved a bifurcation, i.e. the pattern of its eigenvalues are matched to those presented in the previous sections concerning the subject, as well as by computing a single simulation at the lower border of the bisected interval. This simulation aims to detect if any stable cycle exists before the bifurcation, i.e. to differentiate between SNICs and SN-SH bifurcations.

The corresponding operations are performed through the "bifurcationIdentifier" function of the common module.

Finally, a simple "parameterSpaceMapping" function performs the above-mentioned operations in sequence, for every desired value of the parameters (v0,n0), in order to build a complete map of the bifurcations undergone by the system before entering cycling behaviour.

Section 2.2.9 : Common module - Bifurcation identification

The bifurcation can be reliably identified locally through the computation of the eigenvalues of the Jacobian matrix at the location of the stable equilibrium undergoing a destabilizing bifurcation, i.e. the stable equilibrium existing in the configuration corresponding to the excitatory current being equal to the lower bisected interval.

The eigenvalues are matched against the known patterns, as presented in the previous sections. Provided that the interval is close enough to the bifurcation to have the equilibrium display the associated eigenvalues, this method cannot fail or add any noise.

Additionally, a single simulation is run for the value of the excitatory current at the lower border of the bisected interval in order to discern between a SNIC and SN-SH bifurcation. If the simulation displays oscillations, then this means that a cycle existed prior to the saddle-node bifurcation and therefore that this is a SN-SH.

Section 2.2.10 : Common module - Transcritical bifurcation curve

The transcritical bifurcation curve is computed using the equation extracted from [20] ; it is solved approximately using a simple bisection algorithm, as the code was already developed for the bifurcation search algorithm. It is drawn unto the parameter map, and then a finer numerical mapping is performed around the curve in order to display the SNIC behaviour of the system close to it.

This curve is presented in [20] as a the expression of the boundary separating the restorative and regenerative ion currents majority effect. This is discussed in further details in the third chapter concerning the results of the application of the interface to the canonical TC model.

Section 2.2.11 : ODE-based module - Stability detection function

The results of the ODE-based simulation are the vectors of the trajectories computed through simulation, i.e. paths in the space (V,n). Using only the results of simulations, the stability of the system is assessed as follows :

- Two runs of the ODE simulation are started, respectively initialised on the far bottom-left and far top-right in order to ensure that they lie in the basins of attraction of the bottom and top equilibrium or stable cycle, if these exist
- Each simulation is tested for cycling behaviour by testing whether the potential membrane crosses a threshold value ; this threshold cannot be crossed in the same direction twice, as it is equipped with a latch that is only released when crossing in the opposite direction. The threshold is arbitrarily set as the mean value of the latter half of the vector, in an attempt at ignoring transient regime and catching stable cycles. An additional protection restricts the cycling diagnostic to trajectories displaying more than a few cycles, as some trajectories are expected to be sinuous
- Each simulation endpoint is tested for stability using the eigenvalues of the Jacobian matrix
- If the two simulations display cycling behaviour, the system is assumed to be unstable
- If neither of the two simulations display cycling behaviour, either they land on the same stable equilibrium and therefore the system is assumed to be monostable, or they don't and the system is assumed to be bistable ; they cannot land on unstable equilibria
- If one of the simulation displays cycling behaviour, and the other doesn't, then the system is assumed to have only one equilibrium and a stable cycle
- If a simulation that displays no cycling behaviour lands on an endpoint that fails to be detected as stable through the computations of the eigenvalues of the system Jacobian matrix, then it is treated as if it were cycling, i.e. unstable ; this, however, should never happen

Section 2.2.12 : NLsolve-based module - Stability detection function

The results of the Trust-region and Newton-Raphson methods implemented in the NLsolve package are structures containing the convergence coordinates, a flag confirming that convergence happened, as well as some additional practical information.

The methods are found to require in the order of a dozen iterations to converge on equilibrium points when they exist. Barring this, they fail after a thousand iterations. Using the output of the methods, the stability of the system is assessed as follows :

- Two runs of the trust-region method are started, initialized on the far bottom-left and far top-right in order to ensure that they lie in the basins of attraction of the bottom and top equilibrium respectively, if these exist
- The two runs of the method are checked to have converged ; if either one did not, it is ignored as if the corresponding suspected equilibrium did not exist, as there is nothing to be inferred from the lack of a single convergence
- All runs that converge are tested for stability using the eigenvalues of the Jacobian matrix
- If the two runs fail to converge, the system is assumed to be unstable
- If the two runs converge, either they do so on the same stable equilibrium and therefore the system is assumed to be monostable, or they don't and therefore the system is either monostable or bistable
- If only one run converges, then the system is considered to be globally stable with a single equilibrium point or globally unstable

Section 2.2.13 : Roots-based module - Stability detection function

The results of the Roots-based methods are vectors containing the positions of every equilibrium, stable or unstable, that is found in the system by solving its pre-processed equation corresponding to the conditions of stability. Using these vectors, the stability is assessed as follow :

- The equilibria are checked sequentially and tested for stability using the eigenvalues of the Jacobian matrix
- If there is only one stable equilibrium, the system may or may not also feature a stable cycle ; this cannot be detected using Roots-based methods
- If there is a second stable equilibrium, the system is bistable ; the research is stopped here as more equilibria would only arise from non-physiological artefacts due to the shape of the mathematical model outside its validity range
- If there is no stable equilibrium, the system is assumed to be globally unstable

Section 2.2.14 : Graphical module

The graphical module use functions extracted from the Plots and ImplicitEquations packages in order to allow the visualization of the results of the methods :

- Visual inspection of the system is performed on call through the drawing of the nullclines on the phase diagram, with trajectories drawn atop the nullclines
- The results of the bifurcation search and identification algorithm are displayed on a dedicated figure, in order to determine the different behavioural regions of the systems on the parameter space (v0,n0), as well as their evolution depending on the time-scale coupling parameter ε.

Example outputs of the graphical module are presented in Figs. 15 and 16 in the exact same forms that are used and discussed in the third chapter of this document.

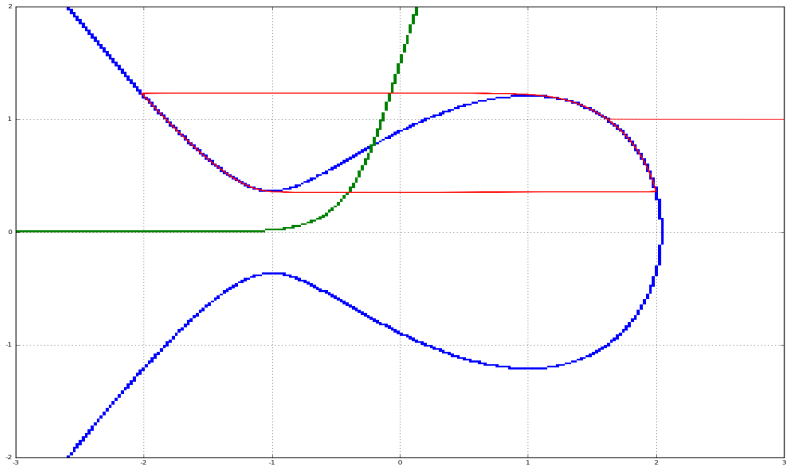

Fig 15: Example of a drawing of the nullclines of the phase plane, with an added trajectory resulting from simulation.

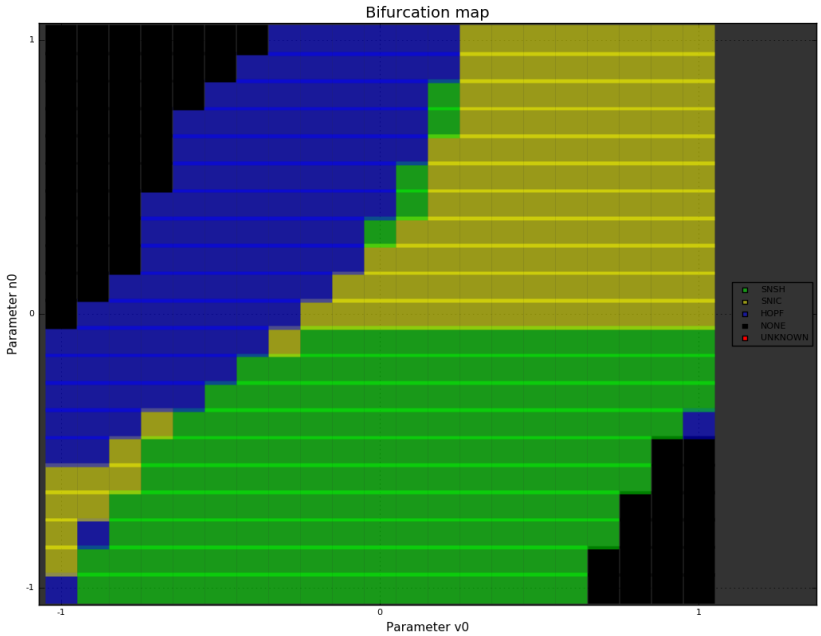

Fig 16: Example of a bifurcation map displayed through the graphical module, after its computation through the interface. See internal legend for details on the regions of the map.

Part 3 : Software performance

Section 2.3.1 : ODE-based utilities

As ODE-based simulations are involved in not only the ODE-based implementation of the stability detection function, but also the bifurcation identification function common to all three implementations, the run times of the main solvers for different values of their parameters are recorded and presented in the tables 1, 2, and 3 right below. The simulations are performed with the system parameters $(v_0,n_0, \varepsilon) = (0,0,10^{-3})$. The first two have no reason to affect the simulation run time, while the third one affects the time-scale separation of the two equations ; therefore, we choose the value that most constraints the problem, i.e. $\varepsilon = 10^{-3}$ corresponding to the strongest time-scale separation, as ODE solvers are known to encounter difficulties with systems displaying strong time scales separation.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Actual number of steps computed by the adaptive solver (in $\log_{10}$)	Runtime (in seconds)
1	4	3.87	0.035396
0	4	4.21	0.049278
-1	4	5.02	0.185367
1	5	4.86	0.339532
0	5	5.21	0.428251
-1	5	6.02	1.808115

Table 1: Simulation run times for the ODE23 solver, ordered by increasing run times, with $(v_0,n_0, \varepsilon) = (0,0, 10^{-3})$. It can be observed that the factor most influencing the run time is the simulation length. Steps of either 1 or 0 cause relatively similar actual number of steps and run times, which can be assumed to be due to the error restriction methods of the adaptive solver.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Actual number of steps computed by the adaptive solver (in $\log_{10}$)	Runtime (in seconds)
1	4	3.82	0.045666
0	4	4.19	0.057822
-1	4	5.02	0.200376
0	5	5.18	0.467218
1	5	4.79	0.526634
-1	5	6.02	1.748854

Table 2: Simulation run times for the ODE45 solver, ordered by increasing run times, with $(v_0,n_0, \varepsilon) = (0,0, 10^{-3})$. Observations to be made are the same as for table 1. The penultimate and ante-penultimate rows are interchanged, for no discernible reason other than numerical variability of fairly similar run times.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Actual number of steps computed by the adaptive solver (in $\log_{10}$)	Runtime (in seconds)
1	4	3.13	0.011032
0	4	4.01	0.081421
1	5	4.01	0.083583
0	5	5.00	0.657585
-1	4	5.00	0.717028
-1	5	6.00	7.022025

Table 3: Simulation run times for the ODE23s solver, ordered by increasing run times, with (v_0,n_o, ε) = (0,0, 10^{-3}). Run times can be seen to be ordered following the number of step, and that the number of steps correspond nearly exactly to the length/step ratio, as expected from a stiff solver.

Compared to ode23 and ode45, it can be observed that the ode23s is faster for large steps, and slower for small steps ; this can be directly identified to the adaptive solvers' property of increasing the number of steps in case of errors detected during the integration. The simulation length seems to affect the computation time the most.

From the results of the ode23s solver, it is assumed that the problem does not identify as stiff equations, and that there is no need for the additional numerical stability of the stiff solvers. Additional performance could have been achieved by explicitly specifying the analytical expression of system's Jacobian matrix, as is advised for the use of stiff solvers ; however, it is assumed that an explicit expression of the Jacobian matrix may not be easily derived for any system, and therefore this is not explored further.

From the results of the ode23 and ode45 solvers, it is concluded that a step of 10^1 hampers the accuracy of the solution, and that it is therefore insufficient ; although in practice, the adaptive solver corrects the insufficient precision automatically. Since it does not induce higher computation times, the ode45 solver is used for its greater precision.

As the ODE package supports the entire simulation capabilities of the interface, an example simulation-based visual inspection is presented in Fig. 17. While the nullclines can be seen to be only roughly defined due to the cluster-based Tupper method, they are entirely sufficient for visual inspection of the system.

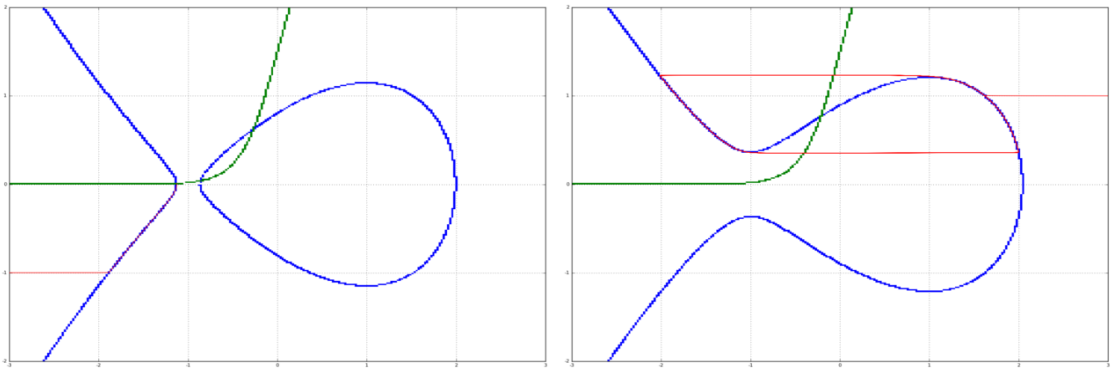

Fig 17: Example of the graphical display of a trajectory obtained through ODE-based simulation, drawn atop the nullclines of the system on the phase plane diagram computed through the ImplicitEquations-based functions. The simulation corresponds to (v_0,n_o, ε) = (0,0, 10^{-3}), with a simulation step of 10^1 and length of 10^4. The system can be seen to go from rest to spiking, as the trajectory originally resting for I_{app} of 0.65 (left) start cycling for I_{app} = 0.8 (right)

Section 2.3.2 : ODE-based implementation

The first observation to be made regarding the accuracy of the ODE-based implementation is that the function used to assess whether a trajectory displays cycling behaviour is not well-motivated theoretically. It is the result of an early attempt that was not pursued further, in favour of work on the two other implementations of the stability detection function. In particular, the choice of the threshold used to detect a given cycle in the system to be the mean value of the latter half of the vector has strong reasons to be disputed.

Furthermore, lacking a dedicated convergence assessment method, transient oscillations cannot be distinguished from stable cycles and misclassification is bound to happen. Convergence assessment methods based on the manipulation of the vectors representing the trajectory do exist, but the research in this direction was not pursued, again in order to work on the two other implementations.

Finally, the study of trajectories obtained through simulations is complicated by the fact that convergence towards equilibria in the system can take a variable amount of time. In certain cases, proximity of the trajectory to the ghost of a SNIC bifurcation, as presented in section 1.2.8, can slow down the trajectory to near-zero speed. Therefore, all the above-mentioned methods should be used along with an active monitoring of the simulation which could detect this and adapt the simulation length, or use any other way to work around this type of problem.

The run times of a full bifurcation mapping using the ODE-based implementation of the stability detection function for different values of its parameters are recorded and presented in the table 4.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Time-scale coupling parameter ε (in $\log_{10}$)	Runtime (in seconds)
1	4	-2	~820
1	4	-3	~840
1	4	-1	~1400

Table 4: Bifurcation mapping run times for the ODE-based implementation of the stability detection function. As the tests cannot be easily reproduced due to their duration, the exact run times must only be taken as indications of orders of magnitude. The simulation step and simulation length where chosen in order to induce the lowest total run times, in order to restrict these to a reasonable duration.

Since this implementation is entirely based on the ODE package, the parameters of the simulation directly and very strongly affect the total computation length of the bifurcation map. The (step,length) = (1,4) parameter combination was chosen as that corresponding to the lowest run time of the simulation ; assuming as a lower border that the total computational time evolves linearly w.r.t. to the simulation time, using combinations of (-1,4) or (1,5) in order to improve accuracy would induce total run times of at about five times the amount of the (1,4), i.e. about one hour in the best cases.

The bifurcation maps corresponding to the first three tests recorded in table 4 are presented in Fig. 18.

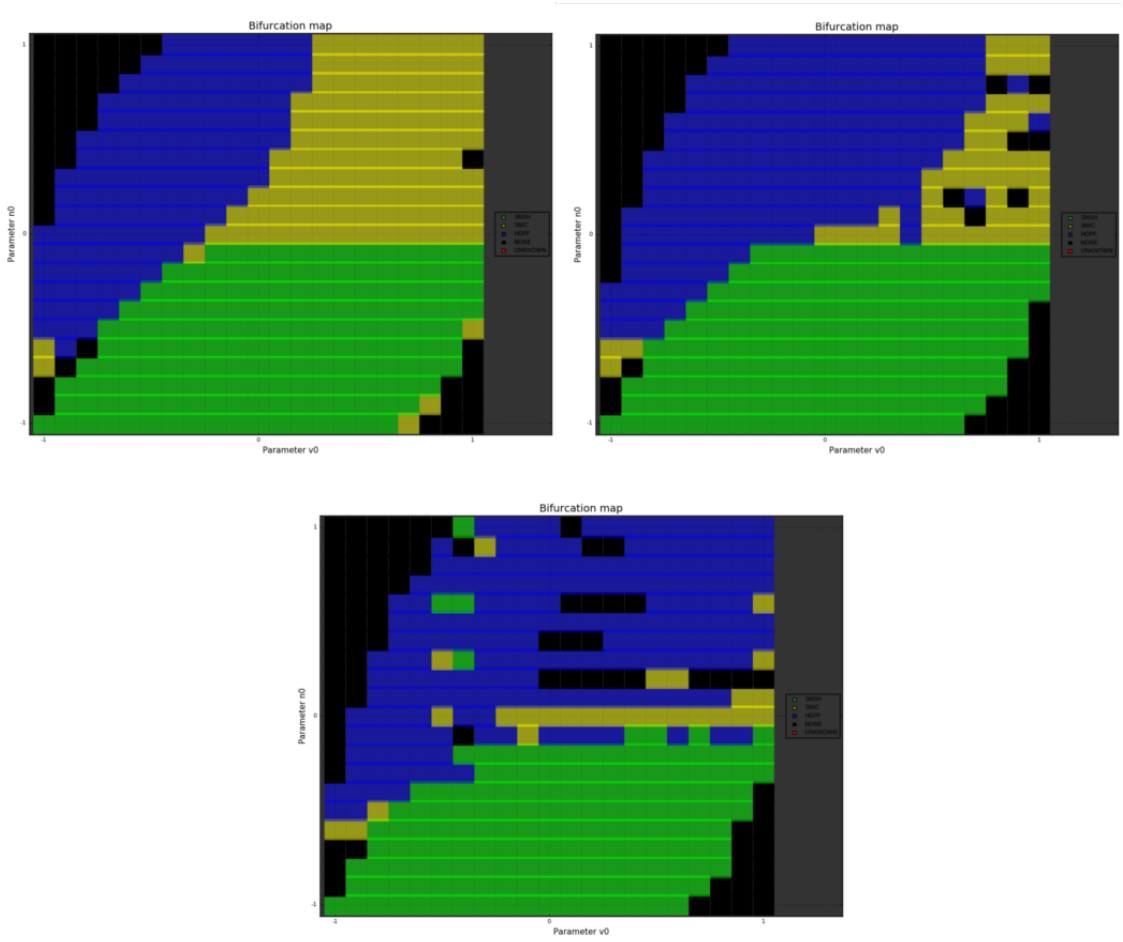

Fig 18: Bifurcation maps corresponding to the test runs recorded in table 4. The time-scale coupling parameter ε has a value of 10^{-1} (top-left), 10^{-2} (top-right) and 10^{-3} (bottom). Well-defined regions are observable to the naked eye, but noise appears and expands in the upper-right quadrant as the parameter ε decreases.

Misclassification noise can be easily identified on the top-left illustration in the bifurcation maps in Fig. 18. These are the points that punctually intrude in otherwise well-defined regions of a single colour.

The quantity of noise seems to display an inverse relationship with the time-scale coupling parameter ε : as ε decreases, the bifurcation map gets cloudier.

Through manual testing, it is observed that the number of cycles found in cycling systems of identical parameters, and with an identical simulation configuration, decreases sharply with the parameter ε, down to only a few cycles for $\varepsilon=10^{-3}$. This corresponds to the observed noise : the system is found to be cycling only when a given number of turns are measured, therefore the system is considered to become unstable only after the bifurcation has happened, and it cannot be identified reliably as the interval detected as containing the bifurcation actually doesn't.

Furthermore, this means that the simulation length for all implementations should exceed 10^4 in order to detect cycles in SN-SH accurately for $\varepsilon = 10^{-3}$.

Section 2.3.3 : NLsolve-based implementation

Arguments to be made regarding the accuracy of the NLsolve-based implementation mainly consist in the fact that ghosts of SNIC bifurcations were experimentally found to occasionally slow down the progress of the line search methods so much that the algorithms could not converge under a thousand iterations, and therefore abort their computations. This is a known property of real non-linear systems, as it arises from the fact that the vector field in both dimensions reduces to near-zero values when the trajectory travels in a zone that is constricted by both nullclines at once, in a bottleneck-like fashion, as documented in [8],[9]. Other cases of the algorithm failing to to converge for no discernible reason have been observed ; the convergence of line search algorithms is a complex topic of its own and is therefore not covered in this document.

While it is not required in order to locate the bifurcation, an additional problem that would hamper a exclusively NLsolve-based method is the fact that it cannot detect cycles. However, this is alleviated by the fact that the bifurcation identification function comports a single simulation in order to check for the existence of stable cycles before the bifurcation.

The run times of a full bifurcation mapping using the NLsolve-based implementation of the stability detection function for different values of its parameters are recorded and presented in the table 5.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Time-scale coupling parameter ε (in $\log_{10}$)	Runtime (in seconds)
0	5	-3	~40
-1	4	-3	~60
0	5	-2	~60
-1	4	-2	~100
0	5	-1	~130
-1	4	-1	~180

Table 5: Bifurcation mapping run times for the NLsolve-based implementation of the stability detection function. The parameter ε can be seen to have the strongest impact on the run time, with the simulation step coming second.

Observing the results in table 5, the parameter that seems to be affecting the total computation time the most is the parameter ε. Following the observation in the last section that lower values of ε induce less cycles, it is assumed that the system evolves at a slower pace for lower values of the time-scale coupling parameter, thus shaping the vector field in such a way that the line search method reaches the equilibria faster, e.g. so that the vector field is straighter towards the equilibria rather than with strong magnitude in the directions perpendicular to the gradient descent.

The run times are all-around considerably lower than those obtained through the use of the ODE-based implementation.

The bifurcation maps corresponding to the six tests recorded in table 5 are presented in Fig. 19. As the tests for the same values of ε actually output the exact same bifurcation maps, the contents of Fig. 19 are simplified to three maps.

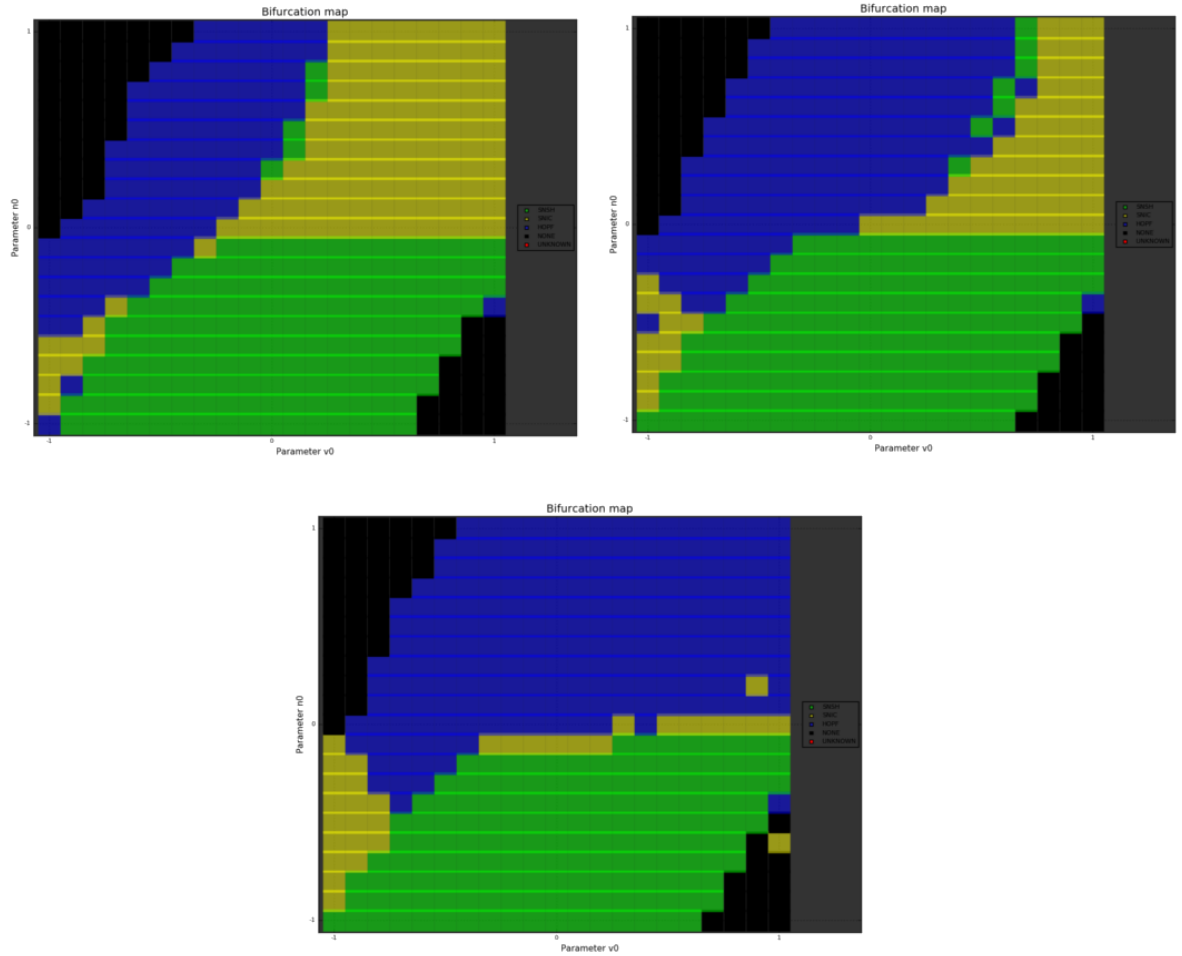

Fig 19: Bifurcation maps corresponding to the test runs recorded in table 5. The time-scale coupling parameter ε has a value of 10^{-1} (top-left), 10^{-2} (top-right) and 10^{-3} (bottom). The six tests are reduced to three maps, as the maps for identical values of ε are themselves identical. Well-defined regions are observable to the naked eye, with slightly noisy frontiers between the regions.

Overall, very little noise is observed on the maps. The regions are well-defined, and only their frontiers are affected by relatively considerable noise. This misclassification noise is attributed to the transition between the regions and can be explained by a continuity argument.

For example, the SN-SH present at the frontier between the upper SNIC and HOPF regions may be the results of systems that are found to start oscillating before the bifurcation, while the complex conjugates eigenvalues characteristic to HOPF bifurcation are not yet observed, due to their presence only very close to the bifurcation.

This corresponds to the continuous phenomenon of complex conjugates eigenvalues appearing after the bifurcation, then right on it, and slowly appearing earlier until they can be detected by the algorithm, depending on its precision.

Section 2.3.4 : Roots-based implementation

The accuracy of the NLsolve-based implementation is strongly affected by the propriety of the available solvers extracted from the package to be unable to discern between two equilibria present in a single bracket. The exact configuration leading to frequent misclassification errors in the top-right quadrant of the bifurcation map is developed and discussed in the last chapter of this document concerning the scientific output of the methods, as it corresponds to a zone that displays a behaviour of significant scientific importance.

Increasing the explicit required number of brackets from 200 to 20 000 has no apparent effect, as if there were a built-in limit that was reached in both cases, and that no more precision could be obtained from the method. Verification of this hypothesis, and further inquiry into the problem, would require heavy involvement with the source code of the package which was avoided in favour of work on the other implementations.

As with the NLsolve-based methods, this implementation cannot detect cycles, and therefore requires the addition of a simulation to the bifurcation identification function in order to distinguish between SN-SH and SNICs.

The run times of a full bifurcation mapping using the Roots-based implementation of the stability detection function for different values of its parameters are recorded and presented in the table 6.

Simulation step (in $\log_{10}$)	Simulation length (in $\log_{10}$)	Time-scale coupling parameter ε (in $\log_{10}$)	Runtime (in seconds)
-1	4	-3	~770
-1	4	-2	~820
-1	4	-1	~830
0	5	-3	~850
0	5	-2	~910
0	5	-1	~1100

Table 6: Bifurcation mapping run times for the Roots-based implementation of the stability detection function. The tests are performed for a required 200 brackets, as varying greater values of the parameter controlling the number of sub-intervals produce identical results. The parameter ε can be seen to have the strongest impact on the run time, with the simulation length coming second.

Observing the results in table 6, the parameter that seems to be affecting the total computation time the most is the parameter ε. This parameter, however, is not involved in this implementation as the equations are pre-processed to a single variable problem, such that the parameter ε is excluded from the final expression. It is a constant multiplying the whole equation that has to be equated to zero, and therefore is directly simplified.

Since benchmarking code in Julia is a fairly difficult and imprecise exercise without mastering the explicit power user compilation functions, the differences could as well be attributed to unrelated numerical variations.

The bifurcation maps corresponding to the six tests recorded in table 6 are presented below, in Fig. 20 :

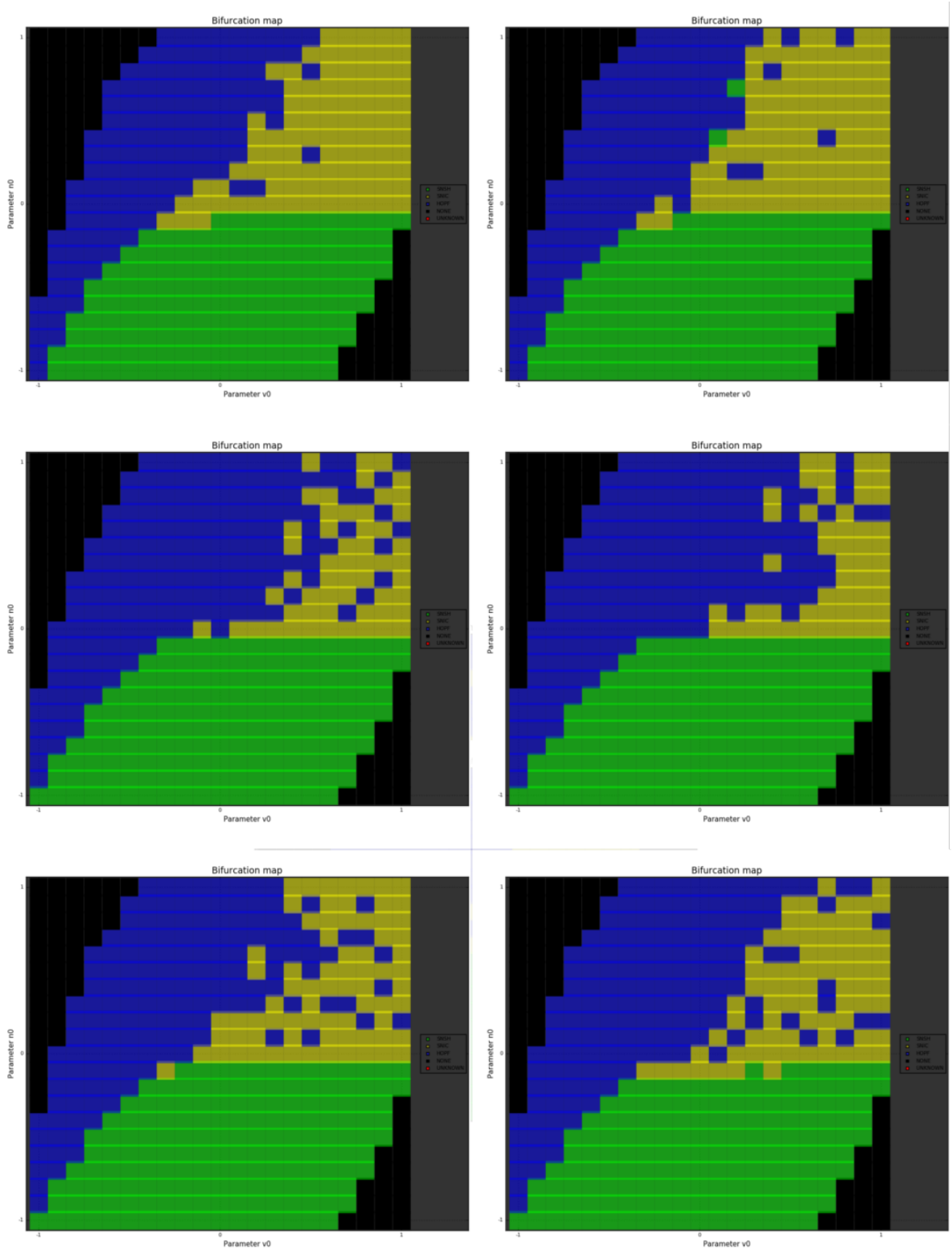

Fig 20: Bifurcation maps corresponding to the test runs recorded in table 6. The time-scale coupling parameter ε has a value of 10^{-1} (top), 10^{-2} (middle) and 10^{-3} (bottom). The simulation parameters have values of (0,5) (left side) and (-1,4) (right side). Well-defined regions can be observed, except for the very noisy top-right quadrant.

Chapter 3 : Scientific results

This third chapter concerns the application of the bifurcation identification interface to the example canonical TC model and the evaluation of the subsequent scientific results.

In the first part of this chapter, the results for a very strong time-scale separation of the system mechanisms are discussed and compared to the results from the scientific literature covering the case of singular limits, i.e. time-scale separation approaching infinity. Extensive visual inspection of the system is performed in order to support, but not replace numerical analysis, and as a way to familiarize the reader with the system visual representation w.r.t. to its behaviour. The upper part of the bifurcation map is found to consist in a type II excitability region, therefore challenging the usual identification of the quadratic and horizontal phase plane local configuration with type I excitability.

In the second part of this chapter, the results for lower time-scale separation are evaluated and discussed with a special focus on the assumptions underlying the canonical TC model w.r.t. to those of other well-known neuron models. The results from two of the three implementations highlight the appearance and growth of a type I excitability region in the previously type II-exclusive region as the time-scale coupling parameter increases. This points out the common practice of exclusively static visual analysis present in the literature as an error borne out of the lack of use of precise, quantitative numerical analysis concerning the identification of bifurcations underlying neuron models.

Part 1 : Comparison with earlier works

Section 3.1.1 : Bifurcation maps

In order to compare our results with the earlier work covered in [8], the bifurcation map corresponding to a very strong time-scale separation is first computed, i.e. with a value of the time-scale coupling factor ε of 10^{-3} approximating the singular limit which corresponds to a value approaching zero. The computation is performed through the Roots-based and NLsolve-based implementations. The ODE-based implementation is ignored, as its results are qualitatively similar, and as it is both much longer to compute and subject to noise whose nature and source are covered in section section 2.3.2.

The results are illustrated in the Figs. 21 and 22.

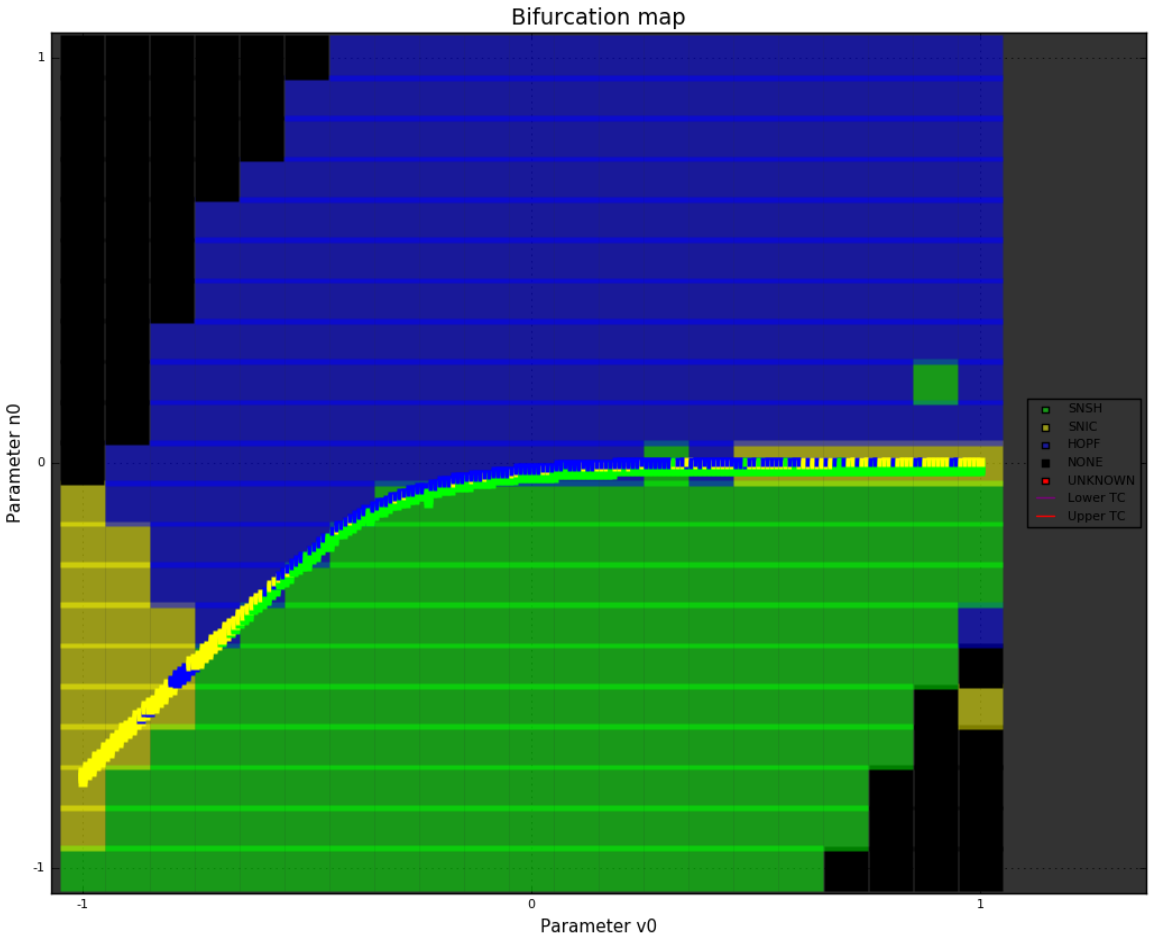

Fig 21: Bifurcation map of the parameter space (V_0,n_0) computed with the NLsolve-based implementation, for ε=10^{-3}. See internal legend and axis for details..The transcritical bifurcation curve is added atop the bifurcation map ; it is first approximately computed through bisection, then a local bifurcation mapping is performed on small regions bordering the curve. The following regions can be observed : a lower region with SN-SHs, an upper region with HOPFs, and a bottom-left region with SNICs. The transcritical curve highlights a thin band of SNIC bifurcations.

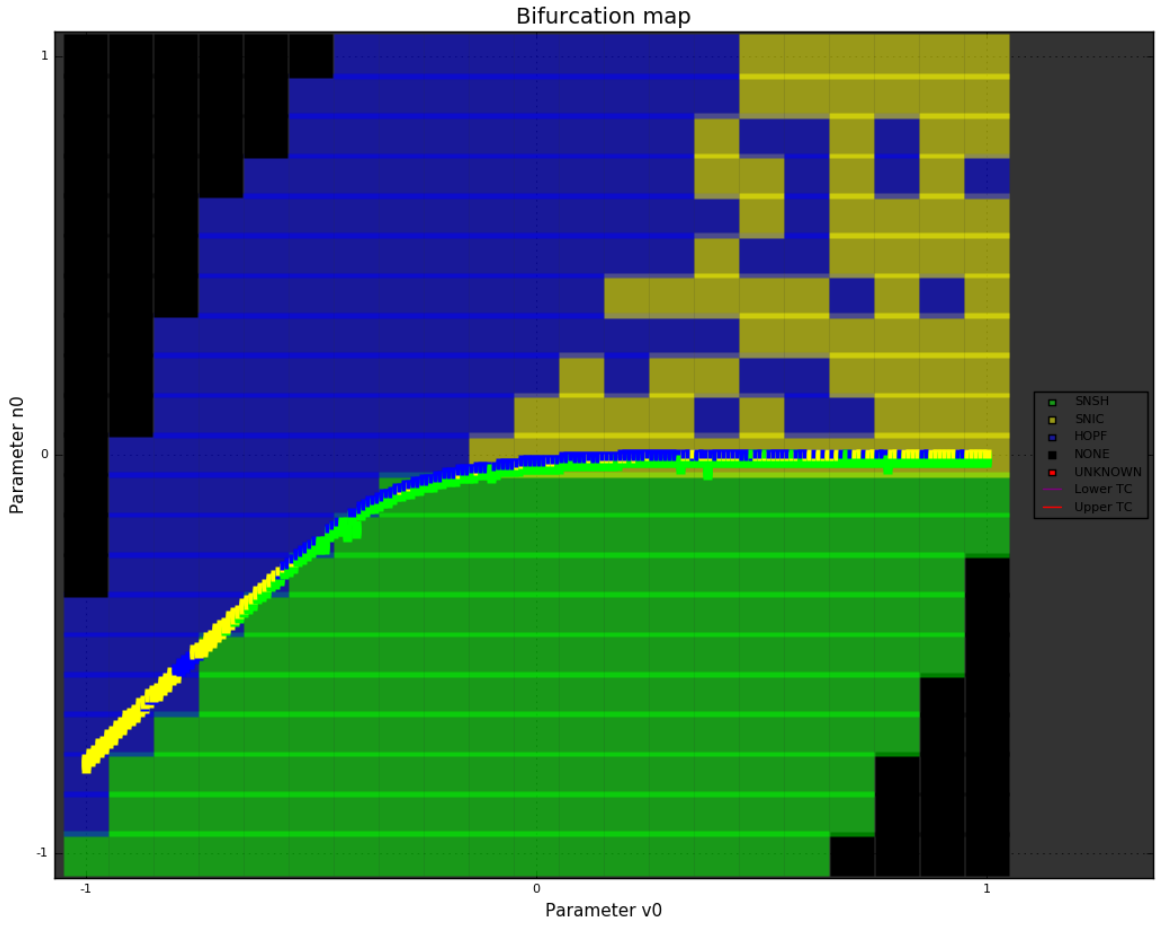

Fig 22: Bifurcation map of the parameter space (V_0,n_0) computed with the Roots-based implementation, for $\varepsilon=10^{-3}$. See internal legend and axis for details. The transcritical curve and local bifurcation mapping are added through the same process as for Fig. 21. The regions observed differ from Fig. 21, as there is no bottom-left SNIC region while there seems to be a top-right SNIC region.

The first observation that can be made is that these results roughly coincide with the theoretical results of [8], as presented in Fig. 13 in section 1.2.13. Looking at Fig. 21, it can be observed that the curve corresponding to the transcritical bifurcation, computed as specified in [9], separates the parameter space into two zones. The upper part corresponds to mainly restorative behaviour, as in the Hodgkin-Huxley model. The lower part corresponds to mainly regenerative ionic behaviour, as in the novel excitability types proposed in [8].

There are, however, a few discrepancies in the placement and nature of the regions w.r.t. the visual analysis on the lower left and upper right corners of the plane : there are two zones of SNIC bifurcations of a shape that do not correspond exactly with the visual analysis. Fig. 21 shows that the upper-right zone is much smaller than expected in Fig. 13, while Fig. 22 show a larger but extremely noisy region and therefore cannot be trusted without extensive verification.

As a matter of fact, the noise present on the upper-right quadrant of the bifurcation map in Fig. 22 is induced by the lack of accuracy of the Roots-based implementations when it has to detect equilibria situated in a single bracket, as discussed in section 2.3.3. The nature of the phenomenon requiring such a high degree of precision is developed in section 3.2.2 in the second part of this chapter, as it is more easily explained after observing the evolution of the system with the parameter ε covered in the section 3.2.1.

In order to assess which regions are actually displayed by the system, and which are the results of misclassification, manual verification is required. During the course of this work, a great number of trajectories have been drawn unto the phase plane, atop the nullclines, and examined ; the intermediate results of the computations leading to the construction of the bifurcation map have been thoroughly controlled, as visual inspection by no means consists in a valid proof.

While the many occurrences of these verification procedures have not been systematically documented, and since the great number of those that were cannot be presented in an efficient and meaningful way, only a restricted number of trajectories are presented in the following figs. 22 to 35 as a way to both try and convince the reader that systematic verification was indeed performed, or at least that the tools to do so are available, and for illustrative purposes regarding the behaviour of the system in the regions defined on the bifurcation maps of Fig. 21 and 22.

For all Figs. 23 to 36, the presentation is structured as follows :

- The left-hand graph is a depiction of the V-nullcline (in blue) and N-nullcline (in green) of the system on the phase plane, with an added trajectory (in red) obtained through simulation from a far bottom-left origin. The horizontal axis displays the value of the parameter V_o, while the vertical axis displays the value of the parameter n_o.
- The right-hand graph is a recording of the temporal evolution of the membrane ; it corresponds exactly to the trajectory displayed on the phase plane. The horizontal axis displays the index of the simulation steps, while the vertical axis displays the value of the membrane potential.
- The figure may be structured an upper and lower part, in which case the lower part corresponds either to a second simulation from a far top-right origin, or a zoom on the behaviour of the system in the close vicinity of the bifurcation

These presentations details are not recalled in every legend of the figures in order to prevent the associated information from uselessly cluttering the legends.

Section 3.1.2 : Lower SN-SH region

The lower region is identified as a SN-SH region by both implementations, as shown in Figs. 21 and 22, as well as by the ODE-based implementation, looking back at the results presented in Fig. 18 in the section 2.3.2 concerning the assessment of the performances of this implementation.

The coordinates chosen in order to observe the behaviour of the region are (V_o,n_o) = (0.0,-0.5) , i.e. roughly in the middle of the region. The figures to be studied for this region are the following :

- Fig. 23 : I_{app}=0.5 - At this value of the excitatory current, the system is still at rest and at a safe distance from the bifurcation in terms of current. A simulation from both a bottom-left and top-right origin are shown in order to assess the existence of a lower equilibrium, but of no upper stable cycle.
- Fig. 24 : I_{app}=0.85 - This value of the excitatory current corresponds to a configuration that is observed right before the bifurcation, considering a continuous increase in the applied excitation until the system destabilizes. A simulation from both a bottom-left and top-right origin are shown in order to assess the existence of both a lower equilibrium and of an upper stable cycle. As this cycle is not created following a global destabilisation of the system, it must have been born from a saddle-homoclinic bifurcation involving the upper saddle.
- Fig. 25 : I_{app}=1.2 - At this value of the excitatory current, the system is spiking and at a safe distance from the bifurcation in terms of current. As the N-nullcline only intersects the V-nullcline on its upper branch, a second simulation is not required. Looking back at Fig. 24, it can be assumed that the lower equilibrium and adjacent saddle have both disappeared in a lower saddle-node bifurcation.

As a matter of fact, the pattern characteristic to a SN-SH bifurcation can be observed : the neuron is at rest at low excitation levels, then forms a stable upper cycle without losing its lower stable equilibrium. Finally, the lower equilibrium disappears through a saddle-node bifurcation and the neuron is restricted to spiking activity for any greater value of the excitatory current. The frequency of the spiking activity seems to have increased between fig. 24 and fig. 25, as expected from the linear current-frequency relationship of a SNIC bifurcation.

This region can be identified to the novel "type IV" excitability type proposed the section 5 of [8]. The bistability can be observed in Fig. 24 ; the spike-latency in Fig. 25, as the first action potential of the spiking pattern is wider, i.e. slower to generate and relax. The after-depolarization potential is not observed, as it would require setting the system in a spiking state, then relaxing the applied excitation.

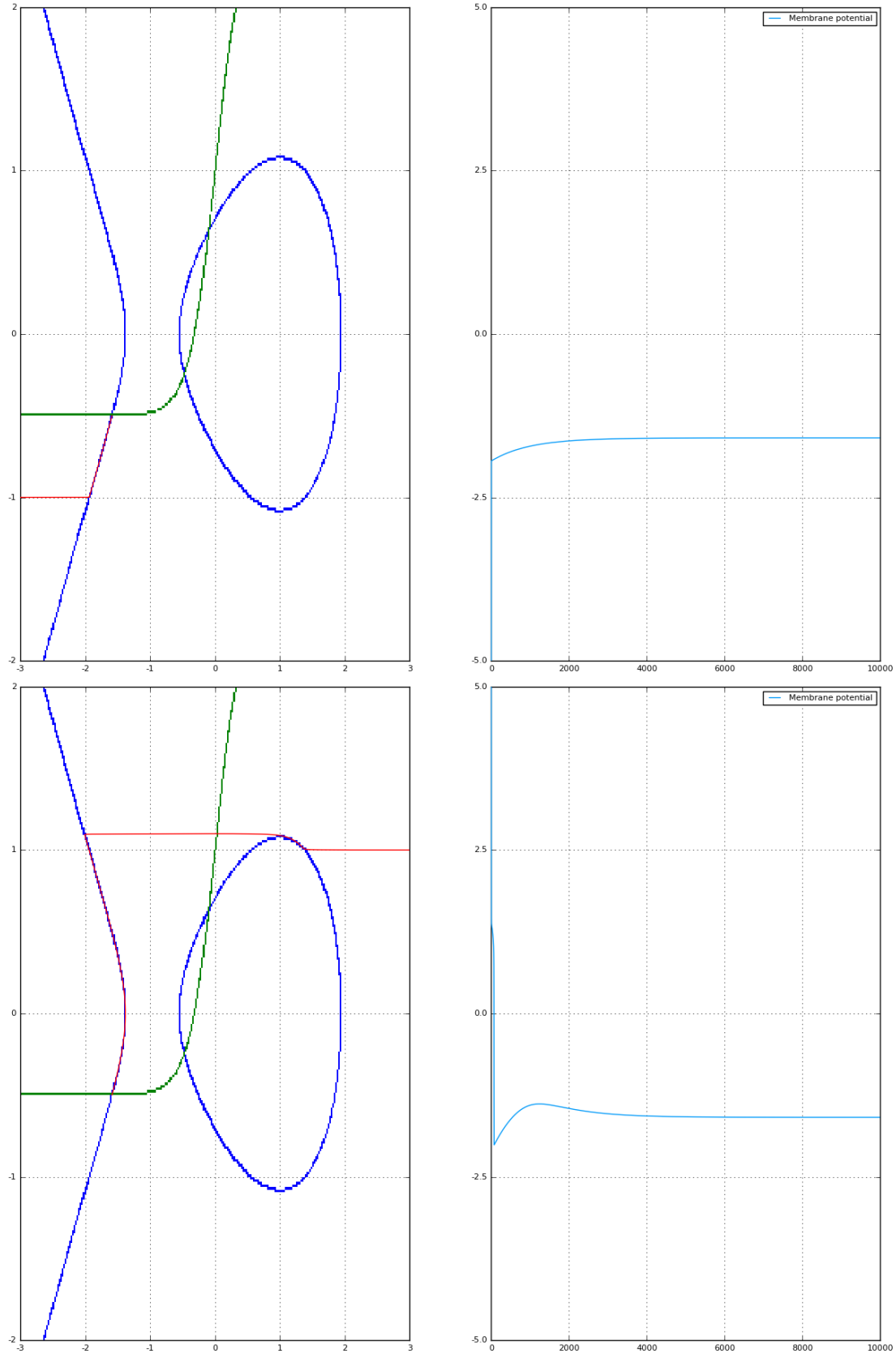

Fig 23: (V_0,n_0)=(0.0,-0.5) - I_{app}=0.5 - See structure description in section 3.1.1. for presentation details. The upper part corresponds to a simulation origin to the lower-left ; the lower part to a simulation origin to the upper-right. The system can be seen to be at rest, as no cycles can be found.

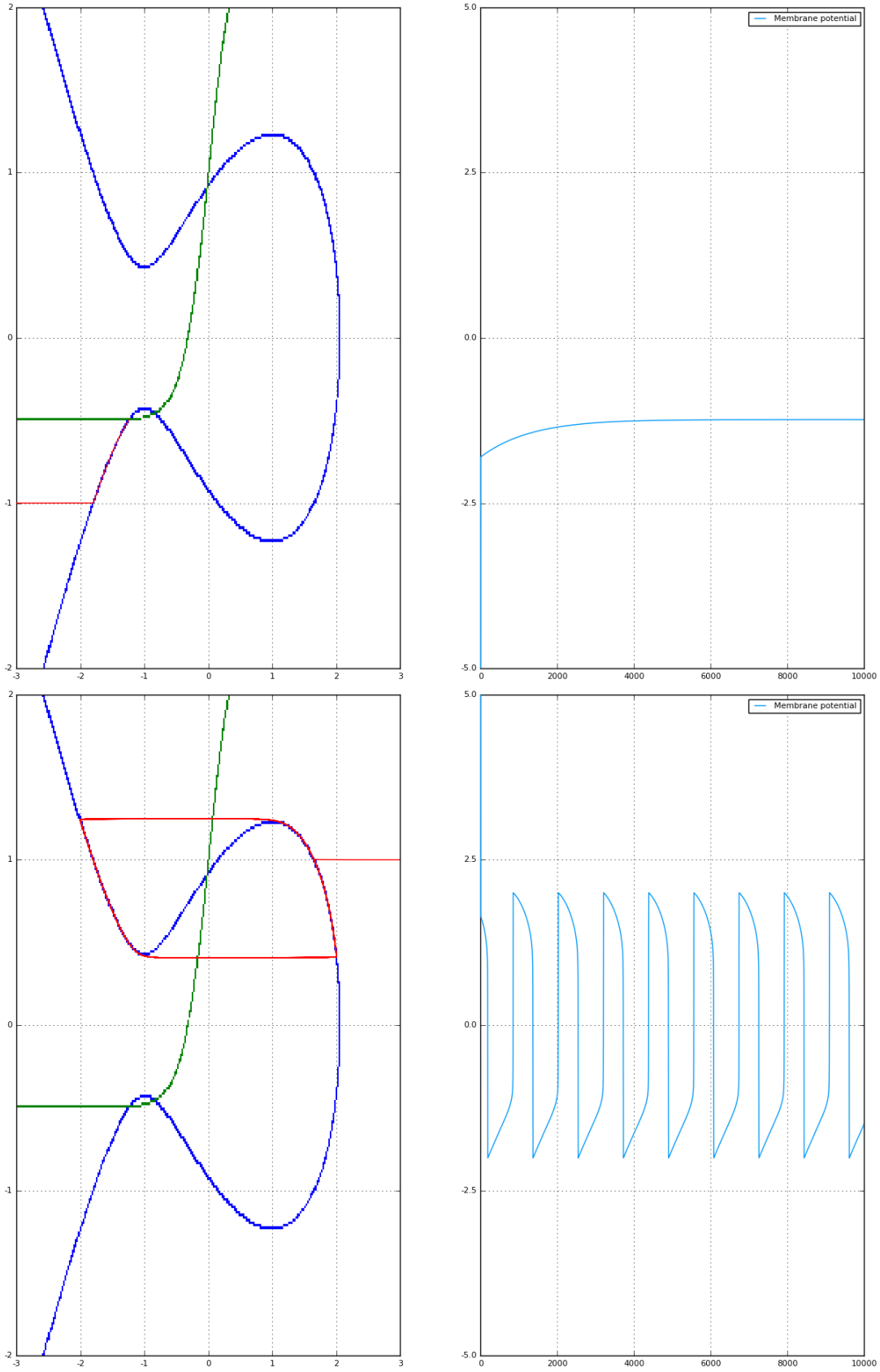

Fig 24: (V_0,n_0)=(0.0,-0.5) - I_{app}=0.85 - See structure description in section 3.1.1. for presentation details. The upper part corresponds to a simulation origin to the bottom-left ; the lower part to a simulation origin to the top-right. The system can be seen to display both a lower stable equilibrium and an upper stable cycle, which is characteristic of SN-SH bifurcations.

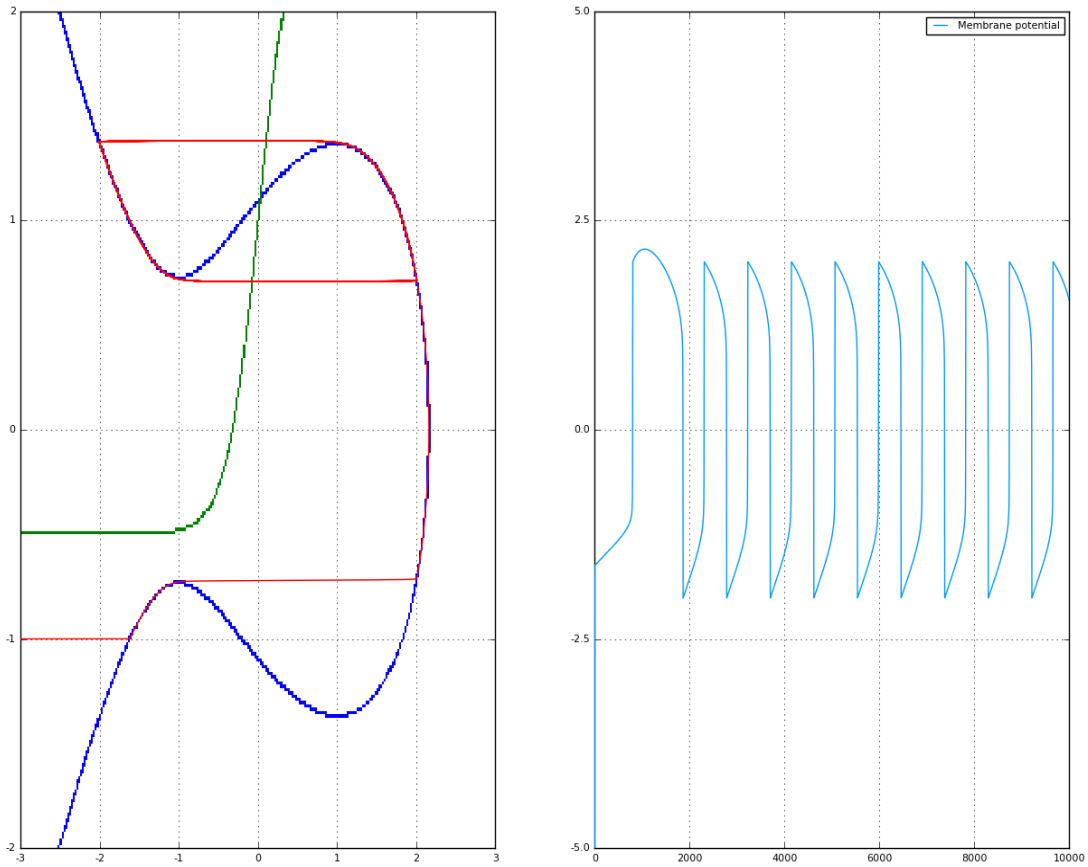

Fig 25: (V_0,n_0)=(0.0,-0.5) - I_{app}=1.2 - See structure description in section 3.1.1. for presentation details. The system can be seen to be spiking, as sustained oscillating activity can be observed.

Section 3.1.3 : Lower-left HOPF/SNIC region

The lower-left region is identified as a HOPF region and as a SNIC region by the NLsolve-based and Roots-based implementations, respectively, as shown in Figs. 21 and 22.

The coordinates chosen in order to observe the behaviour of the region are (V_o,n_o) = (-0.9,-0.5) . The figures to be studied for this region are the following :

- Fig. 26 : I_{app}=0.5 - At this value of the excitatory current, the system is at rest and at a safe distance from the bifurcation in terms of current. The stable equilibrium is situated on the lower part of the phase plane.
- Fig. 27 : I_{app}=2/3 - This value of the excitatory current corresponds to the tipping point of the transcritical bifurcation supporting the model, i.e. the value of the current at which the branches of the V-nullcline intersect. In this configuration, the proximity of the N-nullcline is obvious, and its effect on the trajectories in the system can only be guessed at, as the vector field between the two nullclines approaches zero in both directions. Compared to e.g. fig. 22 in the previous section, the transient regime can be seen to end only after step 4000, while in Fig. 22 it terminates after step 2000. The stable equilibrium has moved from the lower to the upper branch of the V-nullcline.
- Fig. 28 : I_{app}=0.67 - This value of the excitatory current is only slighly greater than that of fig. 26. The V-nullcline can be seen to have separated vertically, thus creating an opening that the trajectory must take while travelling from the lower-left origin towards the upper equilibrium and adding yet further delay to the convergence towards the equilibrium.
- Fig. 29 : I_{app}=1.05 - This value of the excitatory current corresponds to a configuration that is observed right before the bifurcation, considering a continuous increase in the applied excitation until the system destabilizes. Transient oscillations can be seen right before stabilisation, which is characteristic of HOPF bifurcations.
- Fig. 30 : I_{app}=1.2 - At this value of the excitatory current, the system is spiking at high frequency and at a safe distance from the bifurcation in terms of current.

Based on the evidence of transient oscillations in the vicinity of the bifurcation, it is concluded that this corresponds to a HOPF bifurcation, as predicted through the Roots-based implementation.

As a matter of fact, the NLsolve-based algorithm is manually found to fail converging around I_{app}~2/3, i.e. the critical point corresponding to the switch in stability in the transcritical bifurcation as depicted in Fig. 27. In the configuration of this region, the N-nullcline is situated very close to the entrance of the thus created bottleneck. The vector field in this region is therefore of near-zero amplitude along both dimensions, thus hampering the line search algorithm, and occasionally keeping it from converging in a reasonable amount of iterations, i.e. under a thousand iterations.

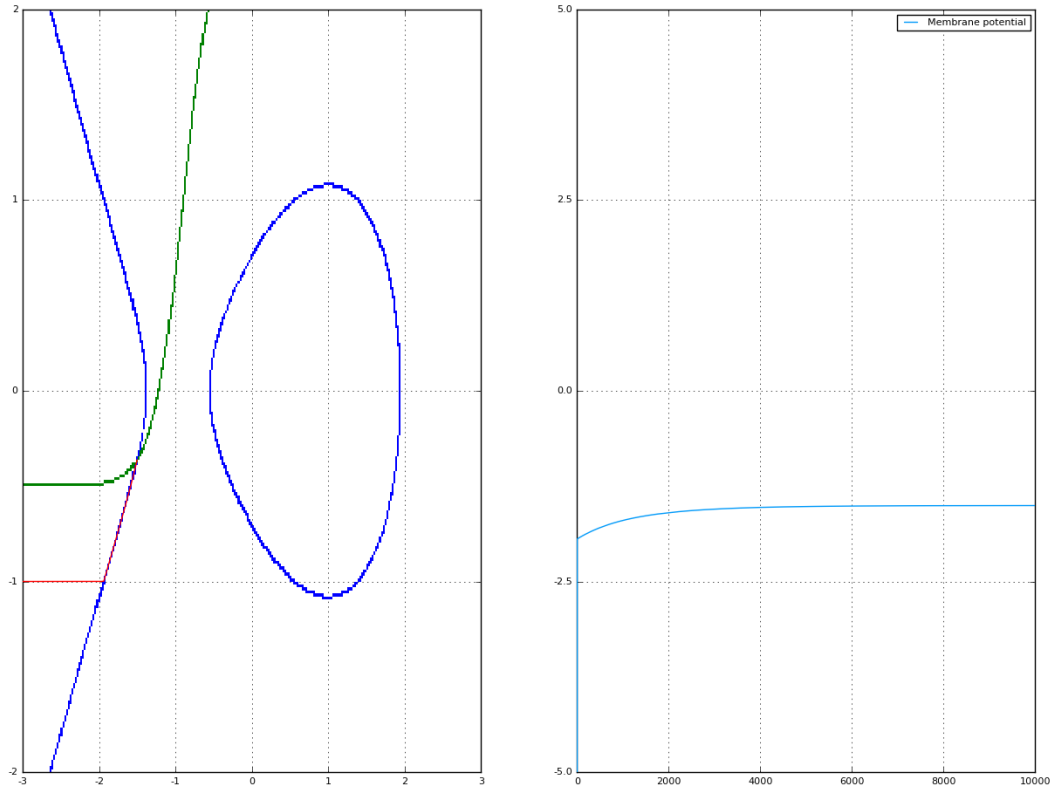

Fig 26: (V_0,n_0)=(-0.9,-0.5) - I_{app}=0.5 - See structure description in section 3.1.1. for presentation details. The system can be seen to be at rest on its only stable lower equilibrium.

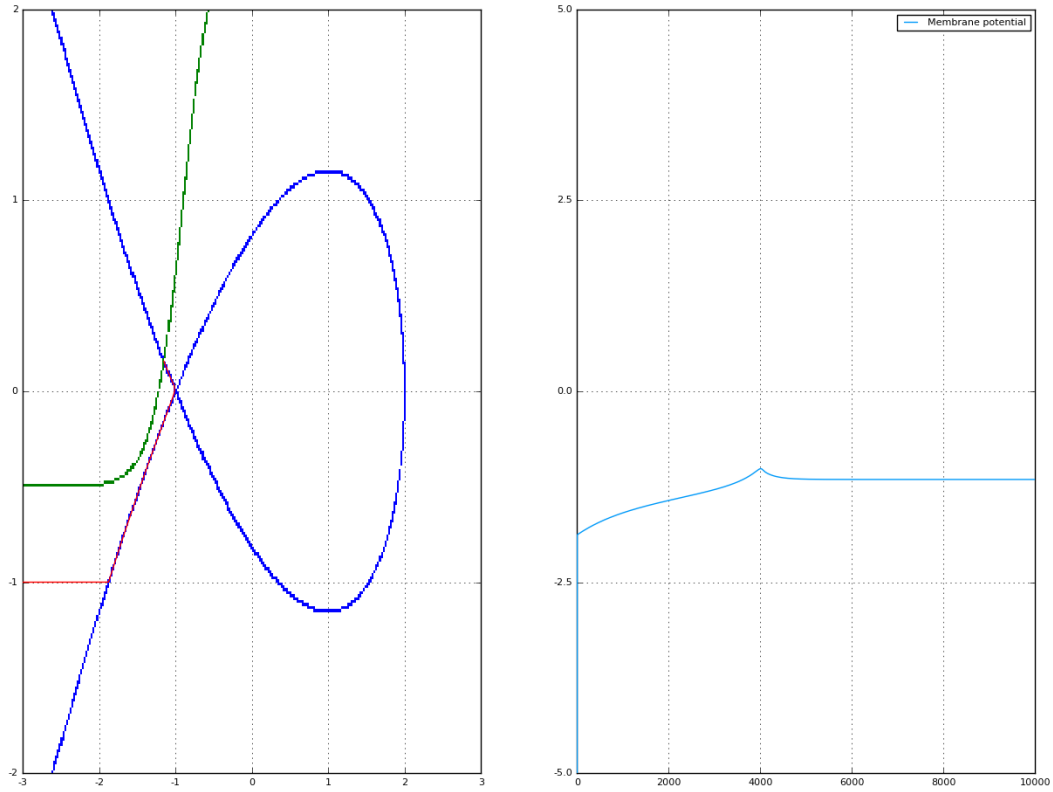

Fig 27: (V_0,n_0)=(-0.9,-0.5) - I_{app}=2/3 - See structure description in section 3.1.1. for presentation details. The N-nullcline can be seen to be situated extremely close to the intersections of the V-nullcline's branches, in such a way that trajectories are slowed down to near zero speed, as the vector field's two components approach zero.

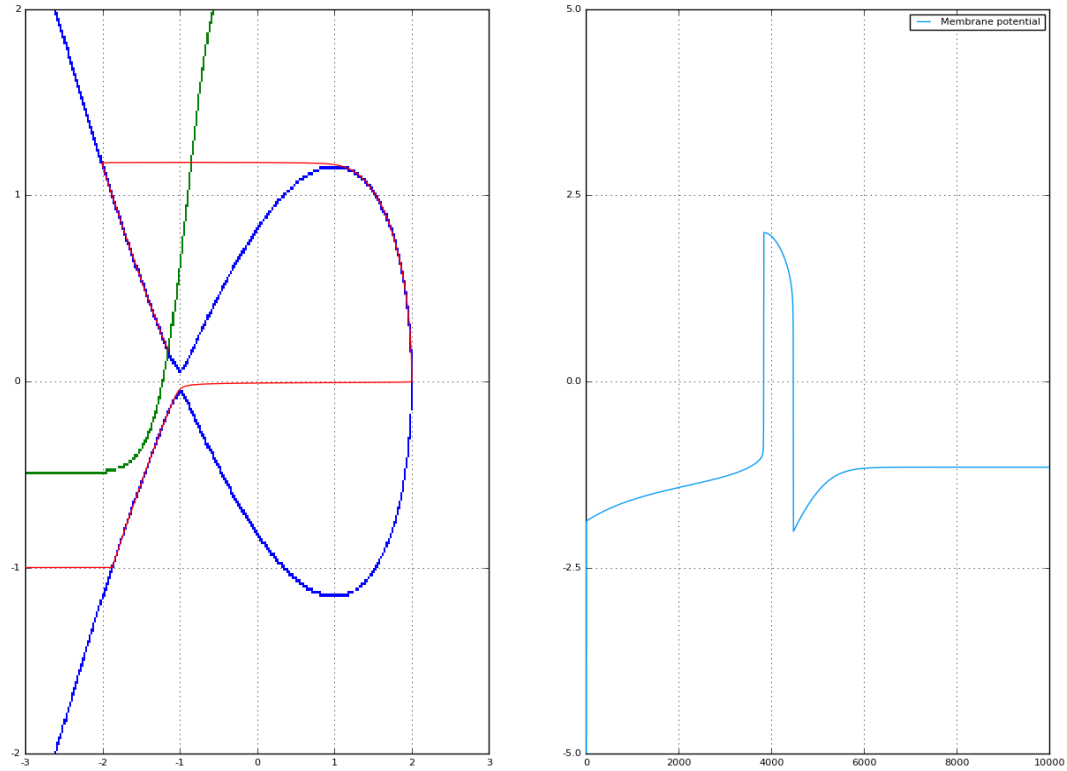

Fig 29: (V_0,n_0)=(-0.9,-0.5) - I_{app}=0.67 - See structure description in section 3.1.1. for presentation details. The system can be seen to be at rest on its only upper stable equilibrium after shooting a single action potential.

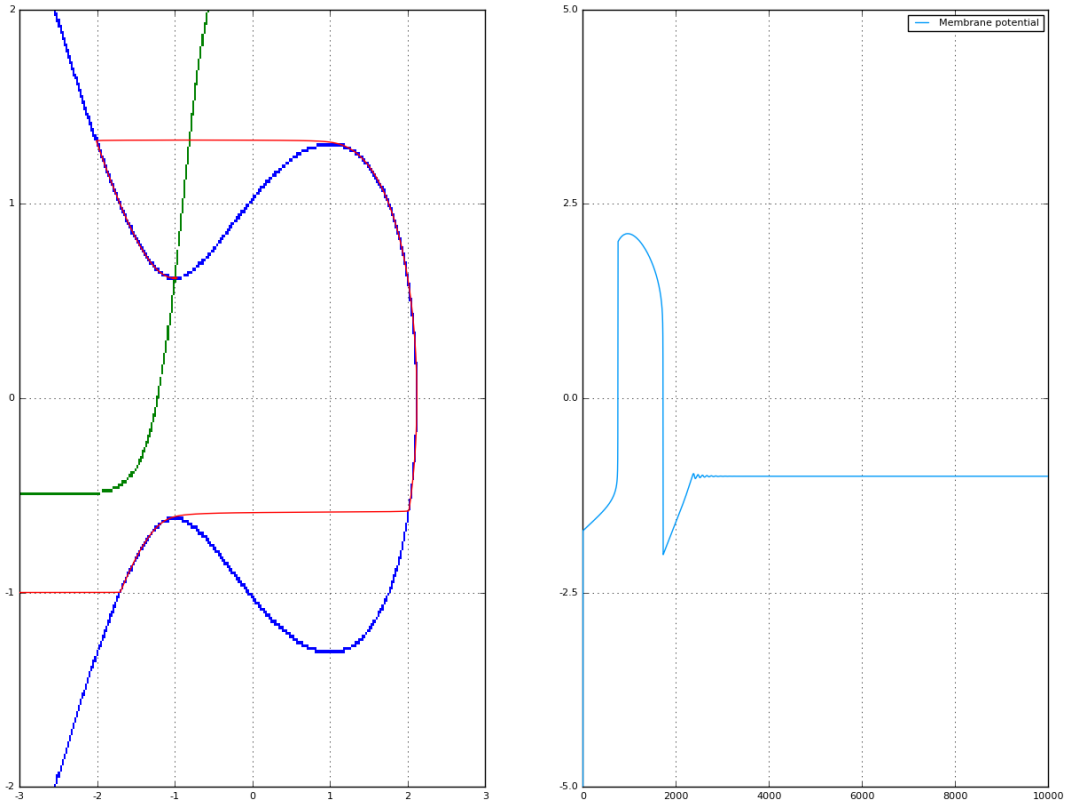

Fig 28: (V_0,n_0)=(-0.9,-0.5) - I_{app}=1.05 - See structure description in section 3.1.1. for presentation details. The system can be seen to be at rest on its only upper stable equilibrium ; however, transient oscillations can be observed right before stabilisation.

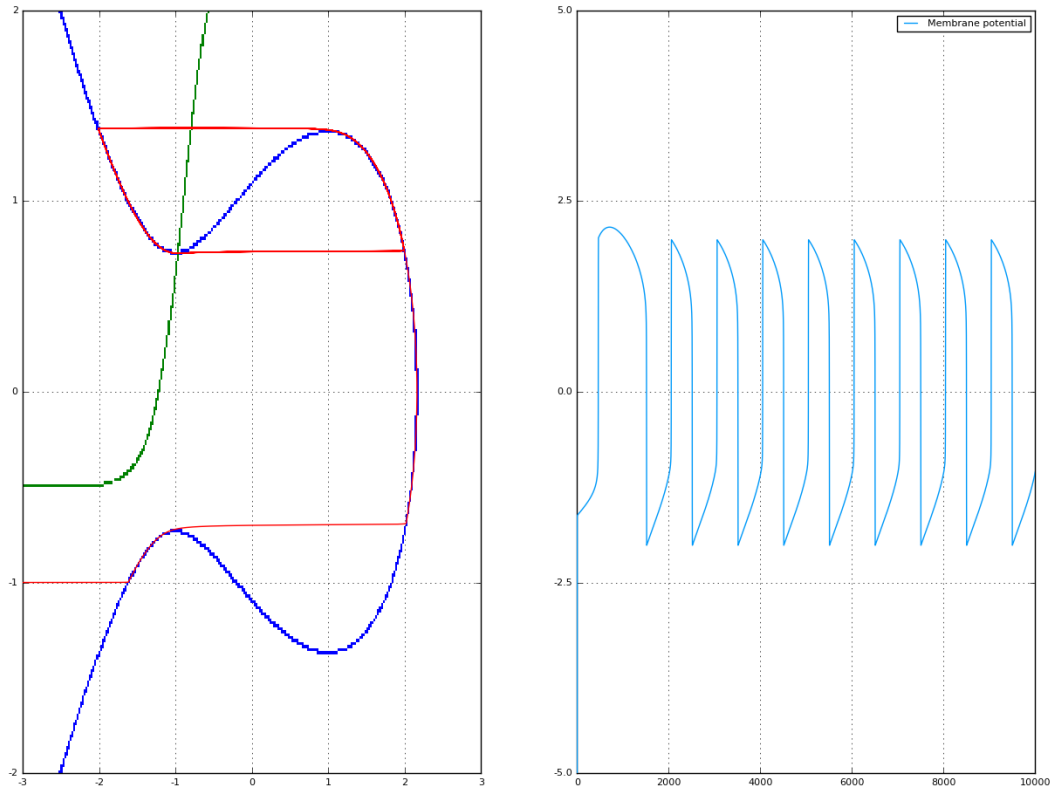

Fig 30: (V_0,n_0)=(-0.9,-0.5) - I_{app}=1.2 - See structure description in section 3.1.1. for presentation details. The system can be seen to be spiking, as sustained high-frequency oscillating activity can be observed.

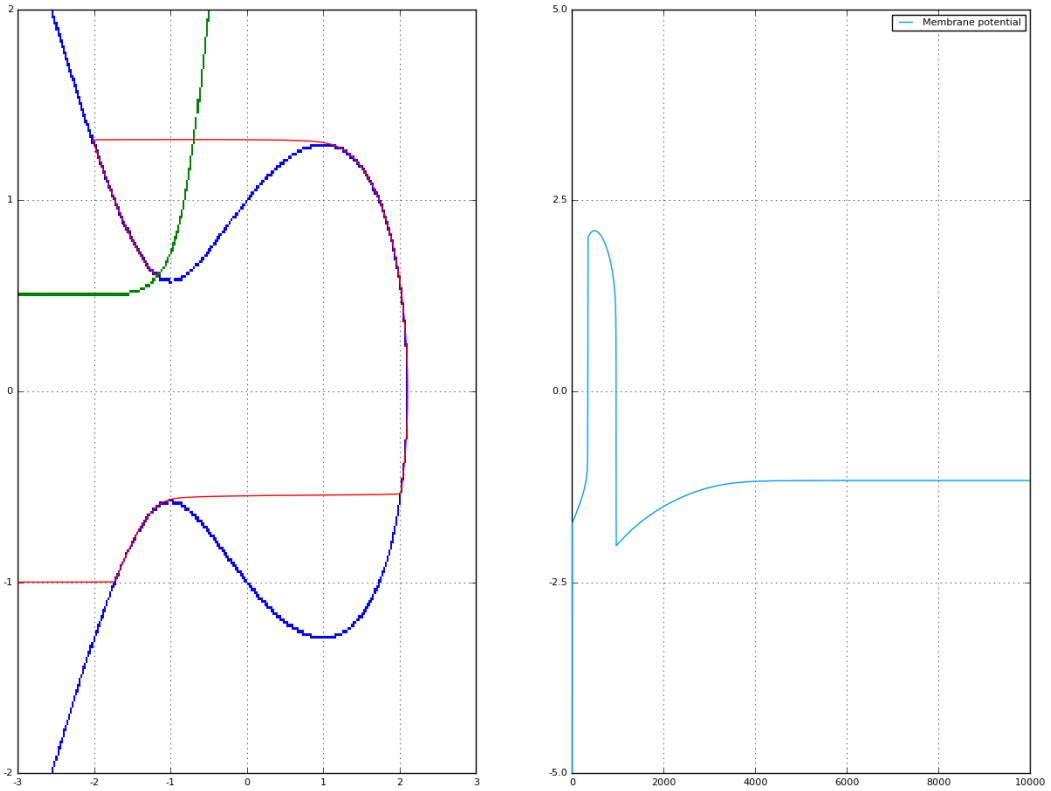

Fig 31: (V_0,n_0)=(-0.5,0.5) - I_{app}=1.0 - See structure description in section 3.1.1. for presentation details. The system can be seen to be resting on the stable upper equilibrium. No lower equilibrium could have existed for a lower excitatory current, considering how high the N-nullcline is situated.

Section 3.1.4 : Upper-left HOPF region

The upper-left region is identified as a HOPF region by both implementations, as shown in Figs. 21 and 22, as well as by the ODE-based implementation, looking back at the results presented in Fig. 18 in the section 2.3.2 concerning the assessment of the performances of this implementation.

The coordinates chosen in order to observe the behaviour of the region are (V_o,n_o) = (-0.5,0.5) . The figures to be studied for this region are the following :

- Fig. 31 : I_{app}=1.0 - At this value of the excitatory current, the system is at rest on its single stable upper equilibrium. Looking at lower values of the excitatory current would not bring anymore information, as the high position of the N-nullcline restricts the possibility of a lower stable equilibrium : there cannot be a lower intersection between the nullclines.
- Fig. 32 : I_{app}=1.19 - This value of the excitatory current corresponds to a configuration that is observed right before the bifurcation, considering a continuous increase in the applied excitation until the system destabilizes. Transient oscillations can be seen right before stabilisation, which is characteristic of HOPF bifurcations. The lower part of the figure displays a zoom on the transient oscillations, in order to highlight this behaviour of HOPF bifurcations.
- Fig. 33 - I_{app}=1.1979 - This value of the excitatory current corresponds to a configuration that is observed right after the bifurcation, considering a continuous increase in the applied excitation until the system destabilizes. High-frequency sustained oscillations can be observed, although the have only a very reduced amplitude. The lower part of the figure displays a zoom on the unstable equilibrium ; compared with Fig. 32, the cycle has stabilised, and the equilibrium at its centre has destabilised.
- Fig. 34 : I_{app}=1.5 - This value of the excitatory current places the system at a safe distance from the bifurcation, in terms of the value of the current. High-frequency and high-amplitude sustained spiking activity can be observed. Compared with Fig. 33, the amplitude has grown considerably, while the frequency seems to have stayed the same.

This region seems to cause no difficulty for any of the three implementation, and the visual inspection close to the bifurcation highlights patterns exclusively characteristic to HOPF bifurcations : the transient oscillating behaviour under the bifurcation, then the constant, high-frequency spiking of growing amplitude with the increase in the excitatory current.

This region is therefore concluded to be a HOPF bifurcation region, with no additional discussion.

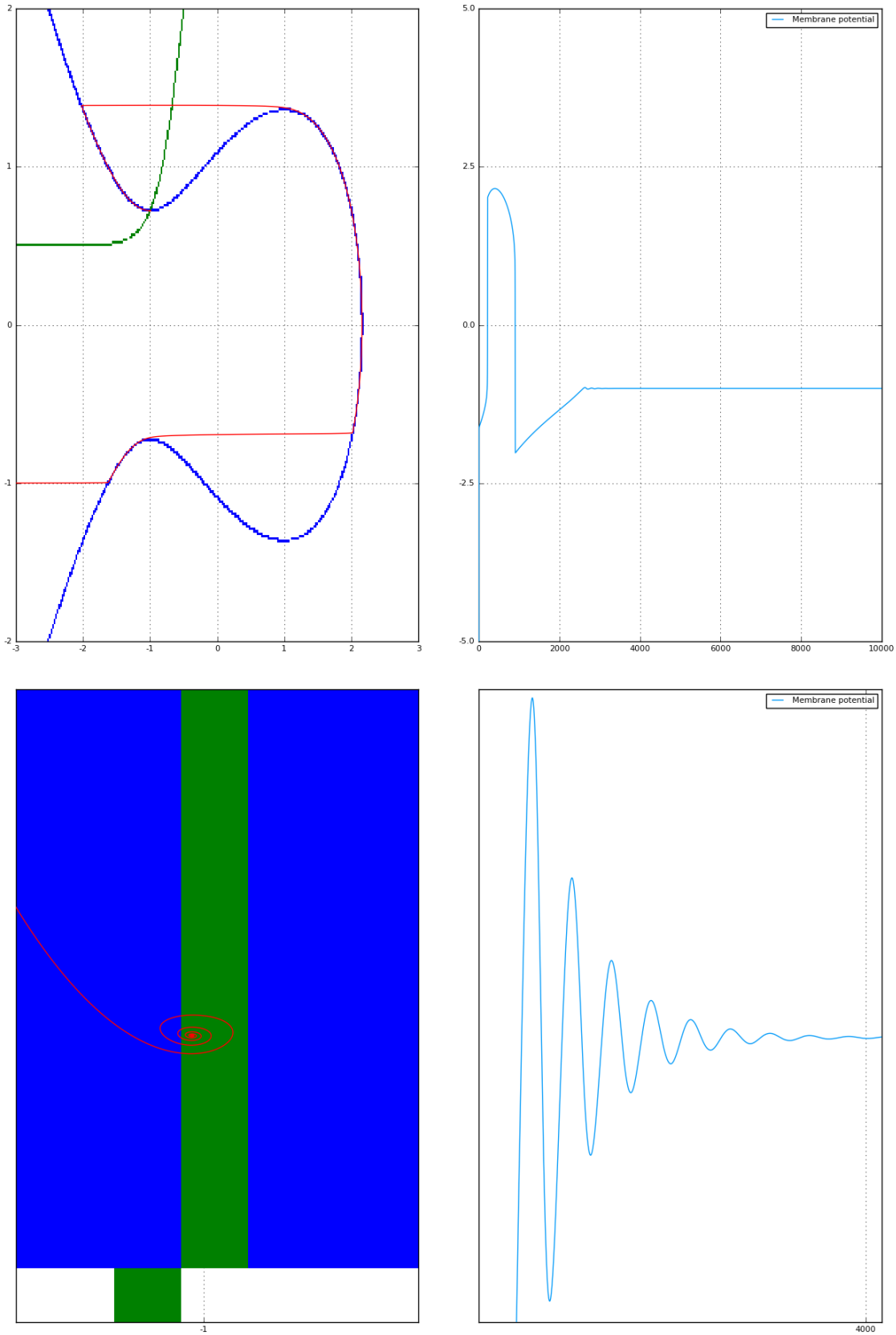

Fig 32: (V_0,n_0)=(-0.5,0.5) - I_{app}=1.19 - See structure description in section 3.1.1. for presentation details. The system can to be seen resting on the stable upper equilibrium ; transient oscillations can be observed at the end of the transient regime, as highlighted by the zoom on the lower part..

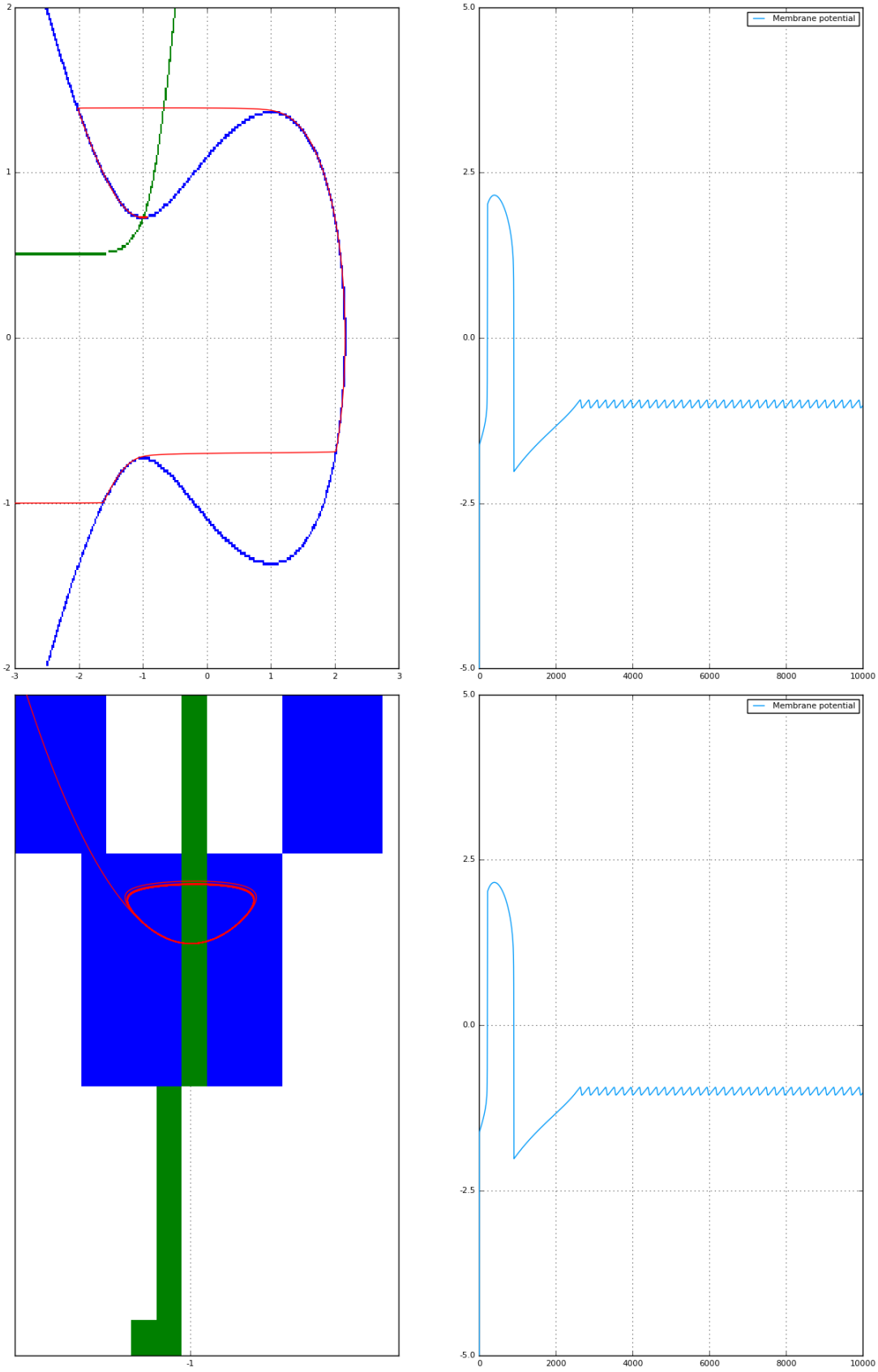

Fig 33: (V_0,n_0)=(-0.5,0.5) - I_{app}=1.1979 - See structure description in section 3.1.1. for presentation details. The system is undergoing sustained spiking activity, with spikes of low amplitude and high frequency, as is characteristic of HOPF bifurcations.

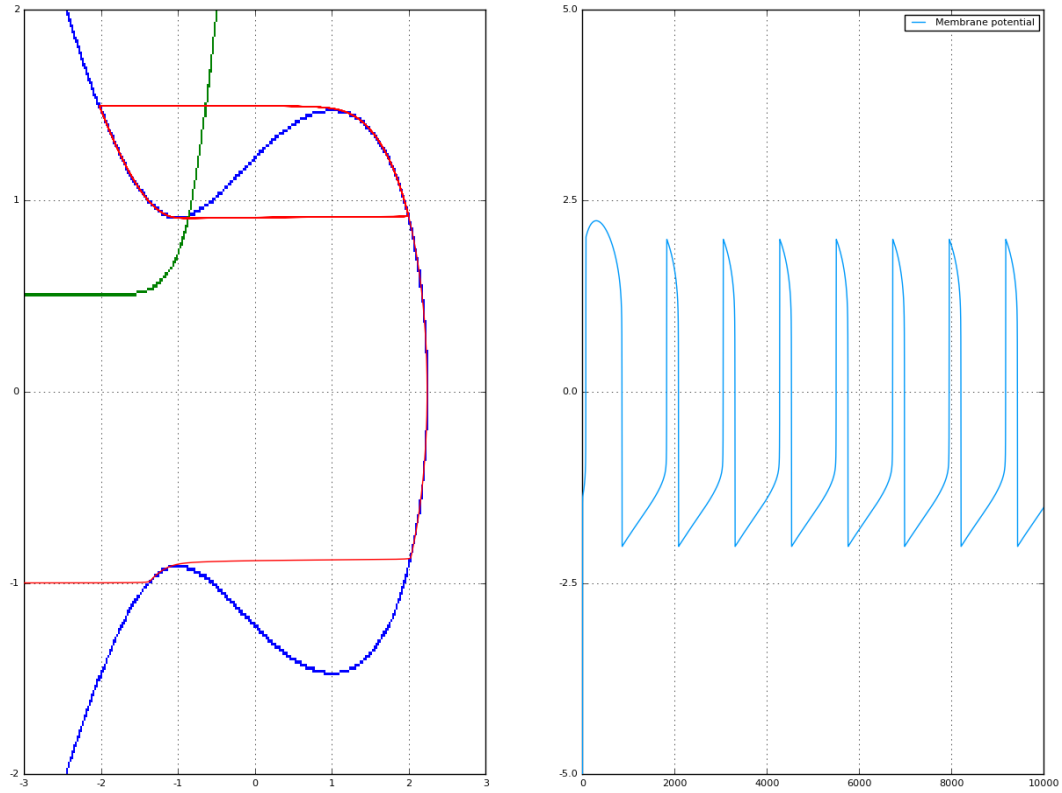

Fig 34: (V_0,n_0)=(-0.5,0.5) - I_{app}=1.5 - See structure description in section 3.1.1. for presentation details. The system can be seen to be spiking with spikes of high amplitude and high frequency ; compared with Fig. 33, the frequency seems to be roughly the same, as is characteristic of HOPF bifurcations.

Section 3.1.5 : Transcritical curve SNIC region

This region is a particular case that is best explained through the theory of non-linear systems and the fact that the transcritical bifurcation is a generalised case of the SNIC bifurcation ; however, this is a subject of its own that is outside of the scope of this document. Therefore, no justification of this region will be attempted, as its nature is the consequence of the structure of the model as presented in [8]. All numerical implementations correctly identify the bifurcations on the curve as SNICs.

As this model is based upon a transcritical bifurcation normal form, the transcritical bifurcation curve defines all the coordinates (V_o,n_o) for which the N-nullcline crosses the phase plane exactly at the position of the transcritical bifurcation critical point, which is placed in $(V_o,n_o)=(-1,0)$ in this model. As can be seen in Figs. 21 and 22, the other coordinates belonging to the transcritical bifurcation curve correspond to sliding the N-nullcline either to the left and to the bottom, or straight to the right ; in all cases, the nullclines will intersect in $(V_o,n_o)= (-1,0)$ at $I_{app}=2/3$.

As a visual example, the nullclines depicted in Fig. 35 corresponding to the configuration $(V_o,n_o)=(0.0,0.0)$ and excitatory current value $I_{app}=2/3$ can be seen to intersect right at the position of the transcritical bifurcation centre. No transient oscillations can be seen, which seems to point away from a HOPF bifurcation, but does not constitute in any form of valid proof.

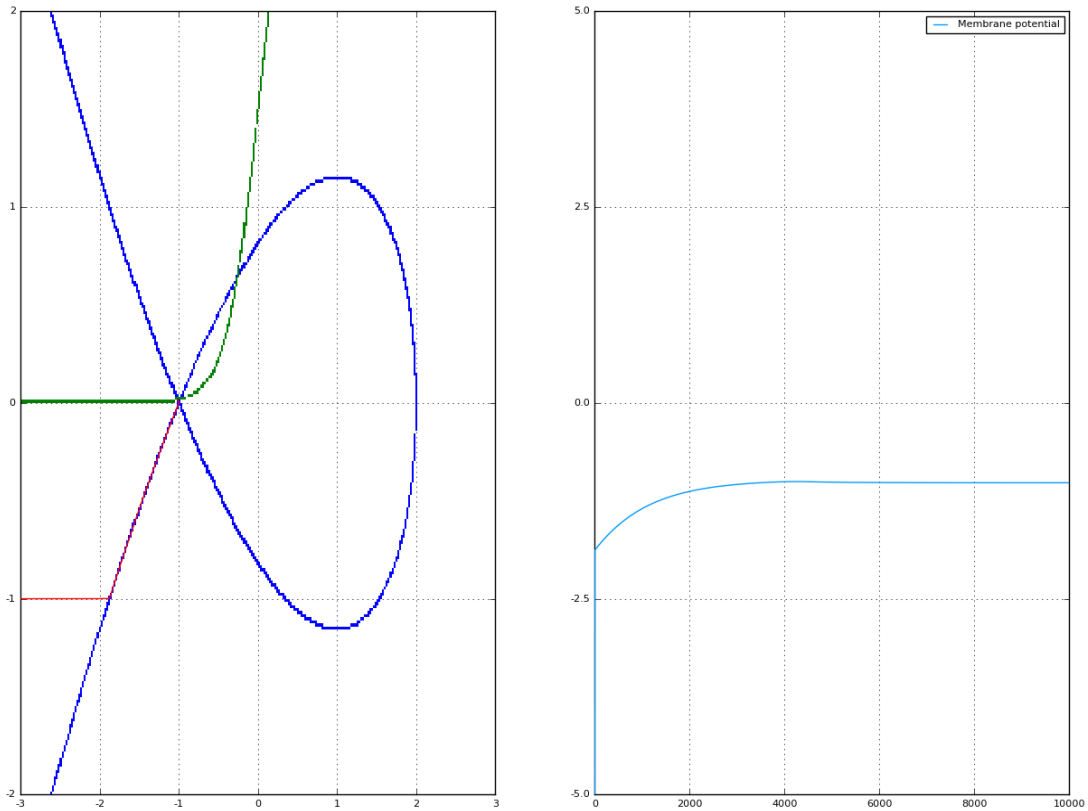

Fig 35: $(V_0,n_0)=(0.0,0.0)$ - $I_{app}=2/3$ - See structure description in section 3.1.1. for presentation details. The system is at rest, and the nullclines intersect exactly on the transcritical bifurcation. From bifurcation theory, it is known that transcritical bifurcations are generalised SNICs.

Section 3.1.6 : Upper-right HOPF/SNIC region

The upper-right region is identified as a HOPF and a noisy SNIC/HOPF region by the NLsolve-based and Roots-based, respectively, as shown in Figs. 21 and 22.

This region is a particular case in the bifurcation map, as discussed in sections 2.3.3 and 3.1.1, due the the very high precision that is required of the algorithm in order to determine the nature of its bifurcation. The exact configuration causing such difficulty for the Roots-based implementation, and any other implementation that would not work with a strict enough interval around the bifurcation from the bisection method, is presented in details in the second part of this chapter.

The visual inspection of the situation proceeds as follows : the nullclines depicted in Fig. 36 correspond to the configuration $(V_0,n_0)=(0.5,0.5)$ and excitatory current value $I_{app}=0.9$. No transient oscillations can be seen, which would eliminate a HOPF bifurcation. The local surroundings of the incoming bifurcation can easily be identified to a SNIC normal form, as the N-nullcline looks like a horizontal curve, while the V-nullcline can be approximated as a quadratic curve. Therefore, from visual inspection only, the bifurcation can be assumed to be a SNIC. This analysis is further discussed in section 3.2.2.

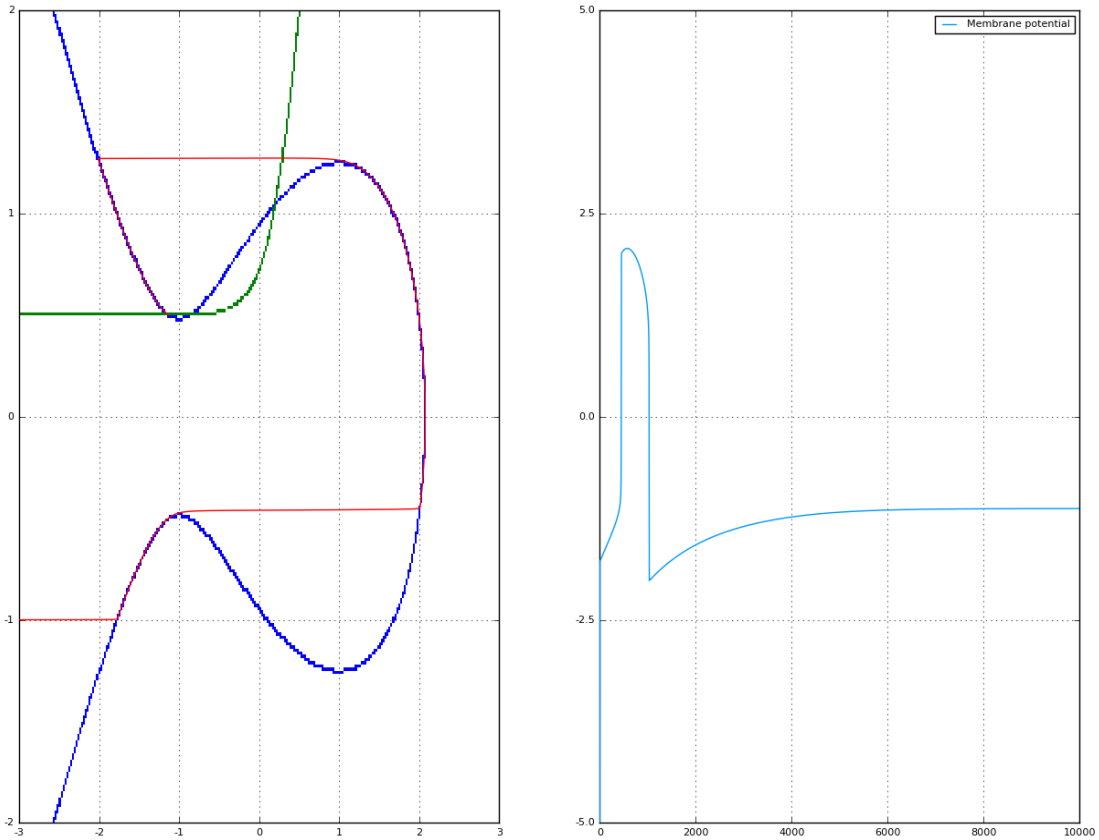

Fig 36: $(V_0,n_0)=(0.0,0.0)$ - $I_{app}=0.9$ - See structure description in section 3.1.1. for presentation details. The system can be seen to be at rest on the upper stable equilibrium. At this scale, this very much looks like a SNIC normal form, as the bifurcation is taking place in surroundings that can be visually identified to a horizontal N-nullcline and a quadratic V-nullcline.

Part 2 : Original findings

Section 3.2.1 : Variation of the time -scale coupling factor

In order to extend on the theoretical work performed in [8] for singular limits, the bifurcation maps corresponding to increasing values of the time-scale coupling factor ε, i.e. 10^{-2} and 10^{-1}, are computed through the interface, for both NLsolve-based and Roots-based implementations of the stability detection function. The results are illustrated on the Figs. 37 to 40.

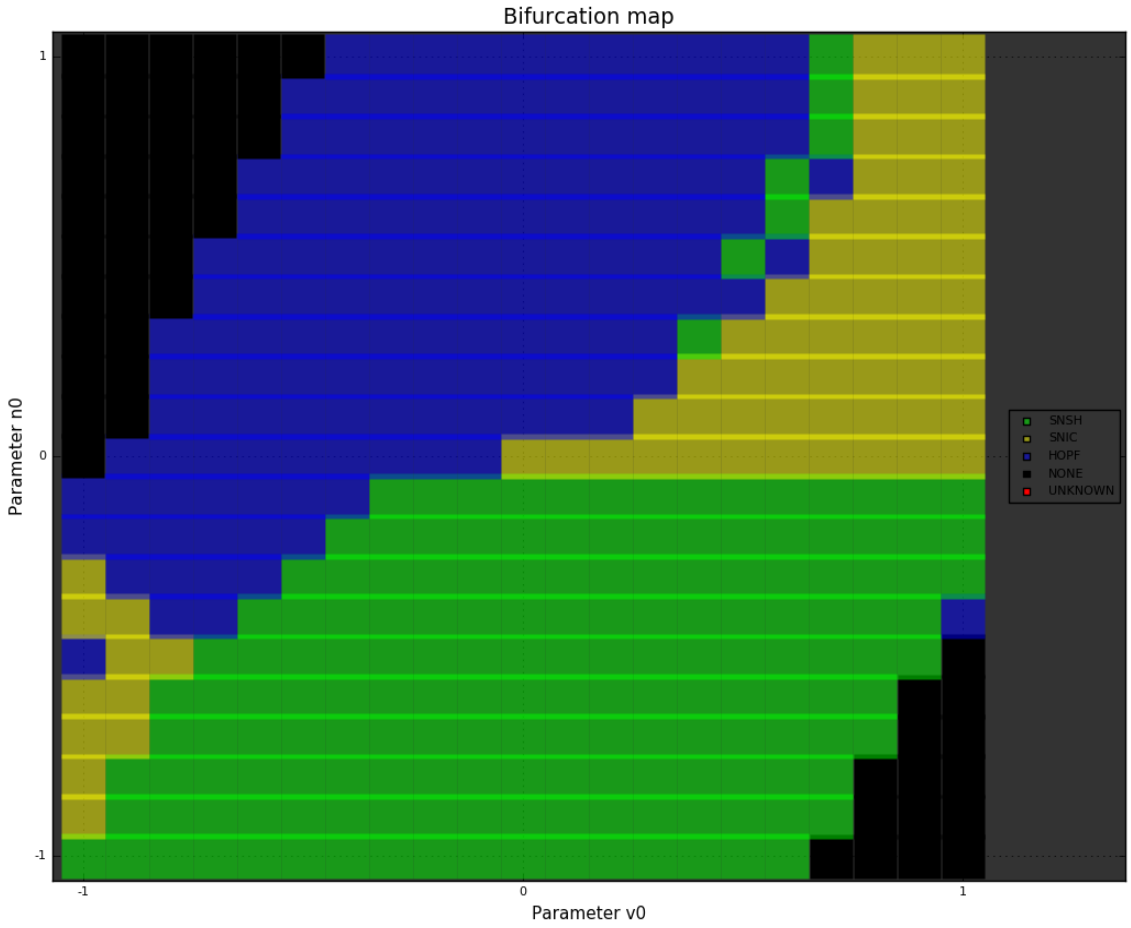

Fig 37: Bifurcation map of the parameter space (V_0,n_0) computed with the NLsolve-based implementation, for $\varepsilon=10^{-2}$. See internal legend and axis for details. The regions defined on the map are essentially identical to those of Fig. 21, except for the growth of a SNIC upper-right region in the originally HOPF upper region. Noise can be observed at the frontier between this new region and the HOPF region ; this noise being of SN-SH bifurcations in nature, it can be assumed that the corresponding coordinates are found to display pre-bifurcation oscillations, but no complex conjugates eigenvalues - as would a HOPF bifurcation display, if its complex eigenvalues could not be found at the lower border of the bisection interval.

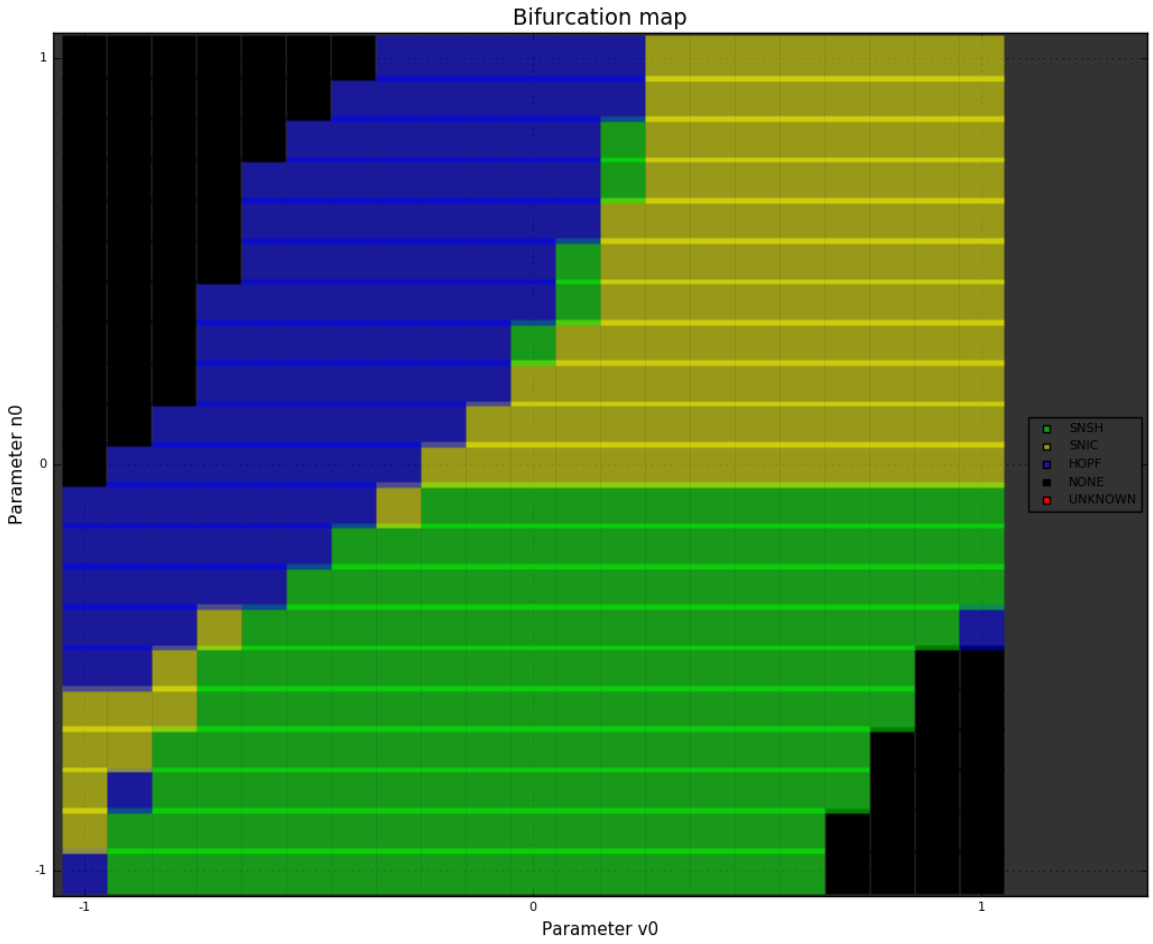

Fig 38: Bifurcation map of the parameter space (V_0,n_0) computed with the NLsolve-based implementation, for $\varepsilon=10^{-1}$. See internal legend and axis for details. The regions defined on the map are essentially identical to those of Figs. 21 and 37, except for yet further growth of the upper-right SNIC region.

The only changes that can be identified in Figs. 37 and 38 as the parameter ε increases are the modification in shape of the top-right and bottom-left regions. As the bottom-left region has been found to be a misclassification error in section 3.1.2, it is not discussed further, except for mentioning the fact that the increase in the parameter ε seems to alleviate the convergence problems of the NLsolve-based algorithms.

The growth of the top-right region is the most striking change in Fig. 38 w.r.t. to Figs. 21 and 37. As a matter of fact, this growth can be observed back in Fig. 18 and 20 in sections 2.3.2 and 2.3.4 ; although both cases are very noisy, the quantity of noise actually seems to decrease as the parameter ε increases, thus giving some credit to the accurate identification of this region for higher values of the time-coupling parameter.

The SN-SH bifurcations identified along the frontier between the top-right SNIC and the top-left HOPF regions are assumed to consist in misclassification noise only. The corresponding coordinates are found to display pre-bifurcation oscillations, but no complex conjugates eigenvalues at the lower border of the bisection interval. This corresponds to a HOPF bifurcation that is on the verge of transitioning to a SNIC bifurcation, as the parameter V_0 slides to the right.

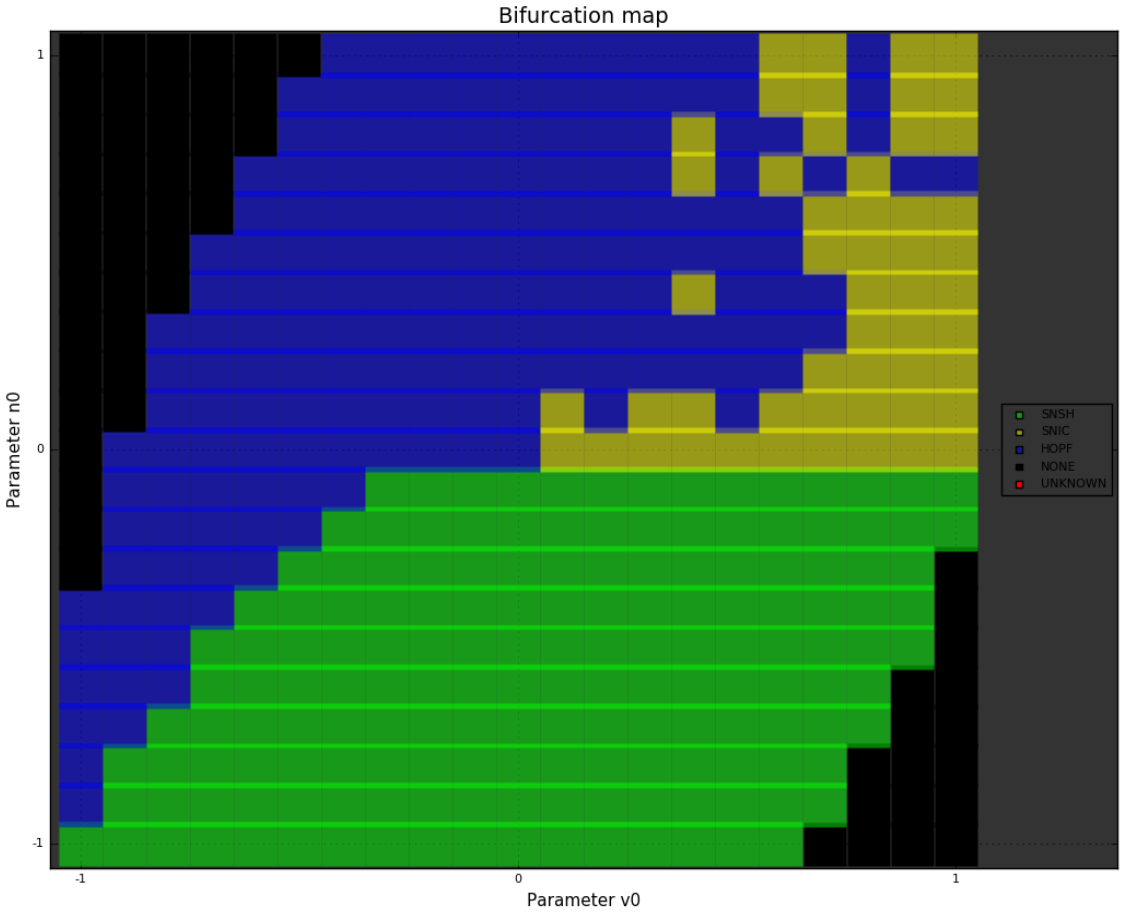

Fig 39: Bifurcation map of the parameter space (V_0,n_0) computed with the Roots-based implementation, for $\varepsilon=10^{-2}$. See internal legend and axis for details. The regions defined on the map are essentially identical to those of Fig. 22, except for the modification in the shape of the SNIC. The region seems to have shrunk, and its size seems to coincide with the corresponding region of Fig. 37, although heavy noise is still present.

As was the case with the NLsolve-based implementation, the increase in the parameter ε seems to only affect the upper-right region in Fig. 39 (as the Roots-based bifurcation maps did not display a bottom-left region, which was affected in the NLsolve-based bifurcation maps). In this case, the region seems to have shrunk down to approximately the same size and shape as that observed in Fig. 37 for the NLsolve-based implementation. While it is difficult to assess, the noise seems to have slightly decreased in quantity, although this may only be due to the reduction in size of the region.

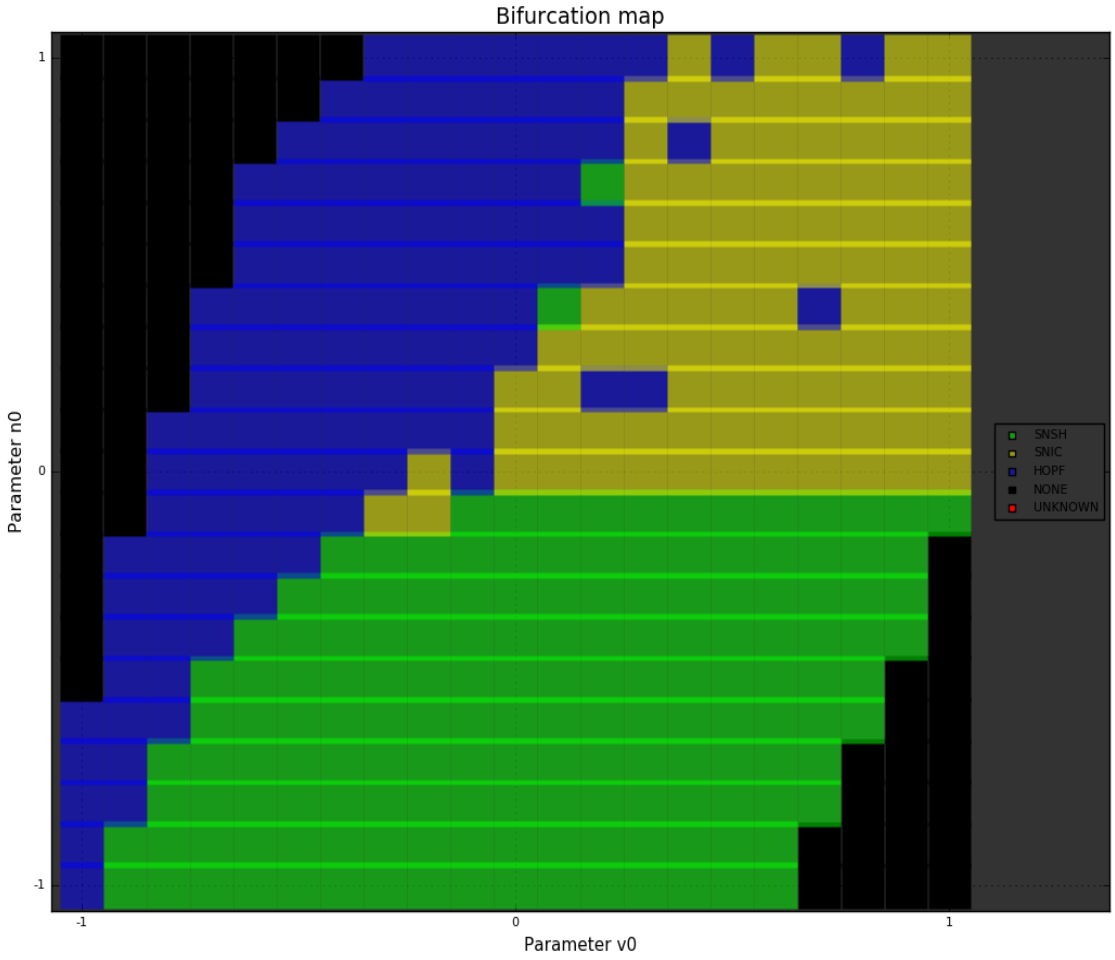

Fig 40: Bifurcation map of the parameter space (V_0,n_0) computed with the Roots-based implementation, for $\varepsilon=10^{-2}$. See internal legend and axis for details. The regions defined on the map are essentially identical to those of Fig. 22, except for the modification in the shape of the SNIC. The region seems to have shrunk, and its size seems to coincide with the corresponding region of Fig. 37, although heavy noise is still present.

Looking at the results in Fig. 40, the upper-right region seems to have grown back to its original shape and size, compared with Fig. 22. However, there is a distinct difference in the quantity of noise, as it seems to have decreased with the increase in the parameter ε. This corresponds to the general tendency hinted at by the Figs. 18 and 20 in sections 2.3.2 and 2.3.4 presented during the study of the implementations' performances.

A few points that are classified as SN-SH can be observed between the HOPF and noisy SNIC regions. They can easily be linked with the clean frontier observed in Figs. 37 and 38, as their location and noise-like nature coincide.

Overall, the nature and evolution of the noisy regions on the Roots-based maps, as well as the observed growth of the apparently noise-free region on the NLsolve-based maps seem to hint at a specific phenomenon. This is explored in details in the next section.

Section 3.2.2 : Upper-right HOPF/SNIC region explained

A rough sketch of the situation is provided in Fig. 41 in order to facilitate the visualisation of the configuration and the induced identification problem.

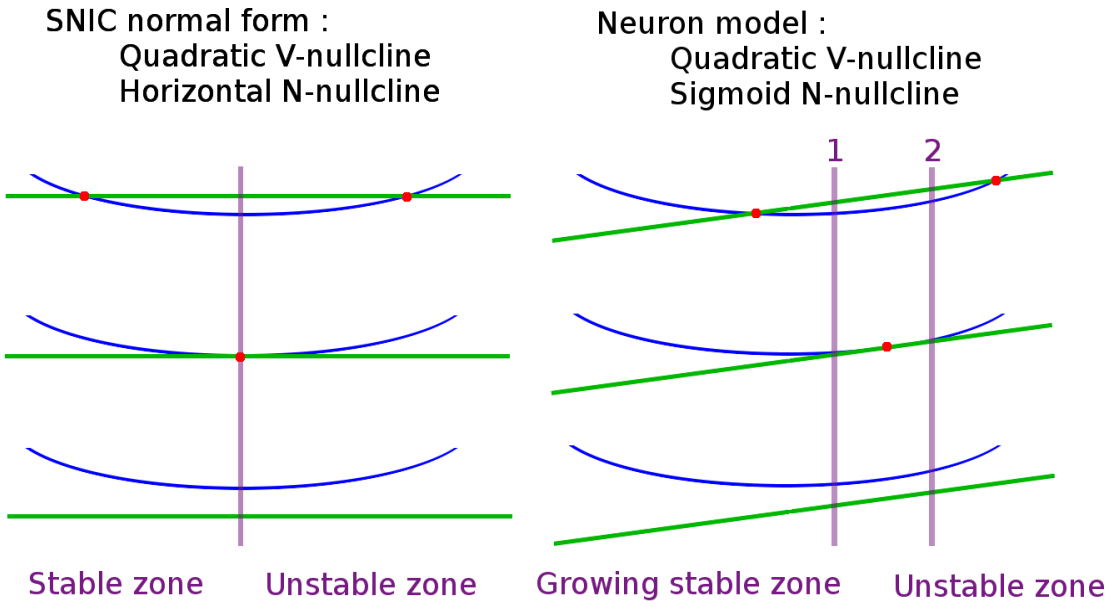

Fig 41: Sketch of the local configuration leading to problems in the identification for the upper-right region. The total sketch amounts for a zone of the order of 10^{-4} in width on the phase plane. The V-nullcline is represented as a blue curve that can be locally identified to a quadratic ; the N-nullcline is represented as a green curve that is either horizontal in the SNIC normal form, or is a sigmoid that can be locally identified to an oblique in the example neuron model. The red dots correspond to equilibria of any stability. The violet lines indicate the roughly horizontal separation between the zones leading to stable and unstable equilibria. On the right side, two violet curves marked "1" and "2" correspond to successive positions of the separation, as it moves to the right for increasing values of the time-scale coupling parameter ε.

The entire discussion on the nature of the bifurcation in this region, and the shape of the local configuration presented in Fig. 41, can be justified by the following facts :

- The nullclines corresponding to the activation variables in the original Hodgkin-Huxley model are experimentally determined to be sigmoids
- The canonical SNIC bifurcation normal form identifies the second nullcline to an horizontal, i.e. a local quadratic and horizontal configuration
- The example model preserves the original sigmoid nullcline, as shown in expression 8 and the numerous visual explorations performed in the previous sections
- The two equilibria collide, merge and disappear when the N-nullcline becomes tangent with the V-nullcline ; since for a sigmoid N-nullcline, this curve is not horizontal, this corresponds to a point that is farther to the right than the local minimum of the V-nullcline

In the case of the SNIC normal form, and all neuron models identified to it, the situation is fairly simple. As the N-nullcline (or the equivalent nullcline of the specific model that is considered) is horizontal, multiplying its corresponding equation in the system by a parameter such as ε that, in effects, controls the time-scale separation of the mechanisms of the system, does not affect the horizontal nature of the curve. Therefore, the V-nullcline and N-nullcline can only be tangent at the local minimum of the V-nullcline, i.e. the minimum of the quadratic to which it can be safely identified.

In this configuration, the vector field, which is strongly locally constrained by the proximity of the nullclines, is made horizontal by the symmetrical structure of the configuration. The phase plane can effectively be separated in two halves, with the left half allowing stable equilibria on the V-nullcline, while the right half only allowing saddles. When the curves become tangent, the left equilibrium is a node and the right equilibrium is a saddle - a saddle-node bifurcation happens.

Remarkably, this is the same situation as that of a one-dimensional system undergoing the same bifurcation, as pictured in Fig. 42. Taking into account the fact that non-linear systems analysis is most often than not taught by first studying one-dimensional, then only two-dimensional systems, it is easy to guess how visual inspection of two-dimensional systems could lead the observer to identify the two situations as very much alike, although the vector field is more complex in a two-dimensional system, and the second variable can cause the second curve that is pictured to differ from a pure horizontal.

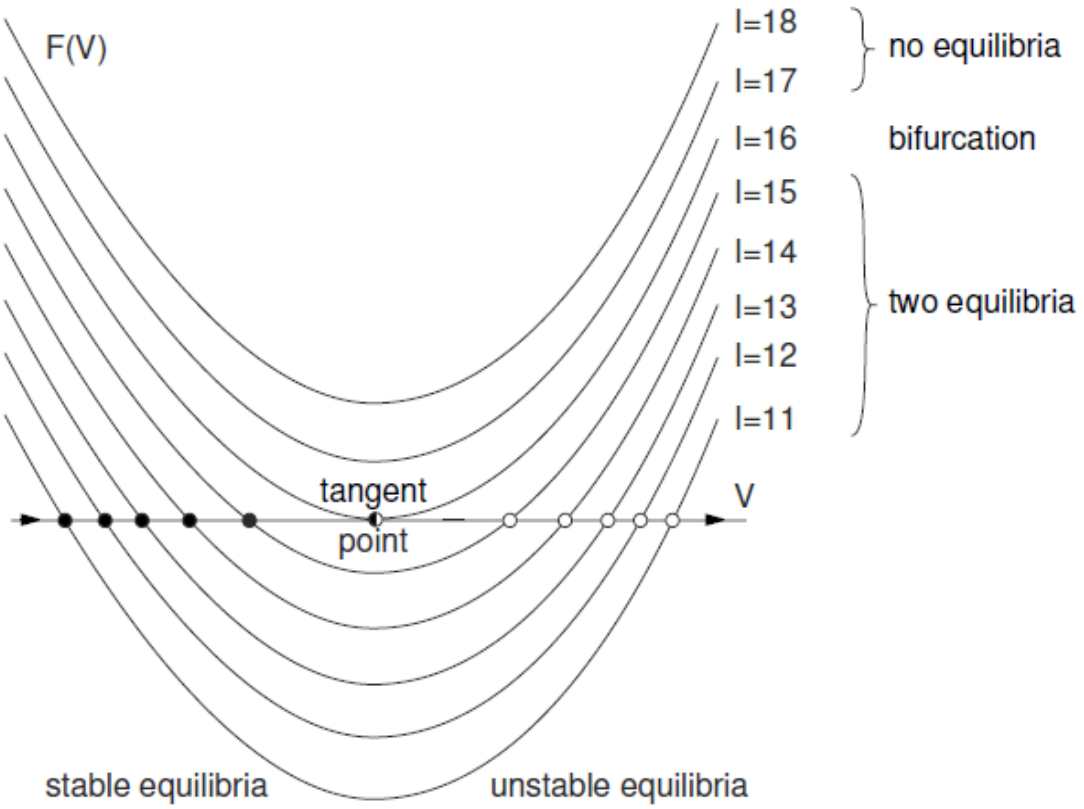

Fig 42: Representation of a saddle-node bifurcation in a one-dimensional system. In this case, the second curve that is considered is always a perfect horizontal. (extracted from [5] - pg 90)

In the case of the canonical TC model, and any other model that preserves the experimentally determined sigmoid shape of the activation variable's nullcline, the situation is complicated by the fact that the vector field is affected by the inclination of the second curve and the magnitude of the parameter ε.

The inclination of the N-nullcline implies that the two curves become tangent at a position that is not the local minimum of the V-nullcline. In this case, this point is located farther to the right. The shape of the vector field, in this new configuration, becomes critical to the identification of the bifurcation.

Assuming singular limits, i.e. approximately zero value for the parameter ε, or stated otherwise, assuming that the generation of action potentials is instantaneous compared to the relaxation phase of the system, the vector field can be assumed to be horizontal, as in the SNIC normal form configuration. In the example model, however, this means that the bifurcation cannot be a SNIC, as the left stable equilibrium will be destabilised by crossing the local minimum and going farther to the right, up to the tangent point When the curves become tangent, two saddles collide and disappear. A HOPF bifurcation has therefore happened before the system underwent any saddle-node bifurcation, rather undergoing a "saddle-saddle" bifurcation.

This corresponds to the NLsolve-based implementation's final assessment that the upper-right region of the bifurcation map is a HOPF region, for very low values of ε, e.g. for $\varepsilon=10^{-3}$. The Roots-based implementation, in turn, outputs incorrect results. This is due to the fact that the local configuration sketched in Fig. 41 corresponds to a width under about 10^{-4} units. This is shorter than any sub-interval division supported by the package's methods - requiring the package to determine the position of equilibria that are this close to each other causes it to randomly fail, thus creating misclassification noise.

When the strong time-scale separation hypothesis is relaxed, corresponding to an increase in the time-scale coupling parameter ε, the local vector field bends upwards and the stable zone separation pictured in violet in Fig. 41 slides to the right. For a given set of coordinates, the position 1 in the figure may correspond to $\varepsilon=10^{-2}$, while the position 2 may correspond to $\varepsilon=10^{-1}$. At a sufficiently low time-scale separation, this separation crosses over the location corresponding to the saddle-node bifurcation. In this new configuration, the saddle-node bifurcation happens before the node is destabilised ; the node disappears in a SNIC and the model therefore displays type I excitability.

This corresponds to the NLsolve-based implementation's final assessment that the upper-right SNIC region grows with increasing values of ε. Furthermore, it justifies that the amount of noise in the Roots-based implementation decreases with ε, as stable equilibria considered to the left or to the right of the V-nullcline's local minimum are both SNIC-like equilibria, i.e. with pure real eigenvalues.

Section 3.2.3 : Importance of the HOPF/SNIC region

In order to assess the importance of the observation of the new region, a few facts must be presented.

Over the years, most bifurcations have been identified on the sole basis of visual inspection of the phase plane of the system. The main way to do this was to identify the normal form of the bifurcation that looked the most like the current situation ; in our case, the TC model can be said to look like it displays a local configuration very much alike to that of a normal-form SNIC bifurcation, as hinted at by the quasi-quadratic V-nullcline and quasi-horizontal N-nullcline. For example, the authors of [5] and [6] have built a number of models on the sole basis of this visual identification with a SNIC bifurcation. However, the fact that the N-nullcline isn't horizontal changes everything, as explained in the previous section, thus justifying the construction of the example model based on a transcritical bifurcation as presented in [8].

As a great number of the models most in use such as the quadratic integrate-and-fire are in fact built on the basis of these normal forms, it is therefore only logical that the bifurcations observed in these models are in fact the very same that the models are based upon, i.e. SNIC bifurcations.

What is more, physiological neurons do not correspond to these normal forms ; looking back at the classical Hodgkin-Huxley model built through fit to extensive experimental data, one can immediately notice that the N-nullcline is in fact a sigmoid. This curve is only asymptotically horizontal when its parameter tends towards negative infinity ; for real, physiological values of the activation variable represented by this parameter, the curve has a positive non-zero and non-constant slope.

Finally, while the methods developed for this work were only applied to the example model extracted from [8], they are built in order to be easily applied to other models - the use of generic numerical algorithms ensures that most two-dimensional systems can be solved, regardless of their order and complexity. Such analysis as that covered in the present work should be applied to known models, and would probably output results that may contradict simple visual analysis due its differences with rigorous numerical analysis at small enough scales.

Furthermore, earlier works hinted at the ability of a number of neurons to switch between excitability types, i.e. between SNIC and HOPF bifurcation patterns. The simple change in the configuration of the system of the canonical TC model that caused so much difficulty in the identification of the upper-right region's bifurcation is an example of a fairly elementary mechanism that could underlie some type of neuromodulation. ([5])

Conclusion

On the current results

A proof-of-concept interface solving the problem of bifurcation identification in a neuron mathematical model has been successfully built and tested on a canonical TC model. This interface was built in the compiled Julia programming language and effectively displays competitive computational efficiency compared to other usual solutions such as the Matlab or Python interpreted environments. The bifurcation identification algorithm is based on robust well-known root-finding algorithms such as the Newton-Raphson or Trust-region algorithms, and the additional logic cores specializing the use of these algorithms to the field of bifurcation analysis can be easily located and maintained as required.

In particular, the use of high convergence rates root-finding algorithm is especially adapted to the application of this interface to high-dimensional systems, thus placing the NLsolve-based stability detection function as the best contestant versus its two proposed rivals, notwithstanding its better computational efficiency and accuracy on the example two-dimensional model.

The application of the interface on the canonical TC model resulted in a strong argument against the common practice of neuron model reduction to bifurcation normal forms through simple visual inspection. Rigorous numerical identification of the bifurcations of the model as a function of its two static parameters proved the ability of a transcritical bifurcation-based neuron model to display various excitability types experimentally found in real neurons. Identically rigorous numerical identification of the bifurcations of the model as a function of the time-scale separation between its internal mechanisms highlighted the critical importance of dynamical numerical analysis of neuron models versus purely static, and possibly only visual analysis in order to determine the exact bifurcation underlying the original high-dimensional models before their reduction to bifurcation normal forms.

In particular, the example model displayed a local configuration very much alike to that of a SNIC bifurcation normal form model. The numerical analysis performed through the interface therefore raises controversy regarding numerous works in the scientific literature concerned with the analysis of neuron models through simple visual inspection of SNIC normal form-like local configurations and the subsequent verdicts of type I excitability allegedly displayed by these models.

In general, the exact origin of type I excitability is challenged as two distinct regions of type I excitability are found in the example model, and a region commonly identified on the basis of visual inspection to type I excitability is numerically proven to display type II excitability. This, in turn, challenges the use of commonly used neuron models in physiological neuronal networks, as the models commonly believed to display type I excitability may do so on the entire range of their parameters, or may do so with specific associated properties that ought to be taken into account, such as variations in their robustness to noise.

On going further

Out of the three proposed implementations, the NLsolve-based implementation proved the fastest and most accurate. Further work and tweaking of the parameters of the other two implementations, however, may bring about better results that sets them as valid options of their own. In particular, finding another, or modifying the existing bracketing strategy of the Roots-based implementation may bring its accuracy up to the level of the NLsolve-based implementation. Better mastery of the parameters of the ODE solvers, as well as theoretically motivated convergence assessment and oscillation detection methods may bring the ODE-based implementation's accuracy up and farther than the other two, while its computation time may be kept under reasonable lengths of time, or accepted as the cost of extreme accuracy.

Furthermore, a number of tolerances and numerical threshold have only be determined experimentally ; extensive study of the interval in terms of excitatory current between two bifurcations in a neuron and between the apparition of the eigenvalues pattern characteristic to the bifurcation and the bifurcation itself should motivate better values of those tolerances and thresholds.

The bifurcation mapping procedure should in particular see as a great improvement the introduction of methods known to be in use in the fields of computer-aided design applied to mechanics, i.e. that of structuring the space to be studied in a mesh. In this context, the parameter space (V_o,n_o) could be evaluated intelligently, with only broad sweeps in the stable regions, and precise, smaller sweeps at the frontiers and any other zones of transition in the classification of the bifurcations of the system. This would improve both the computational efficiency and readability of the final bifurcation map.

An analysis of the robustness of the two type I excitability regions identified in the canonical TC model would doubtlessly output interesting results. As the n_o parameter corresponds to an image of the balance between the restorative and regenerative ion channels in the neuron, or in other words a balance between negative and positive feedback mechanisms induced by the very same ion channels, its value coincides with configuration of fairly open ion channels, or fairly closed ion channels. [8],[9],[20] Therefore, since the permeability of ion channels directly identifies to the conductivity of the membrane to the corresponding ion currents, it is expected that the region corresponding to a configuration where the ion channels are fairly closed will be strongly affected by noise. The region corresponding to a situation where all ion channels are fairly open, however, should display strong resilience against noise, as the associated conductivities would be fairly high. These notions, and the general topic of ion channel degeneracy affecting the resistance of neurons to noise thanks to the action of multiple open ion channels is discussed in [21].

Appendix

Part 1 : Non-linear systems analysis

Section 1.1 : Poincaré-Bendixson theorem

Supposing that :

- R is a closed, bounded subset of the plane ;
- $\dot{x} = f(x)$ is a continuously differentiable vector field on an open set containing R ;
- R does not contain any fixed points ; and
- There exists a trajectory C that is "confined" in R, in the sense that it starts in R and stays in R for all future time

Then either C is a closed orbit, or it spirals toward a closed orbit as time tends towards infinity. In either case, R contains a closed orbit. [7]

Part 2 : Numerical root-finding algorithms

Section 2.1 : Bisection method

Supposing that :

- f(x) is a continuous function defined on an interval [a,b]
- f(a) and f(b) have opposite signs

Then per the intermediate value theorem, the continuous function f must have at least one root in the interval [a,b]. Assuming there is one, and only one root, iterate the following steps to find f(x) = 0 where x is real :

- Compute the midpoint c of the interval [a,b], i.e. $c = \frac{a+b}{2}$
- Compute the value of the function f(x) at the position c, i.e. f(c)
- Replace either a or b as the borders of the interval, according to whether f(c) has the same sign as f(a) or f(b), respectively, in order to stay in the conditions of the intermediate value theorem

The iterations must be stopped when an interval of the desired width has been selected.

The number of iterations required to reduce an interval from a width w0 to a width w1 can be computed as follows, where n is the required number of iterations :

$$n = \log_2\left(\frac{w_0}{w}\right) = \ln\frac{(w_0/w)}{\ln(2)}$$

Section 2.2 : Trust-region method

The trust-region is a numerical root-finding method that approximates the objective function (i.e. the system to solve) in a local subset using a quadratic approximation ; if the approximation is found to be accurate, the local region is expanded, else it is contracted.

The accuracy of the approximation is computed as the ratio between an expected measure of improvement and a measure of actual improvement in the prediction of the objective function's value. A threshold is set as a limit defining a "trustworthy" model in a local subset.

Section 2.3 : Newton-Raphson method

The Newton-Raphson is a numerical root-finding method that approximates the objective function (i.e. the system to solve) using a quadratic approximation. The local minimum of the function is then approached by a line search strategy, i.e. by computing a descent direction as the direction of the minimum of the local quadratic approximation of the objective function. A step size determining how far the algorithm should move in that direction is then computed and applied in the descent direction.

This method displays convergence rates of a quadratic order.

Bibliography

1: John H. Byrne, Ruth Heidelberger, M. Neal Maxham, From Molecules to Networks (Third Edition), 2014, Academic Press, Boston
2: Hodgkin. A., The local electric changes associated with repetitive action in a non-medullated axon, 1948, Journal of Physiology
3: Peter Dayan, L. F. Abbott , Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems, 2005 , The MIT Press
4: Connor JA, Stevens CF, Prediction of repetitive firing behaviour from voltageclamp data on an isolated neurone soma, 1971, J Physiol 213(1):31–53
5: Izhikevich E.M., Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting, 2007, The MIT press
6: FitzHugh R., Impulses and physiological states in theoretical models of nerve membrane, 1961, Biophysical J. 1:455-466
7: Steven H. Strogatz, Nonlinear Dynamics and ChaosWith Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity), 2001, Westview Press
8: Franci A, Drion G and Sepulchre R, An organizing center in a planar model of neuronal excitability, Drion G, Franci A, Seutin V, Sepulchre R, SIAM J. Appl. Dyn. Syst., 11:1698-722
9: Drion G, Franci A, Seutin V, Sepulchre R, A novel phase portrait for neuronal excitability, 2012, PLoS One. 7:e41806
10: Ivo Balbaert, Avik Sengupta, Malcolm Sherrington, Julia: High Performance Programming, 2016, Packt Publishing
11: The Julia language official website, https://julialang.org/
12: Julia Package - ODE.jl, https://github.com/JuliaDiffEq/ODE.jl
13: Rackauckas C., Nie Q., DifferentialEquations.jl - A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia, 2017, Journal of Open Research Software
14: Julia Package - NLsolve.jl, https://github.com/JuliaNLSolvers/NLsolve.jl
15: Julia package - Optim.jl, https://github.com/JuliaNLSolvers/Optim.jl
16: Louveaux Q., Introduction aux Methodes Numeriques, 2013, Universite de Liege, Faculte des Sciences Appliquees
17: Julia package - Roots.jl, https://github.com/JuliaMath/Roots.jl
18: Julia Package - Plots, https://github.com/JuliaPlots/Plots.jl
19: Reliable Two-Dimensional Graphing Methods for Mathematical Formulae with Two Free Variables,
20: Franci A, Drion G, Seutin V, Sepulchre R, A Balance Equation Determines a Switch in Neuronal Excitability, 2013, PLoS Comput Biol 9(5): e1003040.doi:10.1371/journal.pcbi.1003040
21: Drion G, O'Leary T, Marder E, Ion channel degeneracy enables robust and tunable neuronal firing rates, 2015, Biology Department and Volen Center, Brandeis University, Waltham, MA 02454